



Mémoire présenté devant le jury de l'EURIA en vue de l'obtention du
Diplôme d'Actuaire EURIA
et de l'admission à l'Institut des Actuaire

le 21 Septembre 2017

Par : Rémi GAUVILLE

Titre : Projection du ratio de solvabilité : des méthodes de machine learning pour contourner les contraintes opérationnelles de la méthode des SdS

Confidentialité : Non

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

***Membre présent du jury de l'Institut
des Actuaire :***

Anthony Nahelou

Signature :

Entreprise :

Fixage Actuariat

Signature :

Membres présents du jury de l'EURIA :

Philippe Lenca

Sonia Guélou

Directeur de mémoire en entreprise :

Raphaël Lagier

Signature :

Invite :

Signature :

***Autorisation de publication et de mise en ligne sur un site de diffusion de
documents actuariels
(après expiration de l'éventuel délai de confidentialité)***

Signature du responsable entreprise :

Signature du candidat :

Secrétariat :

Bibliothèque :

EURIA
EUro Institut
d'Actuariat

6, avenue le Gorgeu
CS 93837
29238 Brest Cedex 3

T +33 (0)2 98 01 66 55
F +33 (0)2 98 01 66 57
euria@univ-brest.fr

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué à la réalisation de ce mémoire.

Tout d'abord, je remercie Monsieur Piermay, président de Fixage, de m'avoir accueilli au sein de son entreprise, ainsi que pour la confiance qu'il m'a accordée.

Je remercie également mes tuteurs Raphaël Lagier, Quentin Merlet et Julien Paineau pour leurs conseils avisés.

J'adresse également ma gratitude aux différents collaborateurs de Fixage qui m'ont apporté leur aide tout au long de ce mémoire. En particulier Jérôme Contant, Faris Rouchati, Quentin Bazoge et Jean Wilhelm.

Je remercie mon tuteur EURIA, Monsieur Vincent Soulas pour son suivi régulier. Je pense également aux différents professeurs de l'EURIA et notamment à monsieur Franck Vermet directeur des études qui m'a apporté une aide précieuse.

Enfin, j'adresse une pensée particulière à ma compagne ainsi qu'à mes proches qui m'ont soutenu durant la rédaction de ce mémoire.

À la mémoire de Tom.

Résumé

Dans ce mémoire, nous nous intéressons au ratio de solvabilité d'un assureur retraite et à sa distribution à horizon un an. La distribution du ratio de solvabilité futur renseigne l'assureur sur l'évolution de sa solvabilité ainsi que sur son niveau de risque futur. Elle est aussi utilisée dans le cadre de l'ORSA, mais elle peut également permettre d'aiguiller un choix d'allocations d'actifs.

La méthode des Simulations dans les Simulations (SdS) est souvent utilisée par les assureurs pour projeter le ratio de solvabilité dans le futur, ou le bilan prudentiel, dans le cadre d'un calcul de SCR vision modèle interne par exemple. Toutefois, en imbriquant deux niveaux de simulations, elle s'avère très complexe à mettre en oeuvre.

Nous avons alors mis en place une méthode alternative basée sur des méthodes de *machine learning*, qui permet de se substituer aux différentes utilisations des SdS. Les méthodes de *machine learning* permettent d'apprendre une variable à expliquer à partir de variables explicatives sans être explicitement programmées. Cependant, elles présentent l'inconvénient de nécessiter une base de données d'apprentissage et supposent un lien entre les variables explicatives et la variable à expliquer. Nous avons développé des modèles de *machine learning* pour apprendre la provision *Best Estimate* future en fonction de l'environnement économique et de la part action en portefeuille.

Nous avons retenu trois modèles pour cette étude : le SVM, le Random Forest et le modèle XGBoost. Le premier est un modèle qui s'avère particulièrement performant lorsque le nombre de données dans la base d'apprentissage est élevé. Les deux autres sont des méthodes ensemblistes : il s'agit d'une agrégation de modèles simples (des arbres de décisions), permettant d'obtenir un modèle plus robuste.

Nous avons tout d'abord étudié la performance de ces modèles en vue de prédire le SCR vision modèle interne de l'assureur. Nous avons montré que le SVM s'avérait particulièrement efficace dans cette situation. Mais sa performance est moins stable lorsque la part action en portefeuille diminue. Le modèle XGBoost obtient également d'excellentes performances, avec une erreur de l'ordre de 1% pour l'estimation du SCR et il présente l'avantage d'obtenir des performances stables lorsque la part action est modifiée. De plus, il s'agit du modèle le plus rapide à exécuter.

Nous avons ensuite estimé la distribution du ratio de solvabilité à horizon un an. Nous sommes restés dans le cadre d'un calcul de SCR vision modèle interne. La comparaison de la distribution du ratio de solvabilité obtenu par notre méthode à la distribution théorique est plus complexe. Il s'agit alors d'effectuer des Simulations dans les Simulations dans les Simulations. XGBoost semble également le plus adapté à cette problématique et obtient une distribution du ratio de solvabilité futur très proche de la distribution de référence.

Nous avons ainsi mis en place une méthode robuste et capable de se substituer aux SdS dans des situations variées. Toutefois, les méthodes alternatives sont souvent difficiles à utiliser en pratique, faute de justification satisfaisante. C'est pourquoi, nous terminons ce mémoire avec une utilisation concrète de la méthode dans le cadre du choix d'une allocation d'actifs.

Mots clés : machine learning, XGBoost, SVM, Forêt aléatoire, Simulations dans les Simulations, SdS, SdSdS, modèle interne, ratio de solvabilité.

Abstract

In this paper, we are interested in the solvency ratio of a pension insurer and its distribution over a one-year horizon. The distribution of the future solvency ratio informs the insurer about the evolution of its solvency as well as its future level of risk. It is also used within ORSA, but it can also be used to refer a choice of asset allocation.

The Nested Simulation (NS) approach is often used by insurers to project the solvency ratio in the future, or the prudential balance sheet, in the context of a calculation of SCR vision internal model for example. However, by imbricating two levels of simulations it proves to be very complex to implement.

We then put in place an alternative method based on machine learning models, which can replace the different uses of the NS. Machine learning approach allows us to learn an explained variable from explanatory variables without being explicitly programmed. However, they have the disadvantage of requiring a learning database and assume a link between the explanatory variables and the variable to be explained. We have developed machine learning models to learn the Best Estimate provision depending on the economic environment and the equity share in the portfolio.

We selected three models for this study : the SVM, the Random Forest and the XGBoost model. The first is a model that proves particularly effective when the number of data in the learning base is high. The other two are set methods : it is an aggregation of simple models (decision trees), allowing to obtain a more robust model.

We first studied the performance of these models in order to predict the SCR with internal model of the insurer. We have shown that SVM is particularly effective in this situation. But its performance is less stable when the equity allocation of the portfolio falls. The XGBoost model also achieves excellent performance, with an error of the order of 1% for the estimation of the SCR calculation and it has the advantage of obtaining stable performances when the asset allocation is modified. In addition, this is the fastest model to run.

We then estimated the distribution of the solvency ratio to one year. We keep an internal model vision for the calculation of SCR. The comparison of the distribution of the solvency ratio obtained by our method with the theoretical distribution is more complex. It is then necessary to carry out Simulations in the Simulations in the Simulations (SdSdS). XGBoost also seems to be the most adapted to this problem and obtains a distribution of the future solvency ratio very close to the reference distribution.

We have thus put in place a robust method able to replace SdS in different situations. However, alternative methods are often difficult to use in practice because of a lack of justification. That is why we conclude this paper with a concrete use of the method in the choice of an asset allocation.

Key words : machine learning, XGBoost, SVM, Random Forest, Nested Simulations, SdS, SdSdS, Internal model, solvency ratio.

Note de synthèse

Dans ce mémoire, nous cherchons à projeter le ratio de solvabilité d'un assureur retraite afin d'obtenir une estimation de sa distribution à horizon un an. Celle-ci permet de renseigner l'assureur sur l'évolution de sa solvabilité dans le futur et son niveau d'exposition au risque. Un ratio de solvabilité élevé autorise davantage de marge de manoeuvre quant aux différents choix stratégiques de l'assureur.

L'allocation d'actifs de l'assureur a un impact fort sur le SCR et donc sur le ratio de solvabilité. Diminuer la part d'actifs risqués en portefeuille permet une hausse du ratio de solvabilité via une baisse du SCR. Mais cette hausse du ratio de solvabilité initial, peut se faire au détriment du rendement futur espéré. Nous nous donnons pour objectif d'obtenir une estimation de la distribution du ratio de solvabilité futur pour différentes allocations d'actifs.

La méthode des Simulations dans les Simulations permet de projeter le bilan prudentiel d'un assureur dans le futur. Il s'agit d'imbriquer deux niveaux de simulations : l'univers économique est projeté dans le futur. Conditionnellement aux nouvelles données de marché, de nouvelles projections sont effectuées afin d'évaluer la provision *Best Estimate*. Cette méthode est particulièrement complexe à mettre en oeuvre, mais elle permet de répondre à des problématiques variées : estimation de la distribution des Fonds Propres futurs, estimation de la distribution du ratio de solvabilité futur, calcul d'un SCR vision modèle interne, etc.

De nombreuses méthodes alternatives ont été proposées pour contourner cette difficulté technique. Cependant, ces méthodes ne permettent souvent de se substituer aux SdS que dans un cadre restreint : calcul de SCR vision modèle interne, ou projection du ratio de solvabilité et ne sont adaptées qu'au portefeuille de l'assureur.

Nous proposons une méthode alternative, en vue de répondre à notre problématique : obtenir rapidement une estimation de la distribution du ratio de solvabilité futur pour différentes allocations d'actifs. Mais, nous avons de plus souhaité mettre en place une méthode adaptée à la projection du ratio de solvabilité, mais également à un calcul de SCR vision modèle interne, ou, plus simplement à la projection d'un bilan prudentiel dans le futur.

Nous utilisons des modèles de *machine learning* afin de réduire la complexité opérationnelle des SdS. Les modèles de *machine learning* permettent d'apprendre une variable à expliquer en fonction de variables explicatives sans être explicitement programmés. Il a donc fallu choisir une variable à expliquer pertinente, permettant de réduire le coût lié à l'utilisation des SdS.

Nous avons proposé d'apprendre la provision *Best Estimate* future, en fonction de l'environnement économique futur et de la part action en portefeuille. Ce choix présente de nombreux avantages :

- il n'est que peu complexe d'obtenir différents environnements économiques futurs et les bilans prudentiels associés,
- la provision *Best Estimate* est probablement plus simple à apprendre que les Fonds Propres, le SCR ou le ratio de solvabilité,
- l'apprentissage de la provision *Best Estimate* autorise une utilisation de notre méthode pour de nombreuses problématiques actuarielles, là où d'autres méthodes alternatives ne sont adaptées qu'à un calcul de SCR vision modèle interne.

Nous avons alors décidé de rester dans le cadre du modèle interne pour le calcul du SCR. En se substituant au dernier niveau de simulation, notre méthode permet alors de réduire un problème de complexité N^3 en un problème de complexité N^2 selon le schéma suivant :

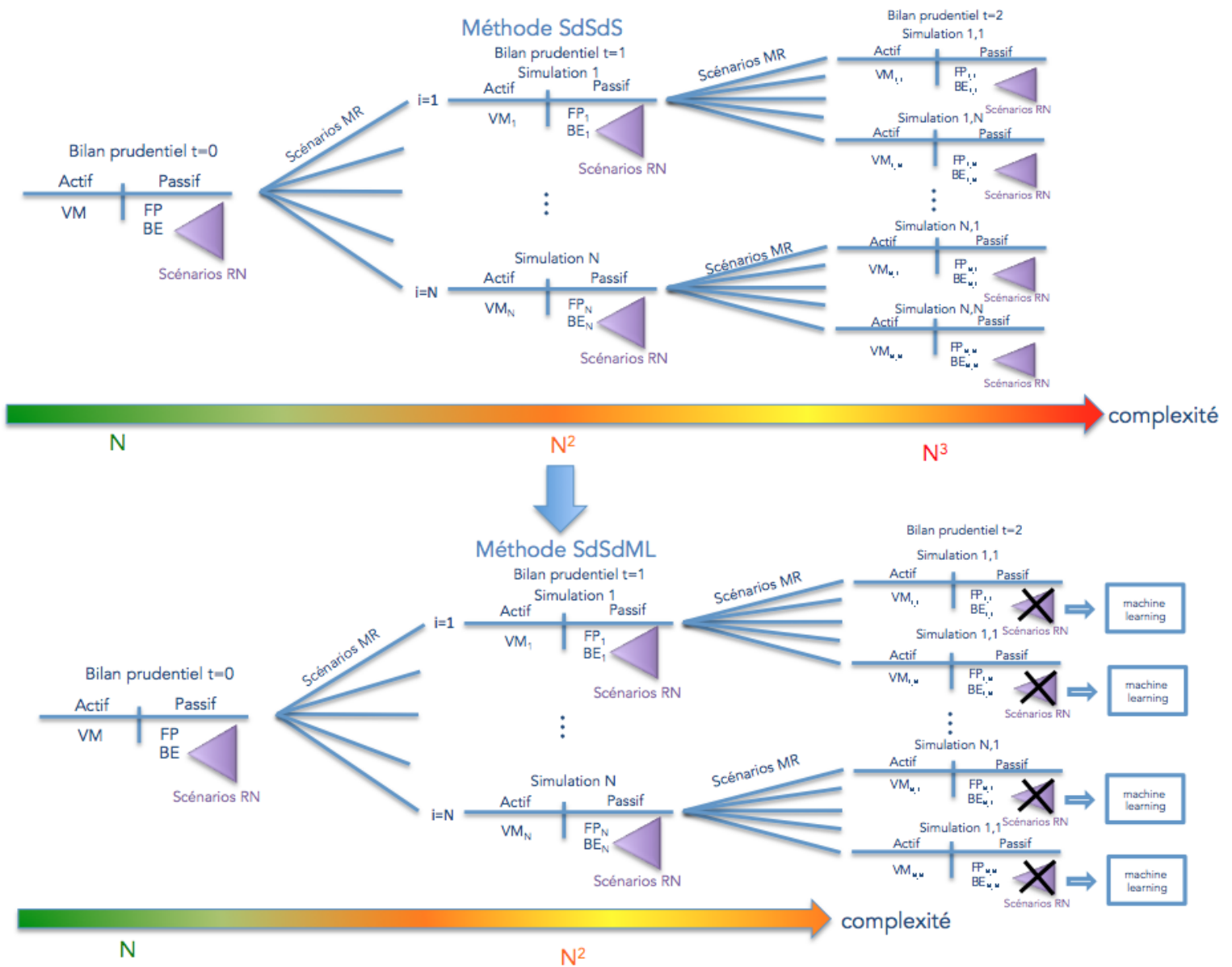


FIGURE 1 – La méthode SdSdML, alternative aux SdSdS

Une fois le choix de la variable à expliquer effectué, nous nous sommes intéressés au choix des modèles de *machine learning*. Nous avons tout d'abord retenu les *Support Vector Machine* (SVM). Il s'agit d'un modèle complexe permettant de modéliser de fortes interactions non linéaires entre les variables explicatives et la variable à expliquer via la construction d'un hyperplan optimal. La construction de l'hyperplan optimal se base sur un nombre restreint d'observations appelées « vecteurs supports ». Ainsi, en augmentant la taille de l'échantillon d'apprentissage, l'algorithme n'est pas ou peu impacté, c'est pourquoi il est souvent utilisé lorsque les bases de données sont très grandes.

Les méthodes ensemblistes ont également éveillé notre intérêt. Elles considèrent qu'il est possible de transformer un modèle simple tel que les arbres de décision en un modèle robuste capable de modéliser les liens les plus complexes entre les variables explicatives et la variable à expliquer. Les arbres de décision présentent l'inconvénient d'avoir un biais fort lorsqu'ils sont peu développés et une variance élevée lorsque l'on cherche à en diminuer le biais.

Les *Random Forest* contournent cette difficulté en agrégeant aléatoirement plusieurs arbres différents de faibles biais afin d'en diminuer la variance. Une autre approche consiste à chercher à diminuer le biais au fur et à mesure. Les arbres sont alors construits en série. Ce principe a été largement plébiscité récemment avec l'algorithme « XGBoost », particulièrement performant dans un cadre extrêmement varié de situations. « When in doubt, use xgboost » disait Owen Zhang, vainqueur d'une compétition Kaggle.

Nous avons toutefois souhaité tester les deux approches qui offrent davantage de possibilités de visualisation des résultats que les SVM. Nous avons alors comparé l'utilisation de notre méthode aux résultats théoriques. Nous avons ainsi cherché à déterminer :

- Si la méthode était adaptée à un calcul de SCR vision modèle interne. Nous nous sommes alors plus particulièrement intéressés aux performances des modèles en terme de prédiction de la queue gauche de la distribution des Fonds Propres à horizon un an.
- Si les différents modèles gardaient des performances homogènes lorsque l'on modifie la part action en portefeuille.
- Si la méthode était adaptée à la projection du ratio de solvabilité à horizon un an. Nous sommes restés dans le cadre d'une vision modèle interne pour le calcul du SCR, nous avons alors comparé les résultats de notre méthode à ceux obtenus par des Simulations dans les Simulations dans les Simulations (SdSdS).

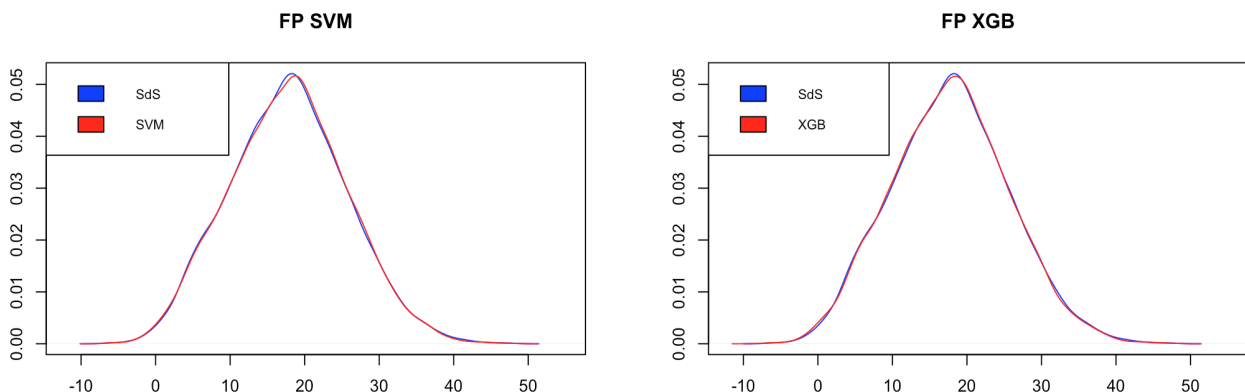
Nous avons tout d'abord comparé l'utilisation de notre méthode au calcul d'un SCR vision modèle interne d'un assureur retraite, investi à 30% en actions. Nous avons généré 10 000 environnements économiques à horizon un an et dressé les 10 000 bilans prudentiels correspondants. Nous calculons l'erreur quadratique moyenne (MSE) de nos différents modèles, ainsi que les SCR obtenus par les différents modèles que nous comparons au SCR théorique (14,89 M€). Nous obtenons les résultats suivants :

Modèle	MSE	SCR prédit	Écart relatif
XGB	0,81	15,06	1,11%
RF	0,98	15,52	4,20%
SVM	0,95	14,95	0,35%

TABLE 1 – Comparaison des différents modèles

XGBoost obtient ainsi les meilleures performances en terme d'erreur quadratique moyenne, mais s'avère malgré tout moins précis lors de l'estimation du SCR vision modèle interne que le SVM.

La distribution des Fonds Propres obtenus par XGBoost est ainsi la plus proche de la distribution théorique, à l'exception de la queue gauche de la distribution où le SVM s'avère plus précis :



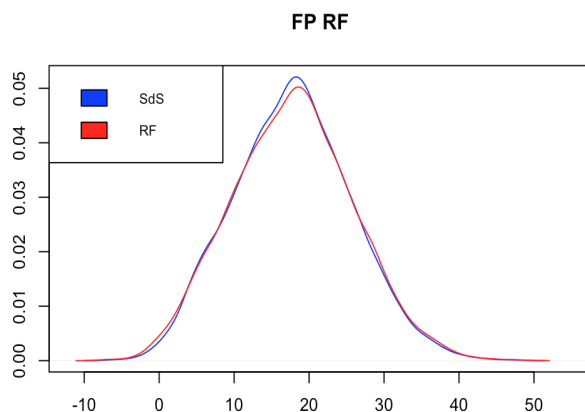


FIGURE 2 – Comparaison des densités obtenues par les modèles à la densité théorique (SdS)

Afin d'étudier les performances des modèles lorsque l'on modifie l'allocation d'actifs en portefeuille, nous effectuons une régression linéaire entre la MSE et la part action en portefeuille, sur une base de test composée de 5 100 environnements économiques en date $t=1$ avec des allocations d'actifs diversifiées.

Les résultats de cette régression linéaire indiquent que la performance des modèles est sensible à la part action en portefeuille et que les modèles sont plus performants lorsque la part action augmente.

Afin de vérifier la sensibilité des modèles à la part action en portefeuille, nous calculons le SCR vision modèle interne en modifiant l'allocation d'actifs. Nous calculons ainsi le SCR pour 20% et 40% action. XGBoost reste relativement stable en terme de MSE et de prédiction du SCR, avec un écart de prédiction de l'ordre de 1% par rapport au SCR théorique. À l'inverse le SVM est beaucoup moins stable et voit ses prédictions en terme de SCR se détériorer fortement lorsque la part action en portefeuille diminue. Enfin, le *Random Forest* semble être plus performant pour 20% actions.

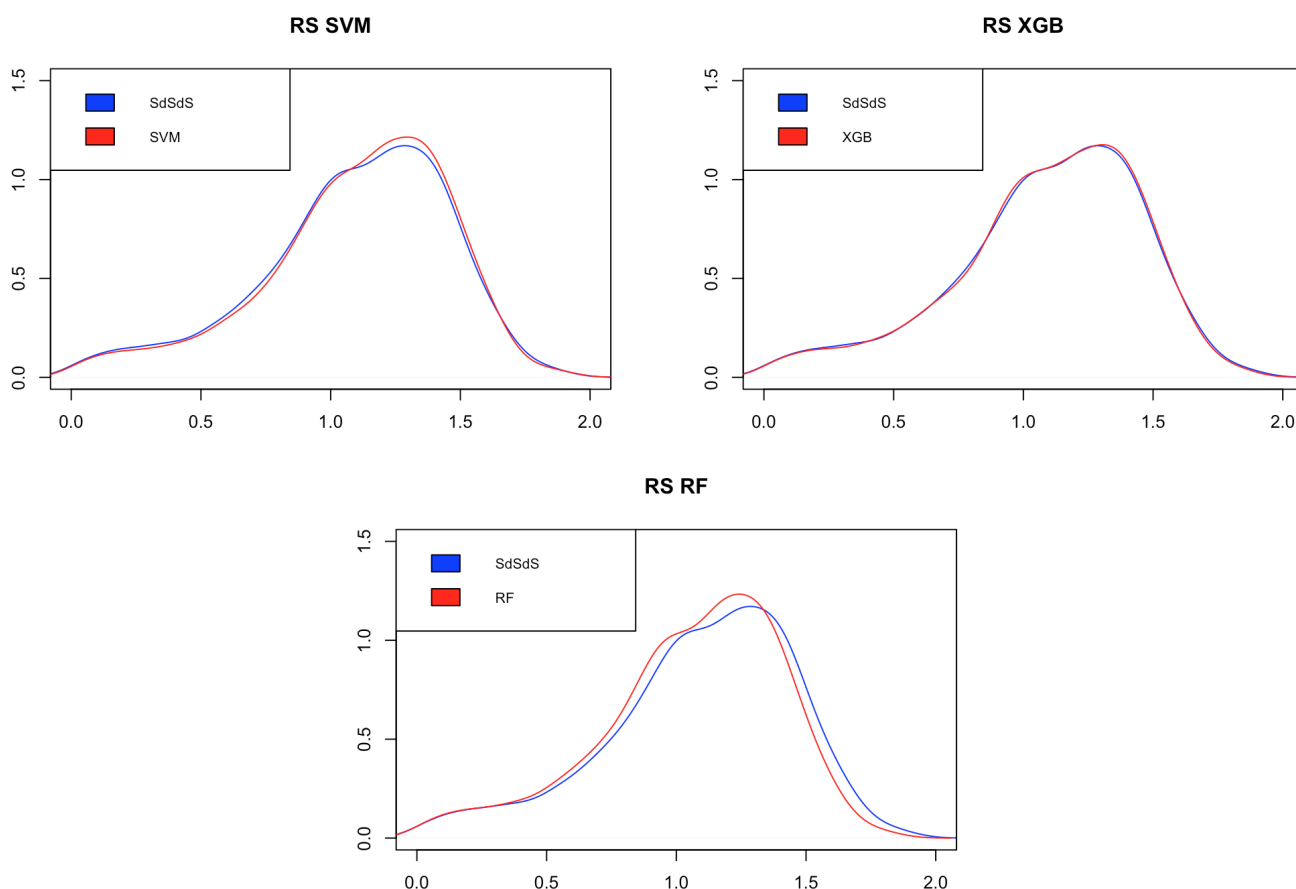
Ainsi, XGBoost n'est pas systématiquement le modèle le plus performant en terme de prédiction du SCR. Cependant, c'est le modèle qui obtient la MSE le plus faible et qui reste le plus stable lorsque la part action en portefeuille est modifiée. Cette capacité adaptative nous intéresse particulièrement pour notre étude, c'est pourquoi, il semble être le plus adapté à la résolution de notre problématique.

Nous devons ensuite nous assurer que notre méthode s'adapte bien à la projection du ratio de solvabilité à horizon un an. Nous utilisons la méthode des Simulations dans les Simulations pour obtenir une base de comparaison théorique. Cet exercice est particulièrement complexe et nous retenons l'utilisation de 1 000 scénarios monde réel entre $t=0$ et $t=1$, suivis de 1 000 scénarios monde réel entre $t=1$ et $t=2$, et 1 000 scénarios univers risque neutre pour le calcul des provisions *Best Estimate* à chaque pas de temps. Nous générons ainsi plus d'1 milliard de scénarios économiques, nécessitant une demi-semaine de calcul sur une machine particulièrement performante.

Notre méthode permet de réduire ce coût à 19 heures de calculs avec l'utilisation du modèle XGBoost, rendant possible la résolution de notre problématique.

Nous calculons l'écart quadratique moyen entre les provisions *Best Estimate* estimées par notre méthode et celles obtenues par les SdS à horizon deux ans. En ayant calculé 1 million de provisions *Best Estimate*, nous avons une bonne indication quant à la qualité de nos différents modèles. SVM et XGBoost obtiennent la meilleure MSE (0.90 et 0.99 respectivement, contre 1,05 pour RF).

Toutefois, XGBoost est le plus proche de la distribution théorique du ratio de solvabilité à horizon un an :



Nous retenons alors le modèle XGBoost pour estimer la distribution du ratio de solvabilité à horizon un an pour différentes allocations d'actifs. Nous projetons le ratio de solvabilité pour des allocations d'actifs allant de 20% à 40% actions avec un pas de 5%. Nous traçons sur un même graphique la performance du ratio de solvabilité à horizon un an en fonction de la volatilité du ratio de solvabilité à horizon un an :

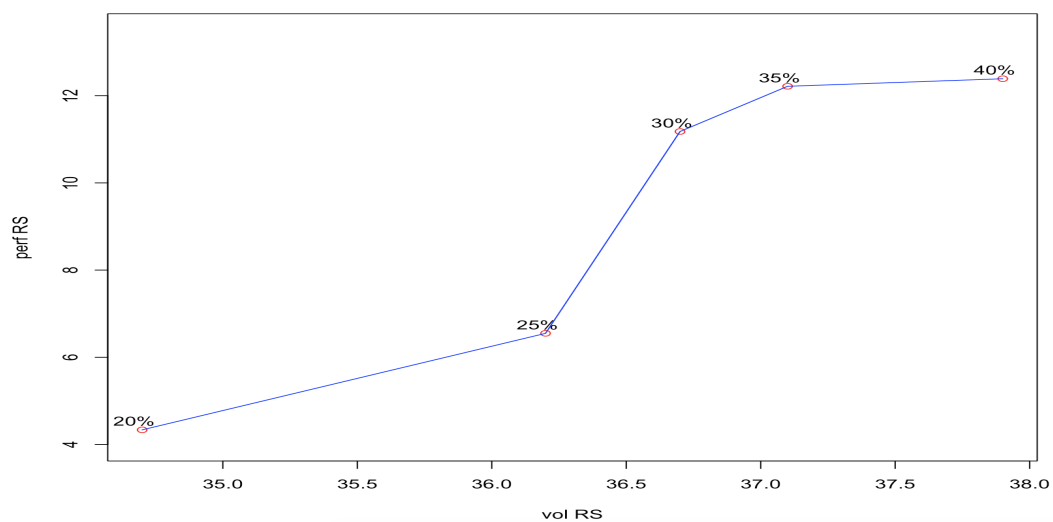


FIGURE 3 – Rendement du ratio de solvabilité à horizon un an en fonction de sa volatilité

Nous remarquons qu'une augmentation de la part action de 25% à 30% mène à une forte hausse du rendement attendu contre une augmentation relativement faible en terme de volatilité du ratio de

solvabilité futur. Ainsi, il semble judicieux de s'approcher d'une allocation d'actifs comprenant 30% d'actions si possible. Au delà, augmenter davantage sa part action amène à une forte augmentation du risque via une forte hausse de la volatilité du ratio de solvabilité futur, pour un rendement futur espéré très proche.

Nous avons ainsi mis en place dans ce mémoire une méthode robuste, capable de se substituer aux SdS pour un calcul de SCR vision modèle interne, mais également pour une projection du ratio de solvabilité, et ce, pour différentes allocations d'actifs. Nous avons alors proposé une application permettant d'éclairer un choix d'allocation d'actifs. En ce substituant aux SdSdS, notre méthode permet de passer de plus de deux semaines de calcul à quatre jours.

Il pourrait être intéressant de compléter ces résultats par une étude plus fine, avec un pas plus faible sur une tranche d'allocations d'actifs retenues. Il serait également judicieux de considérer d'autres critères ou avis d'experts pour aiguiller un choix d'allocations d'actifs.

Summary

In this brief, we attempt to project the solvency ratio of a pension insurer in order to obtain an estimate of its distribution over a one-year horizon. This one enables the insurer to be informed of the evolution of its solvency in the future and its level of risk exposure. A high solvency ratio allows more room for different strategic choices of the insurer.

The asset allocation of the insurer has a strong impact on the SCR and thus on the solvency ratio. Decreasing the proportion of risky assets in the portfolio allows an increase in the solvency ratio through a reduction in SCR. But this increase in the initial solvency ratio can be detrimental to the expected future return. We aim to obtain an estimate of the distribution of the future solvency ratio for various asset allocations.

The Nested Simulations approach (NS) allows to project the prudential balance of an insurer in the future. It consists in imbricating two levels of simulations : the economic universe is projected into the future. As a result of new market data, new projections are made to evaluate the Best Estimate provision. This method is particularly complex to implement, but it allows to answer to various problems : estimation of the distribution of the future Own Funds, estimation of the distribution of the future solvency ratio, calculation of a SCR using internal model, etc.

Many alternative methods have been proposed to circumvent this technical difficulty. However, these methods often do not allow to substitute for the SdS within a restricted framework : calculation of SCR using internal model, or projection of the solvency ratio and are only adapted to the portfolio of the insurer.

To order to answer our problem, we propose an alternative method, in order to answer our problem : to quickly obtain an estimate of the distribution of the future solvency ratio for various asset allocations. However, we also wanted to set up a method adapted to the projection of the solvency ratio, but also to a calculation of SCR using internal model, or, more simply to the projection of a prudential balance sheet in the future.

We use machine learning algorithms to reduce the operational complexity of the NS. Machine learning approach allows us to learn an explained variable from explanatory variables without being explicitly programmed. It was thus necessary to choose wisely the explained variable, allowing to reduce the cost linked to the use of the NS.

We proposed to learn the future Best Estimate provision, depending on the future economic environment and the equity allocation in the portfolio. This choice has many advantages :

- There is little complexity in obtaining different future economic environments and associated prudential balances sheet,
- The Best Estimate provision is probably simpler to learn than the Own Funds, the SCR or the solvency ratio,
- Learning the Best Estimate provision allows us to use our method for many actuarial problems, where other alternative methods are only adapted to a calculation of an SCR using internal model.

We then decided to keep internal model for the calculation of the SCR. By replacing the last level of simulation, our method then makes it possible to reduce a problem of complexity N^3 into a problem of complexity N^2 according to the following scheme :

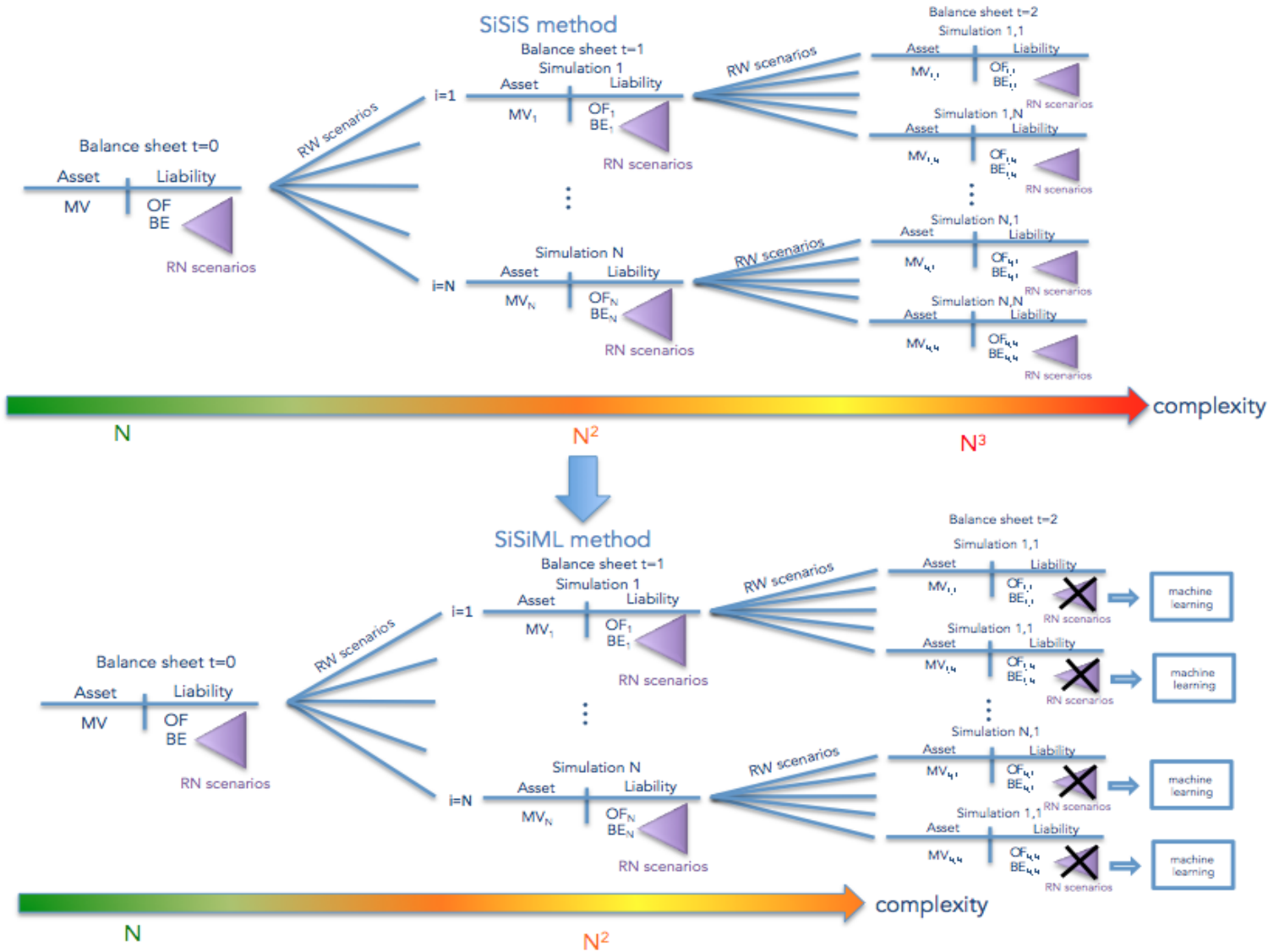


FIGURE 4 – La méthode SdSdML, alternative aux SdSdS

Once the choice of the explained variable was made, we were interested in the choice of machine learning models. We first used Support Vector Machine (SVM). It is a complex model allowing to model strong non-linear interactions between the explanatory variables and the explained variable by constructing an optimal hyperplane. The construction of the optimal hyperplane is based on a limited number of observations called support vectors. Thus, by increasing the size of the learning sample, the algorithm is not or little impacted, which is why it is often used when databases are very large.

Set methods have also aroused our interest. They consider that it is possible to transform a simple model such as decision trees into a robust model capable of modeling the most complex links between the explanatory variables and the explained variable. Decision trees have the disadvantage of having a strong bias when they are not very developed and a high variance when we try to reduce the bias.

The Random Forest bypass this difficulty by randomly aggregating several low-bias trees to reduce their variance. Another approach is to try to reduce the bias gradually. The trees are then built in series. This principle has been widely acclaimed recently with the XGBoost algorithm which is particularly effective in a very varied set of situations. « When in doubt, use xgboost » said Owen Zhang,

winner of a Kaggle competition.

However, we chose to test the two approaches which offer more possibilities of result visualization than the SVM. We then compared the use of our method with the theoretical results. We have thus sought to determine :

- If the method was adapted to a SCR calculation using internal model. We were particularly interested in model performance in terms of predicting the left tail of the distribution of the Own Funds over a one-year horizon.
- If the different models kept their performances consistent when the equity allocation in the portfolio was changed.
- If the method was adapted to the projection of the solvency ratio to one year. We kept an internal model vision for the calculation of the SCR, we then compared the results of our method with those obtained by Simulations in the Simulations in the Simulations (SiSiS).

We first compared the use of our method to calculate a SCR using internal model of a retirement insurer, invested 30% in equities. We generated 10,000 economic environments over a one-year horizon and calculate the 10,000 corresponding prudential balances. We compute the mean square error (MSE) of our different models, as well as the SCR obtained by the different models that we compare to the theoretical SCR (14.89 M€). We obtain the following results :

Model	MSE	predicted SCR	relative gap
XGB	0,81	15,06	1,11%
RF	0,98	15.52	4,20%
SVM	0,95	14,95	0,35%

TABLE 2 – Comparison of our different models

XGBoost thus obtains the best performances in terms of mean square error, but is nevertheless less precise when estimating the SCR using internal model than the SVM.

The distribution of the own funds estimated by XGBoost is thus closest to the theoretical distribution, with the exception of the left tail of the distribution where the SVM is more precise :

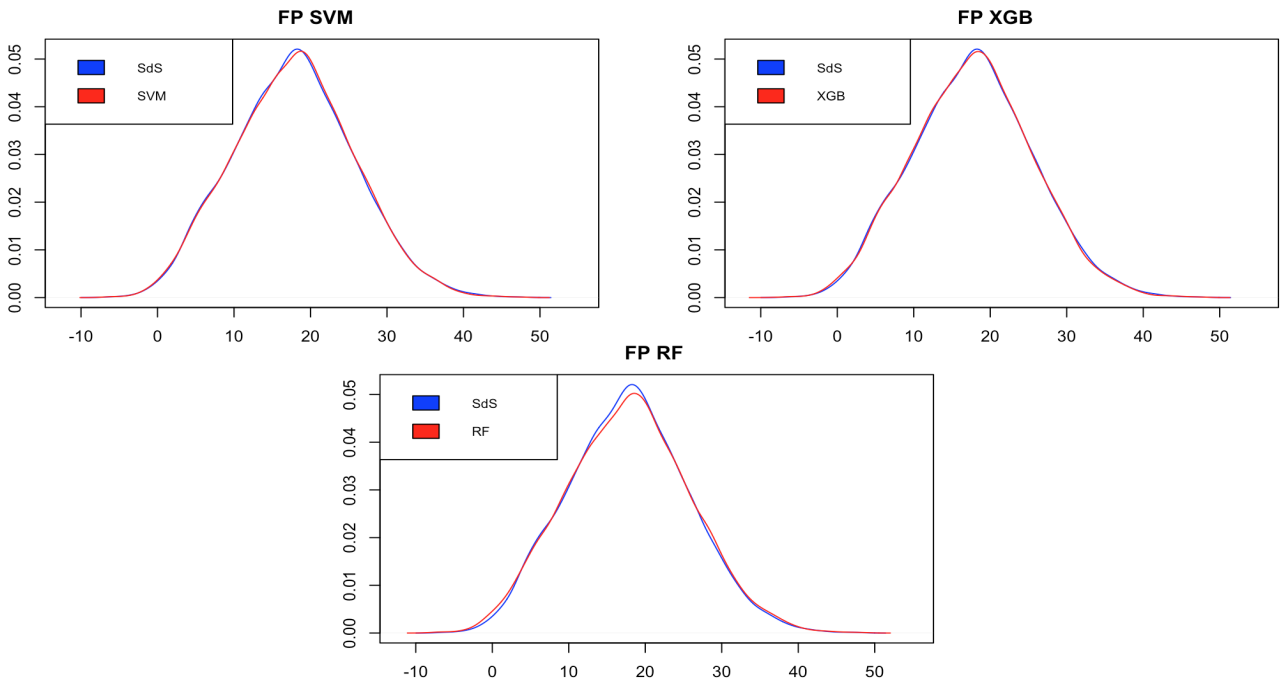


FIGURE 5 – Comparison of the densities obtained by the models with the theoretical density (NS)

In order to study models performance when modifying the asset allocation in the portfolio, we perform a linear regression between the MSE and the equity allocation in portfolio on a test basis composed of 5,100 economic environments $t = 1$ with diversified asset allocations.

The results of this linear regression show that the performance of the models is sensitive to the equity allocation in the portfolio and that the models are more efficient when the equity allocation is high.

In order to verify the sensitivity of the models to the equity allocation in the portfolio, we calculate the SCR using internal model vision by modifying the asset allocation. We thus compute the SCR for 20% and 40% equity in portfolio. XGBoost remains relatively stable in terms of MSE and SCR prediction, with a prediction deviation of the order of 1% relative to the theoretical SCR. Conversely, the SVM is much less stable and sees its predictions in terms of SCR deteriorating sharply when the equity allocation in the portfolio decreases. Finally, Random Forest seems to be performing better for 20% equity allocation.

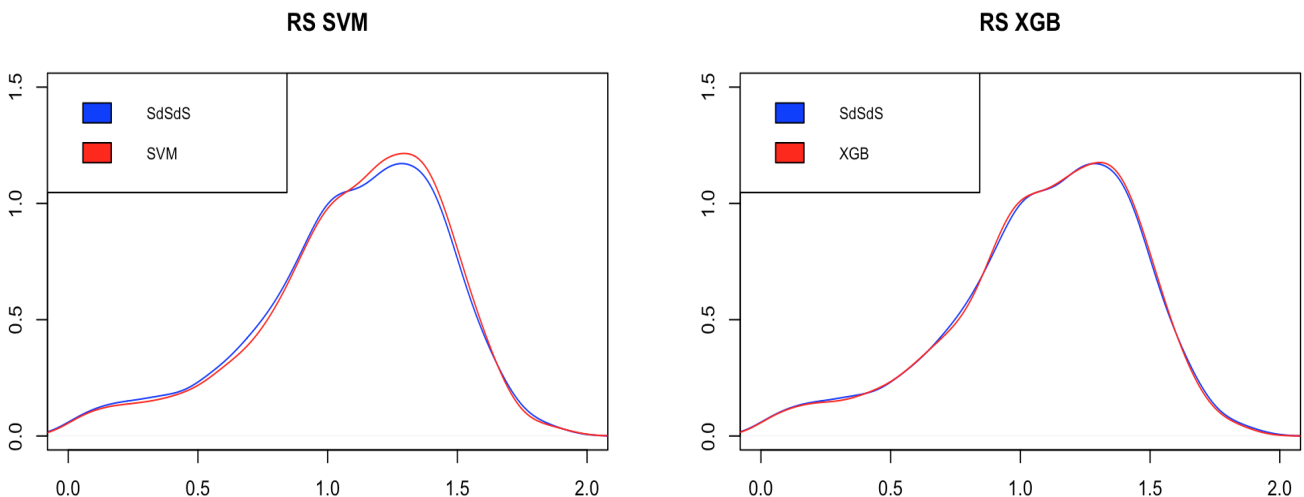
Thus, XGBoost is not always the most efficient model in terms of SCR prediction. However, it is the model that obtains the weakest MSE and remains the most stable when the equity allocation in the portfolio is modified. This adaptive capacity is really important for our study, which is why it seems to be the most suitable model for solving our problem.

We then have to ensure that our method is well suited to the projection of the solvency ratio to one year. We use the Simulations in Simulations in Simulations method to obtain a theoretical comparison basis. This exercise is particularly complex and we retain the use of 1,000 real world scenarios between $t = 0$ and $t = 1$, followed by 1,000 real world scenarios between $t = 1$ and $t = 2$, And 1,000 risk neutral scenarios for the calculation of Best Estimate provisions at each time step. We thus generate more than 1 billion economic scenarios, requiring half a week of calculation on a particularly performing machine.

Our method reduces this cost to 19 hours of calculations with the use of the XGBoost model, making it possible to solve our problem.

We calculate the mean square error between the Best Estimate provisions estimated by our method and those obtained by the SiSiS approach over two years. While having calculated 1 million Best Estimate provisions, we have a good indication as to the quality of our different models. SVM and XGBoost obtained the best MSE (0.90 and 0.99 respectively, compared to 1.05 for RF).

However, XGBoost is closest to the theoretical distribution of the solvency ratio to one year :



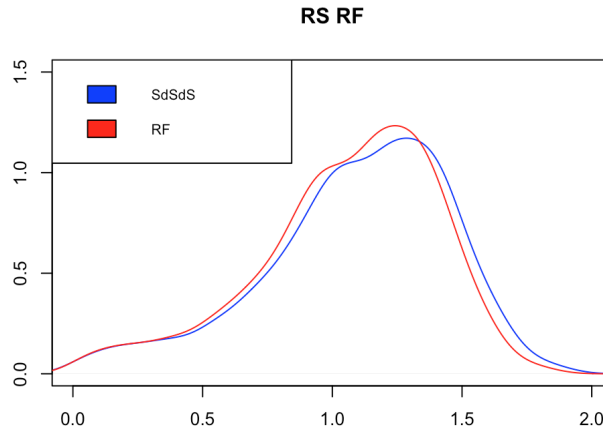


FIGURE 6 – Comparison of the distributions of SR obtained by the models to the theoretical distribution

We then use the XGBoost model to estimate the distribution of the solvency ratio at one year for different asset allocations. We project the solvency ratio for asset allocations ranging from 20% to 40% equity with a step of 5%. We draw on the same graph the performance of the solvency ratio to one year horizon according to the volatility of the solvency ratio to one year :

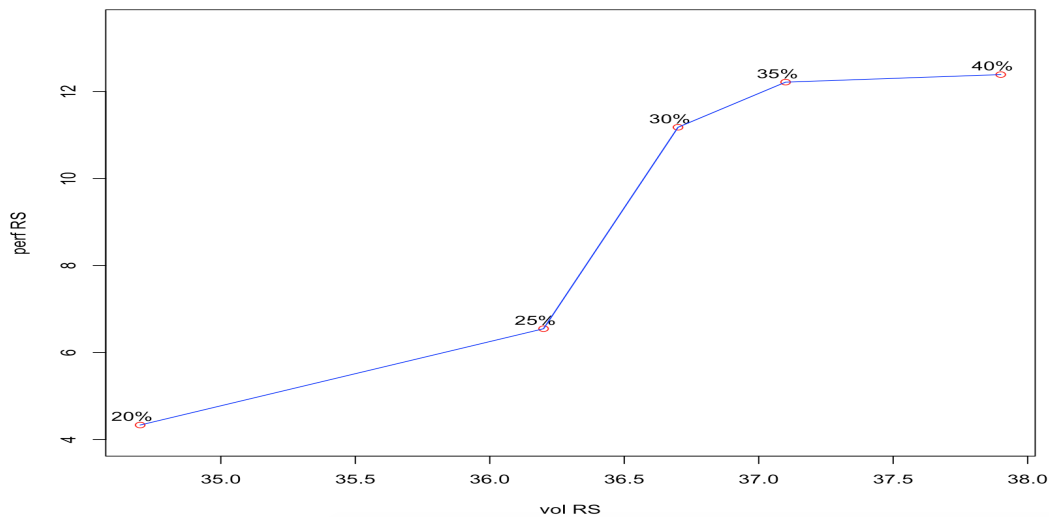


FIGURE 7 – Solvency ratio returns to one year based on volatility

We note that an increase in the equity part from 25% to 30% leads to a strong increase in the expected return against a relatively low increase in volatility of the future solvency ratio. Thus, it seems sensible to approach an asset allocation comprising 30% equity if possible. Beyond that, to increase the equity part leads to a strong increase of the risk via a strong increase of the volatility of the future solvency ratio, for a very low return expected.

In this paper, we have developed a robust method, capable of substituting the NS for a SCR calculation using internal model but also for a projection of the solvency ratio for different asset allocations. We then proposed an application to enlight a choice of asset allocation. By substituting for the SiSiS, our method allows us to spend more than two weeks of calculation at four days.

It may be worthwhile to supplement these results with a finer study, with a smaller step on a tranche of allocated asset allocations. It would also be wise to consider other criteria or expert advice to guide a choice of asset allocation.

Table des matières

Remerciements	i
Résumé	ii
Abstract	iii
Note de synthèse	ix
Summary	xiv
Introduction	1
1 L'estimation du ratio de solvabilité	2
1.1 Pourquoi s'intéresser au ratio de solvabilité?	2
1.2 La valorisation du bilan prudentiel	3
1.2.1 La valorisation des éléments d'actif	4
1.2.2 La valorisation du passif	5
1.3 Le Générateur de Scénarios Économiques (GSE)	8
1.3.1 Les corrélations entre les différentes variables financières	8
1.3.2 Présentation des modèles de diffusion	9
1.3.3 Le test de Martingalité	11
1.3.4 Le test de <i>Market-Consistency</i>	12
1.4 Le calcul du SCR	12
1.4.1 Le SCR en formule standard	12
1.4.2 Le SCR en modèle interne	13
2 La méthode des Simulations dans les Simulations : limites et alternatives	14
2.1 La méthode des SdS	14
2.1.1 Présentation de la méthode des SdS	14
2.1.2 La valorisation du bilan prudentiel futur	15
2.1.3 Les utilisations classiques de la méthode des SdS :	19
2.1.4 Le choix du nombre de scénarios primaires	22
2.1.5 La problématique du temps de calcul	22
2.2 Les alternatives usuelles à la méthode des SdS	23
2.2.1 Les approches paramétriques	23
2.2.2 Les <i>Replicating Portfolios</i>	23
2.2.3 Les avantages et inconvénients de ces méthodes	24
3 Le machine learning : alternative à la méthode des SdS	25
3.1 Le machine learning	25
3.1.1 Qu'est ce que le <i>machine learning</i> ?	25
3.1.2 L'apprentissage supervisé	26
3.1.3 Le dilemme biais-variance	26
3.1.4 Le sur-apprentissage	28

3.1.5	Utilisation de la validation croisée dans les méthodes d'apprentissage supervisé	30
3.1.6	Quelles méthodes d'apprentissage choisir ?	32
3.2	Présentation des méthodes d'apprentissage supervisé retenues	33
3.2.1	Les arbres de décision à la base des méthodes ensemblistes	33
3.2.2	Les <i>Random Forest</i>	36
3.2.3	Les méthodes de <i>Boosting</i>	37
3.2.4	Les SVM	41
3.3	Le machine learning comme alternative aux SdS	50
3.3.1	Présentation de la méthode alternative aux SdS	50
3.3.2	La construction des bases d'apprentissage et de test	54
3.3.3	Inférence statistique et validation des modèles	58
4	Application de la méthode à un contrat retraite	61
4.1	Présentation du contrat	61
4.1.1	Les engagements de l'assureur au 31/12/2016	61
4.1.2	Le bilan initial de l'assureur	62
4.2	Utilisation de la méthode pour calculer un SCR vision modèle interne	63
4.2.1	Apprentissage et calibrage des modèles	63
4.2.2	Comparaison des différents modèles	65
4.2.3	Le calcul du SCR vision modèle interne	70
4.2.4	La performance de la méthode en terme de temps de calcul	77
4.3	Utilisation de la méthode pour projeter le ratio de solvabilité à horizon un an	78
4.3.1	Calibrage des modèles	78
4.3.2	Comparaison des modèles	79
4.3.3	Étude préliminaire sur le bilan de l'assureur et rappels de la méthode	81
4.3.4	Application de la méthode à la projection du ratio de solvabilité à horizon un an	84
4.4	Application de la méthode au choix d'une allocation d'actifs	86
	Conclusion	90
	Liste des abréviations, des sigles et des symboles	92
	Annexes	95
	Bibliographie	97

Table des figures

1	La méthode SdSdML, alternative aux SdSdS	v
2	Comparaison des densités obtenues par les modèles à la densité théorique (SdS)	vii
3	Rendement du ratio de solvabilité à horizon un an en fonction de sa volatilité	viii
4	La méthode SdSdML, alternative aux SdSdS	xi
5	Comparison of the densities obtained by the models with the theoretical density (NS) .	xii
6	Comparison of the distributions of SR obtained by the models to the theoretical distribution	xiv
7	Solvency ratio returns to one year based on volatility	xiv
1.1	Bilan Prudentiel	4
1.2	Bilan Prudentiel Simplifié	7
1.3	Pieuvre SCR	13
2.1	La méthode des SdS	14
2.2	Flux d'une obligation	17
2.3	La méthode des SdS pour le calcul du SCR	19
2.4	La méthode des SdS pour l'estimation de la distribution du ratio de solvabilité à horizon un an	21
3.1	Représentation du biais et de la variance d'un modèle	27
3.2	Le sur-apprentissage ou « <i>over-fitting</i> »	28
3.3	Le compromis biais-variance [15]	29
3.4	Méthode LOOCV	31
3.5	Exemple d'arbre de décision pour des données quantitatives	34
3.6	Exemple d'hyperplans permettant de classifier des données	42
3.7	Hyperplan optimal	43
3.8	Exemples de marge floue	45
3.9	Influence du paramètre C dans le choix de l'hyperplan séparateur	46
3.10	Astuce du noyau	47
3.11	Le SVR en dimension 2	48
3.12	Le SVR en dimension 2	49
3.13	Le <i>kernel trick</i> pour un problème de régression	49
3.14	La méthode SdSdML, alternative aux SdSdS	53
3.15	Construction des bases d'apprentissage et de test à horizon un an	55
3.16	Construction des bases d'apprentissage et de test à horizon 2 ans	56
3.17	Vitesse de convergence du SCR	59
4.1	Échéancier des flux de prestations non revalorisées vu au 31/12/2016	61
4.2	Bilan Prudentiel de l'assureur au 31/12/2016	62
4.3	Variables explicatives les plus pertinentes pour les modèles XGBoost et RF	64
4.4	Comparaison des densités reconstituées des BE	66
4.5	Comparaison de la répartition des erreurs commises par les différents modèles	67
4.6	Part action des scénarios les plus défavorables aux modèles	68
4.7	distribution des Fonds Propres et <i>Best Estimate</i> dans un an	70
4.8	Comparaison des provisions <i>Best Estimate</i> obtenues	72

4.9	qq-plot modèles vs SdS	73
4.10	Comparaison des densités obtenues par les modèles à la densité obtenue par les SdS . .	74
4.11	Comparaison des densités obtenues par les modèles à la densité obtenue par les SdS Zoom sur la queue gauche de la distribution	74
4.12	Comparaison des densités obtenues par les modèles à la densité obtenue par les SdS Zoom sur la queue gauche de la distribution	75
4.13	Importance relative des variables explicatives pour les modèles XGBoost et RF	78
4.14	qq-plot entre l'erreur commise par les différents modèles	80
4.15	Comparaison des densités des BE obtenus à horizon 2 ans par les modèles à la densité théorique	80
4.16	La méthode des SdSdS pour l'estimation de la distribution du ratio de solvabilité à horizon un an	81
4.17	La méthode SdSdML	82
4.18	Distribution du SCR et du RS à horizon un an	83
4.19	Comparaison des distributions des SCR obtenus par les modèles à la distribution théorique	85
4.20	Comparaison des distributions des RS obtenus par les modèles à la distribution théorique	86
4.21	Rendement du ratio de solvabilité à horizon un an en fonction de sa volatilité	87
4.22	Exemple de qq-plot	94
4.23	qq-plot Fonds Propres modèle vs SdS	95

Liste des tableaux

1	Comparaison des différents modèles	vi
2	Comparison of our different models	xii
1.1	Matrice de corrélation	8
4.1	La politique de revalorisation de l'assureur	62
4.2	Performance des modèles sur $Base_{test-1}^{t=1}$	65
4.3	Répartition des erreurs entre les différents modèles	66
4.4	erreur en fonction de la tranche action	69
4.5	Régression linéaire erreur en fonction de la part action en portefeuille	69
4.6	Comparaison des différents modèles	71
4.7	Résultats des modèles sur les parts 20%, 30% et 40% action	76
4.8	Répartition des erreurs entre les différents modèles	79
4.9	MSE des différents modèles à horizon deux ans	84
4.10	écart type et moyenne du SCR à horizon un an	84
4.11	écart type et moyenne du RS à horizon un an	85
4.12	Ratio de solvabilité futur pour différentes allocations d'actifs	87

Introduction

En assurance vie, les évolutions de l'actif et du passif du bilan sont liées. Le taux de revalorisation annuel des prestations dépend notamment du rendement de l'actif et le besoin en capital réglementaire (SCR¹) est également fonction du choix de l'allocation d'actifs. Les assureurs doivent alors tenir compte de l'impact du choix de leur allocation d'actifs sur leur bilan et sur leur SCR.

Un indicateur adapté à cette problématique est le ratio de solvabilité. Il est défini par le rapport des Fonds Propres au sens Solvabilité II sur le SCR. Un ratio de solvabilité élevé permet davantage d'amplitude quant aux différents choix stratégiques de l'assureur.

Le choix d'une allocation d'actifs doit également tenir compte de la solvabilité future de l'assureur. Une augmentation de la part d'actif risqué peut en effet se traduire par une hausse immédiate du SCR, mais être compensée à plus long terme par une augmentation des Fonds Propres via un rendement plus élevé de l'actif. Nous cherchons alors à obtenir une estimation de la distribution du ratio de solvabilité futur afin d'intégrer ce phénomène.

Les assureurs utilisent souvent la méthode des Simulations dans les Simulations (SdS) pour estimer la distribution du ratio de solvabilité futur. Cette méthode est également utilisée dans le cadre d'un calcul de SCR vision modèle interne. Elle présente cependant l'inconvénient d'être particulièrement complexe à mettre en oeuvre et de nombreuses alternatives ont déjà été proposées. Dans ce mémoire, nous allons proposer une approche alternative basée sur du *machine learning*.

L'apparition de grandes bases de données via internet et l'augmentation des capacités informatiques donne un nouveau souffle à cette discipline. Si les banques s'étaient déjà intéressées aux possibilités offertes par le *machine learning* via le scoring, les assureurs semblent commencer à franchir le pas. De nombreux assureurs s'essayaient en effet à des méthodes de tarification alternatives basées sur des modèles de *machine learning*. Mais les possibilités offertes par les méthodes de *machine learning* sont très nombreuses et l'idée d'adapter ces modèles à la méthode des SdS est particulièrement séduisante.

Malgré un engouement manifeste pour cette approche, les résultats obtenus sont souvent limités. Les méthodes alternatives proposées le sont généralement dans un cadre spécifique et il est difficile d'envisager une généralisation des modèles. Les résultats obtenus en terme de prévision du SCR (et a fortiori du ratio de solvabilité) sont souvent peu satisfaisants lorsque celui-ci est obtenu via la formule standard. À l'inverse, lorsque le SCR est obtenu par une approche de type modèle interne, la méthode n'est pas adaptée à la projection du ratio de solvabilité. Enfin, la justification des méthodes de *machine learning* peut s'avérer délicate. Il est pourtant difficilement envisageable de proposer une méthode qui ne trouverait aucune application pratique faute de justification.

Dans ce mémoire, nous allons tenter de répondre à ces différentes questions. Nous chercherons à obtenir une méthode adaptable aux différentes problématiques actuarielles nécessitant le recours à la méthode des SdS. Nous nous interrogerons également sur le choix des méthodes de *machine learning* à privilégier. Enfin, nous proposerons une application de la méthode pour un choix d'allocation d'actifs.

1. Le SCR est le *Solvency Capital Requirement* : il s'agit de l'exigence réglementaire de capital dont doit disposer l'assureur ; nous reviendrons sur son calcul dans les parties suivantes.

Chapitre 1

L'estimation du ratio de solvabilité

« On fait la science avec des faits, comme on fait une maison avec des pierres ; mais une accumulation de faits n'est pas plus une science qu'un tas de pierres n'est une maison. »

Henri Poincaré

1.1 Pourquoi s'intéresser au ratio de solvabilité ?

En 1952, Markowitz publie un article [22] dans lequel il introduit le paradigme Espérance/Variance et le fondement de la théorie moderne du portefeuille. Ce paradigme suppose une aversion au risque de la part des investisseurs. Ainsi, à rendement identique un investisseur préférera un risque plus faible. Le rendement est mesuré par le rendement espéré du portefeuille et le risque par la volatilité du portefeuille.

Cette approche ne tient compte que de l'actif. Elle est réductrice et mal adaptée à l'assurance. En assurance vie, les évolutions de l'actif et du passif du bilan sont liées. Pour un contrat retraite, le taux de revalorisation annuel des prestations dépend notamment du rendement de l'actif. Il est alors plus judicieux de s'intéresser aux Fonds Propres qui vont prendre en compte l'actif et le passif.

Cependant, l'allocation d'actifs et le choix des classes d'actifs (obligation d'état européen/non européen, obligation corporate, action, immobilier, etc) impactent également le besoin en capital de l'assureur. Il faut en tenir compte lors du choix de l'allocation d'actifs. En effet, seule la part des Fonds Propres excédant la couverture du SCR est réellement disponible pour l'assureur.

Le ratio de solvabilité, défini comme le rapport des Fonds Propres sur le SCR, tient compte de ces différentes problématiques. Un ratio de solvabilité égal à 1 signifie que les Fonds Propres couvrent le SCR, sans surplus. Plus le ratio de solvabilité est élevé, plus il autorise une marge de manoeuvre élevée à l'assureur. Il peut alors envisager de prendre davantage de risque (et donc avoir un rendement futur espéré supérieur), avoir une politique commerciale plus agressive, faire davantage d'investissements, etc. Dans ces conditions, il semble intéressant de vouloir maximiser le ratio de solvabilité.

Toutefois, un changement d'allocation d'actifs peut avoir un impact à horizon plus éloigné que le ratio de solvabilité aura du mal à mesurer. Par exemple, augmenter la part action d'un portefeuille¹ va se traduire par une hausse immédiate du SCR. Les Fonds Propres ne seront quant à eux potentiellement que peu modifiés. Le ratio de solvabilité va alors diminuer en augmentant la part action. Si le ratio de solvabilité reste suffisamment élevé, il ne faut pas en conclure qu'augmenter la part action est un mauvais choix. En effet, l'espérance de rendement d'un actif risqué, tel que les actions, est supérieure à celle d'un actif moins risqué. Le ratio de solvabilité va prendre en compte le changement de niveau de risque de façon « immédiate » via une hausse de SCR, mais ne sera pas à même d'apprécier la hausse

1. Le SCR bénéficie d'un effet de diversification donc une augmentation de la part action peut se traduire par une baisse du SCR, si le portefeuille n'était que peu ou pas diversifié. Nous considérons ici un portefeuille ayant déjà une part action suffisante pour bénéficier de l'effet diversification du SCR.

du rendement futur espéré obtenu par le changement d'allocation d'actifs. Il est pourtant crucial pour un assureur de se positionner à long terme et de tenir compte de cette différence de rendement espéré.

Il peut ainsi être intéressant pour un assureur de choisir une allocation d'actif ne maximisant pas le ratio de solvabilité initial, mais amenant à des rendements futurs espérés plus élevés. Nous pouvons alors regarder la progression de l'actif et de sa volatilité estimée à un horizon futur, en conservant en parallèle l'information du ratio de solvabilité initial. Nous avons ainsi une vision de l'impact de l'allocation d'actifs en terme de rendement futur espéré, de risque (via la volatilité future de l'actif) et de besoin en capital/solvabilité de l'assureur (via le ratio de solvabilité initial).

Une autre approche consiste à estimer la distribution du ratio de solvabilité futur. Le ratio de solvabilité futur permet d'apprécier à la fois l'impact de l'allocation d'actifs sur le passif, sur le besoin en capital et sur le rendement futur espéré. La distribution du ratio de solvabilité futur nous permet également d'évaluer le ratio de solvabilité moyen et donc d'apprécier la « performance » de notre portefeuille. La volatilité du ratio de solvabilité ou la VaR (à 5% par exemple) nous renseignent sur le niveau de risque de notre portefeuille.

En estimant la distribution du ratio de solvabilité futur pour différentes allocations d'actifs, nous pourrions proposer un critère permettant d'aiguiller un choix d'allocation d'actifs. Par exemple, nous pourrions chercher à déterminer l'allocation d'actif maximisant le ratio de solvabilité futur espéré, sous contrainte d'être supérieur à un certain seuil dans les $x\%$ cas les plus défavorables. Le choix de ces seuils incombant aux instances de direction.

Cette approche est cependant plus complexe à mettre en oeuvre et il semble plus difficile de l'utiliser dans le cadre du choix d'une allocation d'actifs. Néanmoins, le ratio de solvabilité et l'estimation de la distribution du ratio de solvabilité futur d'un portefeuille apportent de nombreuses informations indépendamment de la problématique du choix d'une allocation d'actifs, c'est pourquoi nous allons nous y intéresser tout au long de ce mémoire.

Notons toutefois que le ratio de solvabilité seul ne peut suffire à piloter l'allocation d'actifs d'un assureur. En cherchant à maximiser leur rendement sous contrainte de SCR certains assureurs ont par exemple privilégié les obligations grecques peu avant leur défaut. Nous proposons donc l'étude d'un indicateur intéressant pour l'assureur, mais qu'il conviendrait de compléter par d'autres indicateurs.

Après avoir mis en lumière les intérêts et limites du ratio de solvabilité initial et de l'estimation de la distribution du ratio de solvabilité futur, nous allons présenter les éléments nécessaires au calcul du ratio de solvabilité. Nous présenterons au chapitre 2 une méthode souvent utilisée pour estimer une distribution du ratio de solvabilité et les limites opérationnelles liées à un tel exercice. Au chapitre 3, nous proposerons une méthode alternative permettant de contourner cette complexité. Enfin, au chapitre 4, nous appliquerons notre méthode sur un contrat retraite. Nous chercherons tout d'abord à calculer le SCR vision modèle interne par notre méthode. Puis, nous appliquerons la méthode à la projection du ratio de solvabilité à horizon un an de l'assureur. Nous pourrions alors déterminer l'allocation d'actifs maximisant le rendement futur espéré sous contrainte de risque.

1.2 La valorisation du bilan prudentiel

Le bilan prudentiel est le bilan établi selon la norme Solvabilité II.

Le calcul des provisions techniques du bilan prudentiel est régi par certaines règles. La directive 2009/138/UE en fixe les principes généraux. Ces principes ont été complétés par la Commission Européenne avec les règlements délégués et l'EIOPA.

Voici quelques unes de ces règles :

- la meilleure estimation des engagements de l'assureur (*Best Estimate*) est valorisée sans marge de prudence,
- la valorisation doit être cohérente avec les prix du marché : c'est la *Market-Consistency*,
- l'ajout d'une marge de risque calculée suivant le principe de l'*exit value* (valeur que l'assureur devrait payer pour transférer ses engagements à un autre assureur).

Les provisions techniques sont calculées d'une manière prudente, fiable et objective.

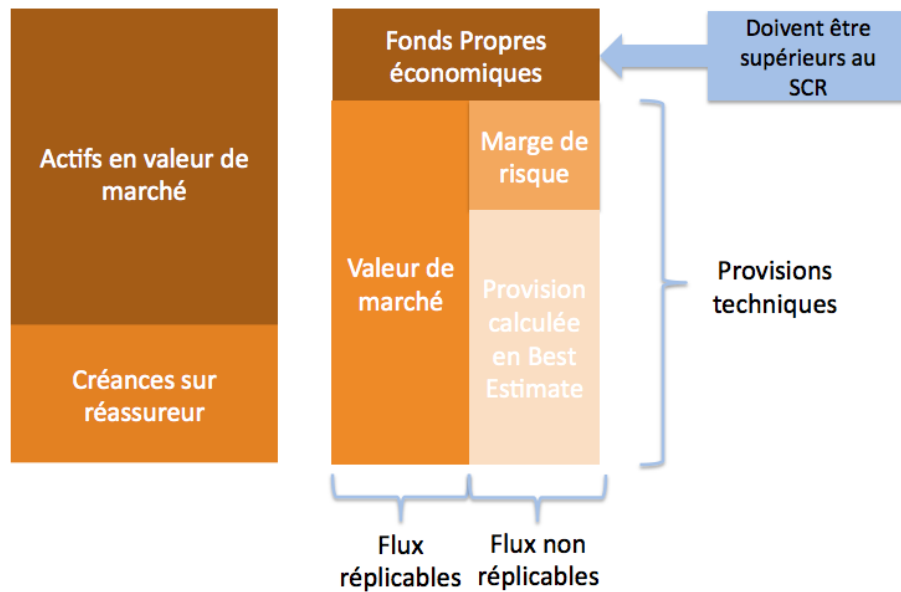


FIGURE 1.1 – Bilan Prudentiel

La directive distingue les flux répliquables des flux non répliquables. Les flux répliquables concernent les engagements de l'assureur qui pourraient être couverts parfaitement par un actif équivalent ayant une valeur de marché fiable et observable. Pour les flux non répliquables, les assureurs doivent procéder en deux temps :

- calcul de la meilleure estimation,
- calcul de la marge de risque.

Les flux répliquables sont peu présents en pratique et ne nous concerneront pas dans notre étude.

Nous allons présenter plus en détail le calcul des éléments de bilan dans cette partie. Nous exposerons tout d'abord les attentes de la directive Solvabilité II pour le calcul du bilan prudentiel, puis nous présenterons les méthodes de calcul et les hypothèses retenues pour notre étude.

1.2.1 La valorisation des éléments d'actif

Sous Solvabilité II, les éléments d'actifs du bilan prudentiel sont valorisés en valeur de marché. Pour les éléments cotés sur les marchés, cette information est disponible. La valorisation d'éléments non cotés sur les marchés se fait via l'utilisation de modèles mathématiques. Dans ce mémoire, nous prenons l'hypothèse que l'actif du contrat d'assureur est composé d'une poche action et d'une poche obligataire qui sont cotées sur les marchés. Aussi, nous ne rentrerons pas plus en détails sur la valorisation des actifs non cotés sur les marchés financiers.

1.2.2 La valorisation du passif

Le passif de l'assureur est composé des provisions techniques et des Fonds Propres. Les provisions techniques du bilan prudentiel comprennent le *Best Estimate* et la Marge de Risque.

• La provision *Best Estimate* (BE) :

L'article 77 de la Directive 2009/138/CE4 définit le cadre général de calcul des provisions techniques à inscrire au bilan prudentiel. Le *Best Estimate* est défini de la manière suivante : « [Le *Best Estimate*] correspond à la moyenne pondérée par leur probabilité des flux de trésorerie futurs, compte tenu de la valeur temporelle de l'argent (valeur actuelle attendue des flux de trésorerie futurs), estimée sur la base de la courbe des taux sans risque pertinente. ».

Mathématiquement, cela se traduit par la formule :

$$BE = E \left[\frac{\sum_k Flux_k}{(1 + i_k)^k} \right]$$

avec :

- $Flux_k$: les flux correspondants aux engagements de l'assureur vis-à-vis des assurés augmentés des frais de gestion pour l'année k ,
- i_k : le taux d'actualisation pour l'année k .

Tous les flux futurs potentiels liés à l'activité d'assurance doivent être identifiés et évalués, ces flux incluent les prestations, les frais et les taxes. L'horizon de projection considéré doit donc être suffisamment lointain pour tenir compte de tous les flux significatifs. L'article 77 précise également que les flux doivent être calculés bruts de réassurance.

Les assureurs sont tenus de revaloriser les garanties financières et options contractuelles incluses dans les contrats [30]. Dans le cadre d'une assurance vie, les contrats contiennent de nombreuses options cachées. Par exemple, les contrats prévoient une clause de rachat, un taux minimum garanti (TMG), etc.

L'article 77 précise : « Le calcul de la meilleure estimation est fondé sur des informations actualisées et crédibles ainsi que sur des hypothèses réalistes. Ce calcul fait appel à des méthodes actuarielles et statistiques adéquates, applicables et pertinentes. ». Or, les options cachées s'apparentent à des garanties financières. En particulier, la re-valorisation des prestations futures de l'assureur dépend des performances de son actif. La valorisation du passif s'apparente ainsi à la valorisation d'une option financière, elle doit donc être évaluée de la même manière. Les mathématiques financières nous enseignent que la valeur d'un instrument financier est égale à l'espérance sous la probabilité risque neutre des *payoffs* actualisés au taux sans risque.

Le référentiel Solvabilité II fait intervenir deux cadres de simulation distincts, à savoir les approches « Monde Réel » et « Risque Neutre ».

La probabilité historique correspond à la probabilité « réelle » des variations des actifs, observée sur le marché. Elle est utilisée lorsque nous nous plaçons dans un univers monde réel, pour obtenir une projection de l'univers économique à un horizon de temps futur. Sous la probabilité historique, on constate que les actifs plus risqués sont plus rentables en moyenne. Cela s'interprète comme une « prime de risque » : les investisseurs ne sont prêts à accepter du risque que sous condition qu'il soit plus rémunérateur.

La probabilité risque neutre est liée à la valorisation des produits dérivés. Un dérivé étant un instrument de couverture, le vendeur d'un tel produit se doit de « couvrir » son risque en construisant un portefeuille dynamique et suffisamment liquide permettant de reproduire les mouvements du dérivé. Le prix de ce portefeuille (qui correspond alors au prix du dérivé) est obtenu en passant de l'univers historique à un univers dit risque neutre. Ceci conduisant à la propriété : sous un univers risque neutre, les prix actualisés au taux sans risque sont des martingales [18].

Soit $M = (M_t)_{t \geq 0}$ un processus stochastique, (Ω, F, P) un espace probabilisé. M est une martingale si :

- $\forall t \geq 0, M_t \in L^1(\Omega, F, P)$
- $\forall t \geq 0, M_t$ est F_t - mesurable
- $\forall t \geq s, E[M_t | F_s] = M_s$

Ainsi, en synthétisant la distinction suivante peut être réalisée [18] :

- l'univers historique ou monde réel est l'univers de référence pour le calibrage lorsque l'on cherche à évaluer la distribution du rendement d'un actif et le risque de ruine, en particulier les probabilités des quantiles extrêmes,
- l'univers risque neutre permet de calculer le prix de la couverture d'un instrument financier comme une espérance de rendement, cohérente avec la notion d'absence d'opportunité d'arbitrage.

La provision « *Best Estimate* » de Solvabilité II entre dans cette logique d'évaluation du passif et se fera donc en univers risque neutre. Vu de l'instant initial nous avons :

$$BE_0 = E^Q \left[\sum_{t=1}^T Flux_t \times Factu_t \right]$$

avec :

- Q la probabilité risque neutre,
- T l'horizon de projection considéré,
- $Factu_t$ le facteur d'actualisation en date t .

En pratique, l'évaluation de la provision *Best Estimate* se fait par la méthode de Monte-Carlo. Nous utilisons un Générateur de Scénarios Économiques (GSE²) pour générer un grand nombre de scénarios économiques allant de la date d'évaluation jusqu'à l'échéance des contrats en portefeuille. Les scénarios permettent en particulier de revaloriser chaque année les flux de prestations de l'assureur en fonction de sa politique de revalorisation.

Nous utilisons alors l'estimateur suivant :

$$\widehat{BE}_0^K = \frac{1}{K} \sum_{j=1}^K \sum_{t=1}^T Flux_{t,j} \times Factu_{t,j}$$

$\sum_{t=1}^T Flux_{t,j} \times Factu_{t,j}$ peut être vu comme une réalisation de la variable aléatoire *Best Estimate* dans le scénario économique risque neutre numéro j obtenu par le GSE.

$$Factu_{t,j} = \exp\left(-\int_0^t r_x^j dx\right)$$

avec :

r_x^j le taux nominal x ans généré par le scénario risque neutre numéro j .

2. Un Générateur de Scénarios Économiques (GSE) est un outil permettant de générer des scénarios stochastiques modélisant différentes variables économiques et financières. Nous le présenterons plus en détail ultérieurement.

On a :

$$\lim_{K \rightarrow +\infty} \widehat{BE}_i^K = BE_0 \quad Q - p - s.$$

Il s'agit alors de prendre un nombre K de scénarios suffisamment grand pour estimer avec précision la provision *Best Estimate*.

• Marge de risque :

Les provisions techniques doivent refléter les incertitudes des marchés et doivent contenir une marge de risque. Elle a pour objectif :

- soit de garantir la possibilité de transférer le portefeuille à un tiers avec un niveau de confiance suffisant,
- soit de re-capitaliser la compagnie avec un niveau de confiance suffisant pour couvrir un scénario de *run off* préalablement défini.

La marge de risque n'a pas d'impact sur les conclusions de ce mémoire, nous décidons de la négliger. Aussi, nous ne rentrerons pas dans le détail du calcul de la marge de risque. Le lecteur pourra se reporter à l'article 37 du règlement délégué pour davantage de précisions.

• Fonds Propres :

Les Fonds Propres sont obtenus en faisant la différence entre les éléments d'actif et de passif.
Fonds Propres = Actifs en valeurs de marché - Provisions Techniques.

• Bilan Prudentiel simplifié :

Nous avons vu que, selon l'article 77 de la directive Solvabilité II, les assureurs doivent calculer différemment les engagements selon qu'ils soient répliquables ou non. Les engagements répliquables sont des cas marginaux qui ne concernent pas ou peu le marché français. Nous décidons de les négliger dans ce mémoire. Nous écartons également la marge de risque et nous ne nous intéresserons pas aux créances sur réassureurs. En omettant ces éléments, nous obtenons le bilan prudentiel simplifié suivant :

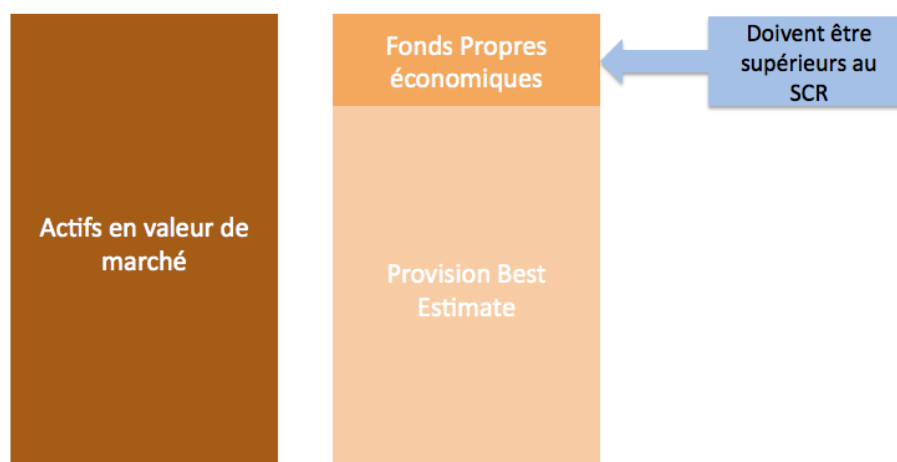


FIGURE 1.2 – Bilan Prudentiel Simplifié

Nous aurons : Fonds Propres = Actifs en Valeur de Marché - Provision *Best Estimate*.

Nous allons à présent présenter le GSE, puis nous présenterons le calcul du besoin réglementaire en capital de l'assureur (SCR).

1.3 Le Générateur de Scénarios Économiques (GSE)

Un Générateur de Scénarios Économiques (GSE) est un outil permettant de simuler des trajectoires pour différentes variables économiques. Le GSE de fixation permet notamment de générer les taux nominaux, les rendements actions, le taux d'inflation, etc.

Il peut être utilisé dans un objectif de gestion des risques ou dans un objectif d'évaluation. Nous avons vu à la section précédente que la gestion des risques se faisait en univers monde réel tandis que l'évaluation se faisait en univers risque neutre. Nous allons détailler ici le fonctionnement du GSE, les hypothèses de calcul et les tests de calibrage que nécessitent l'utilisation du GSE en univers monde réel ou en univers risque neutre.

1.3.1 Les corrélations entre les différentes variables financières

Les modèles stochastiques utilisés par le GSE nécessitent l'utilisation de mouvements browniens.

Un processus stochastique $(B_t)_{t \geq 0}$ dépendant du temps est un mouvement brownien standard si :

- $B_0 = 0$,
- $\forall t, s$ avec $t > s$, on a : $B_t - B_s$ suit une loi normale d'espérance nulle et de variance $t-s$: $N(0, t-s)$,
- $\forall n \geq 1, \forall (t_i)_{i=0, \dots, n}, 0 \leq t_0 \leq t_1 \leq \dots \leq t_n$ les accroissements $B_{t_0}, B_{t_1} - B_{t_0}, \dots, B_{t_n} - B_{t_{n-1}}$ sont indépendants.

Dans le GSE nous simulons des variables qui sont corrélées, nous devons donc tenir compte de ces corrélations. La matrice de corrélation utilisée est la suivante :

	Taux nominaux	Taux réels	Actions
Taux nominaux	1	0.9	0.17
Taux réels	0.9	1	0.08
Actions	0.17	0.08	1

TABLE 1.1 – Matrice de corrélation

Cette matrice est obtenue via les corrélations historiques entre les trois classes d'actifs que nous projetons. Nous utilisons la décomposition de Cholesky pour obtenir des mouvements browniens corrélés selon cette matrice.

Le théorème de factorisation de Cholesky nous dit : Si A est une matrice symétrique définie positive, alors il existe une matrice triangulaire inférieure L telle que $A = LL^T$ avec tous les éléments diagonaux de L positifs, où L^T est la transposée de la matrice L .

Soit Y un vecteur aléatoire gaussien centré de matrice de variance-covariance la matrice identité. Soit L la matrice issue de la factorisation de Cholesky de la matrice A . Alors le vecteur YL suit une loi multinormale centrée, de matrice de variance-covariance A .

On utilise cette décomposition pour obtenir des mouvements browniens corrélés.

1.3.2 Présentation des modèles de diffusion

Nous allons maintenant utiliser la décomposition de Cholesky en pratique afin d'obtenir une corrélation entre les mouvements browniens utilisés pour chaque modèle. Nous allons ainsi présenter les différents modèles de projection des scénarios de taux, d'inflation et de rendement action.

• Les modèles de taux :

Les taux réels et les taux nominaux simulés sont obtenus par un modèle de Vasicek à deux facteurs. Celui-ci est régi par les équations différentielles stochastiques suivantes :

$$\begin{cases} dr(t) = a_1(m(t) - r(t))dt + \sigma_1 dW_1(t) \\ dm(t) = a_2(\mu - r(t))dt + \sigma_2 dW_2(t) \end{cases}$$

avec :

- $r(t)$ le taux court en t , avec $r(0)$ le taux court initial,
- $m(t)$ le taux long en t , avec $m(0)$ le taux long initial,
- W_1, W_2 deux mouvements browniens indépendants,
- μ la moyenne des taux longs,
- a_1 la vitesse de retour du taux court vers le taux long,
- a_2 la vitesse de retour du taux long vers sa moyenne,
- σ_1, σ_2 la volatilité du taux court et du taux long.

W_2 sera simulé en utilisant la décomposition de Cholesky afin d'obtenir la corrélation souhaitée entre les différents actifs considérés.

Le modèle de Vasicek à deux facteurs est souple et permet de générer des taux négatifs, ainsi que de nombreuses formes de courbes de taux. Il comporte un effet de retour à la moyenne et dispose également d'une formule fermée pour calculer le prix des zéros coupons :

$$B(t, T) = \exp(A(T - t) - B_1 \times (T - t) \times r(t) - B_2 \times (T - t) \times m(t))$$

Cette formule sera détaillée en Annexe A.

Une obligation zéro coupon de maturité T est un contrat qui garanti à son détenteur le paiement d'une unité de monnaie à la date T sans paiement intermédiaire. La valeur de ce contrat est notée $B(t, T)$ à la date $t < T$.

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé complet, $F = (F_t)_{t \geq 0}$ la filtration brownienne engendrée par W_1 . Soit $P_t = \int_0^t r(u) du$ et \mathbb{P}^* la probabilité risque neutre définie par le fait que pour tout prix d'actif $X = (X_t)_{t \geq 0}$, $X_t^* = \frac{X_t}{P_t}$ est une $(\mathcal{F}, \mathbb{P}^*)$ martingale.

$B(t, T)^* = \frac{B(t, T)}{P_t}$ est une $(\mathcal{F}, \mathbb{P}^*)$ martingale.

$B(t, T)^* = E^Q[B(T, T)^* | F_t]$ ie $\frac{B(t, T)}{P_t} = E^Q[\frac{1}{P_T} | F_t]$ car $B(T, T) = 1$.

$B(t, T) = P_t E^Q[\frac{1}{P_T} | F_t] = E^Q[\frac{P_t}{P_T} | F_t]$ car P_t est F_t -mesurable.

Et finalement : $B(t, T) = E^Q[e^{-\int_t^T r(u) du} | F_t]$.

Si on se place en date $t=0$, on a : $B(t, T) = E^*[e^{-\int_t^T r(u) du}]$.

Le taux continu zéro-coupon au comptant (taux spot) à la date t est noté $R(t, T)$, il est défini par :

$$R(t, T) = -\frac{\ln B(t, T)}{T - t}.$$

C'est le taux d'intérêt continu associé à une obligation zéro coupon. La courbe des taux en t est la courbe de la fonction $T \mapsto R(t, T)$. Le taux court permet ainsi de reconstituer la courbe des taux.

Nous avons vu que le prix $B(t, T)$ est conditionné à l'information à la date t . D'où l'intérêt de la formule fermée du modèle de Vasicek à deux facteurs qui permet de calculer le prix des obligations zéro coupons à une date future sans passer par des Simulations dans les Simulations.

Le lecteur pourra se référer au livre « *Interest Rate Models - Theory and Practice* » [7] pour plus de détails concernant les modèles de taux.

• Le modèle de projection de l'inflation :

La projection de l'inflation repose sur la relation de Fisher :

$$i_{reel}(t) = \frac{i_{nominal}(t) - i(t)}{1 + i(t)}$$

avec :

- $i_{reel}(t)$ le taux court réel à la date t ,
- $i_{nominal}(t)$ le taux court nominal en t ,
- $i(t)$ l'inflation en t .

On en déduit : $i_{nominal}(t) = (1 + i_{reel}(t))(1 + i(t))$. Ce qui donne à l'ordre 1 : $i(t) \approx i_{nominal}(t) - i_{reel}(t)$.

Nous déduisons ainsi l'inflation à partir des taux courts réels et nominaux obtenus par les modèles de Vasicek à deux facteurs.

• Le modèle de rendement des actions :

La projection des actions se fait avec le modèle de Black & Scholes. Le modèle de Black & Scholes est construit à partir de lois normales. La loi normale a des queues de distributions inférieures à celles des performances actions observées sur les marchés. Pour éviter cet effet, le modèle est la somme de trois lois normales :

- une loi pour le coeur de la distribution,
- une loi pour chaque queue de la distribution pour traduire le risque de krach boursier et de forte hausse.

Nous pouvons passer d'un régime à l'autre avec une certaine probabilité. Dans chaque état, les performances actions futures sont obtenues avec un modèle de Black & Scholes dont les paramètres diffèrent.

Le modèle de Black & Scholes repose sur différentes hypothèses que nous supposons vérifiées. Pour plus de détails se rapporter au livre « *Options, futures et autres actifs dérivés* » de John Hull [17].

Le modèle de diffusion est le suivant :

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

avec :

- S_t le prix de l'action à la date t ,
- W_t un mouvement brownien géométrique avec volatilité σ et dérive μ , les paramètres σ et μ sont différents pour chaque régime.

Les paramètres utilisés par le GSE doivent être calibrés de façon à générer des scénarios réalistes.

Le règlement délégué de la commission précise notamment les méthodes de calcul des provisions techniques : « Les entreprises d'assurance et de réassurance fixent des hypothèses concernant les paramètres et scénarios futurs des marchés financiers qui sont appropriées et conformes à l'article 75 de la directive 2009/138/CE. Lorsque l'entreprise d'assurance ou de réassurance recourt à un modèle pour produire des projections de paramètres et scénarios futurs des marchés financiers, celui-ci respecte l'ensemble des exigences suivantes :

- il génère des prix d'actifs cohérents avec les prix des actifs observés sur les marchés financiers,
- il ne suppose aucune opportunité d'arbitrage,
- le calibrage des paramètres et des scénarios est cohérent avec la courbe des taux sans risque pertinents utilisée pour calculer la meilleure estimation visée à l'article 77, paragraphe 2, de la directive 2009/138/CE. »

Ces précisions indiquent en particulier que :

- les scénarios économiques générés sont cohérents avec les prix observés sur les marchés : c'est la *Market-Consistency*,
- dans chaque scénario généré, la valeur actualisée des actifs financiers est une martingale : c'est le test de martingalité.

Le calibrage des paramètres du GSE doit donc se confronter à la réglementation, mais également être adapté à l'univers de projection (monde réel ou risque neutre). Pour passer de l'univers historique à l'univers risque neutre, nous devons notamment risquer-neutraliser les rendements actions qui seront alors centrés sur le taux sans risque.

Le calibrage du GSE se fait en deux étapes. Tout d'abord, il s'agit de déterminer un jeu de paramètres qui vérifient le test de martingalité. Ensuite, nous vérifions que ce jeu de paramètres vérifie également le test de *Market-Consistency*. Si ce n'est pas le cas, nous retournons à l'étape précédente et testons un nouveau jeu de paramètres. Nous allons présenter plus en détail ces deux tests.

1.3.3 Le test de Martingalité

Le test de Martingalité a pour objectif de vérifier que sous la probabilité risque neutre les prix actualisés des actifs financiers sont des martingales.

• Le test de Martingalité pour les performances actions :

Les scénarios de performance action vérifient le test de martingalité si la moyenne du prix des actions actualisé à la date initiale est égale au prix initial de l'action.

• Le test de Martingalité pour les scénarios de taux :

Pour les taux nominaux, le test de martingalité consiste à vérifier que la moyenne des intégrales des taux courts générés est suffisamment proche de la courbe de référence fournie par l'EIOPA.

En notant $P^M(0, T)$ la courbe de référence, il s'agit de vérifier si :

$$\forall T, \quad E^Q[\exp(-\int_0^T r_u du)] = P^M(0, T)$$

avec :

r_u le taux court nominal.

Cette condition est suffisante pour vérifier le test de Martingalité. Ce résultat a été démontré dans le mémoire de Quentin Bazoge : « Effet du choix des modèles et de leur calibrage dans le GSE sur le bilan » [1].

1.3.4 Le test de *Market-Consistency*

Le test de *Market-Consistency* est un test de dispersion. Ce test consiste à vérifier que la volatilité des variables financières permet bien de retrouver le prix des options sur le marché.

- **Le test de *Market-Consistency* pour les performances actions :**

Nous retenons un indice de Volatilité implicite du CAC40 : VCAC pour réaliser ce test.

- **Le test de *Market-Consistency* pour les scénarios de taux :**

Ce test est réalisé avec un put sur un Bund allemand. Nous re-calculons le prix de cette option avec la courbe des taux générée par notre GSE.

1.4 Le calcul du SCR

Le SCR (*Solvency Capital Requirement*) est le capital économique dont a besoin une entreprise d'assurance ou de réassurance pour limiter la probabilité de ruine à 0,5% à horizon un an. Les assureurs peuvent estimer le SCR :

- soit à partir d'un modèle interne propre à chaque assureur,
- soit à l'aide d'une formule standard proposée par l'EIOPA.

1.4.1 Le SCR en formule standard

Le calcul du SCR dans la formule standard se fait en évaluant l'effet sur l'actif net de l'assureur de différents chocs fixés par la commission européenne.

Il concerne différents domaines de l'activité des organismes d'assurance :

- les risques liés aux évolutions des marchés financiers,
- le risque de souscription en vie,
- le risque de souscription en non-vie,
- le risque de souscription en santé,
- le risque de crédit,
- le risque opérationnel.

Pour chaque domaine, la formule standard propose de calculer les Fonds Propres nécessaires pour faire face à différents risques identifiés. Cette exigence en capital est déterminée par des chocs pour chaque risque élémentaire. En agrégeant ces montants à l'aide de matrice de corrélations nous obtenons

le capital requis de base : BSCR (*Basic Solvency Capital Requirement*). Le SCR est ensuite estimé en ajoutant au BSCR les ajustements et le risque opérationnel [30].

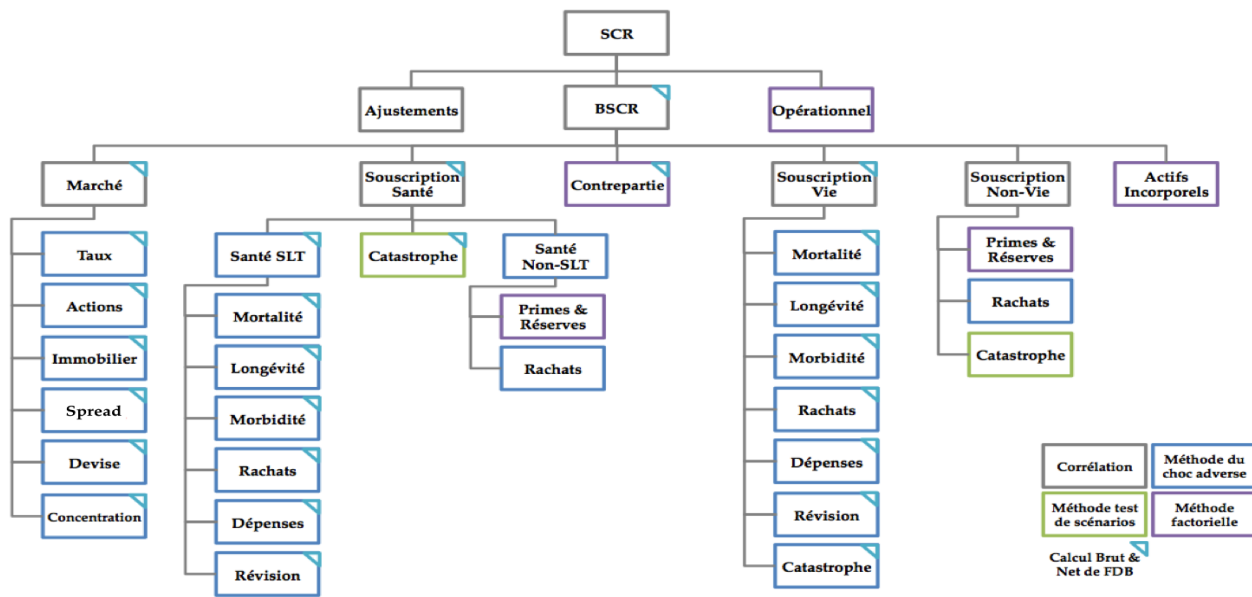


FIGURE 1.3 – Pieuvre SCR

L'approche formule standard repose ainsi sur trois étapes fondamentales pour le calcul du BSCR :

- Étape 1 : déterminer le capital économique pour chaque « risque élémentaire » : actions, taux, mortalité, etc. Durant cette étape, nous calculons les Fonds Propres économiques après chaque choc considéré. Le capital économique requis pour faire face au choc est la différence des Fonds Propres après le choc et des Fonds Propres avant le choc.
- Étape 2 : agréger les capitaux au sein de chaque module de risques (marché, vie, non-vie, etc) → agrégation intra-modulaire,
- Étape 3 : agrégation des capitaux des différents modules → agrégation inter-modulaire [12].

$$\text{On aura : } BSCR = \sqrt{\sum_{i,j} \rho_{i,j}^M \times SCR_i \times SCR_j}$$

avec :

SCR_i le capital associé au module i .

Enfin, le SCR est obtenu via la formule suivante : $SCR = BSCR + Adj + SCR_{op}$

avec :

- Adj l'ajustement pour les effets d'atténuation des risques provenant de la participation aux bénéfices et des impôts différés,
- SCR_{op} l'exigence de capital pour risque opérationnel.

1.4.2 Le SCR en modèle interne

Dans ce mémoire, nous proposons l'utilisation d'un modèle interne. Le calcul du SCR en modèle interne repose sur la formule suivante :

$$SCR_0 = FP_0 - Factu_1 \times q_{0,5\%}(FP_1)$$

Cette méthode est privilégiée par l'EIOPA, car elle permet une modélisation personnalisée du portefeuille de l'assureur. Cependant, l'estimation du quantile à 0,5% des Fonds Propres à horizon un an s'avère très délicate. Pour ce faire les assureurs utilisent souvent la méthode des Simulations dans les Simulations dont nous allons présenter les principes.

Chapitre 2

La méthode des Simulations dans les Simulations : limites et alternatives

« Si les gens ne croient pas que les mathématiques sont simples, c'est seulement parce qu'ils ne réalisent pas combien la vie est compliquée. »

John Louis von Neumann

2.1 La méthode des SdS

2.1.1 Présentation de la méthode des SdS

La méthode des Simulations dans les Simulations (SdS) consiste à réaliser deux niveaux de simulations imbriquées. Elle est souvent utilisée par les assureurs en vue d'obtenir une estimation de la distribution des Fonds Propres dans le futur.

Plaçons nous à la date initiale t , où toute l'information présente sur les marchés est supposée connue. La méthode des SdS nous permet de projeter le bilan prudentiel à une date future t' . Pour ce faire, nous projetons l'univers économique actuel en date t' via des simulations univers monde réel obtenues par le GSE. Connaissant alors l'univers économique en date t' issu de la simulation, nous pouvons re-dresser un bilan prudentiel. Nous valorisons l'actif du bilan conditionnellement à l'information de marché de la même manière qu'à l'instant t . Le calcul de la provision *Best Estimate* est plus complexe. Nous devons re-générer des scénarios univers risque neutre pour réaliser ce calcul. Ces scénarios secondaires sont conditionnés à l'information de marché en date t' . En particulier, le GSE doit être re-calibré en date t' pour correspondre aux nouvelles données de marché.

Voici un schéma récapitulatif de la méthode des SdS :

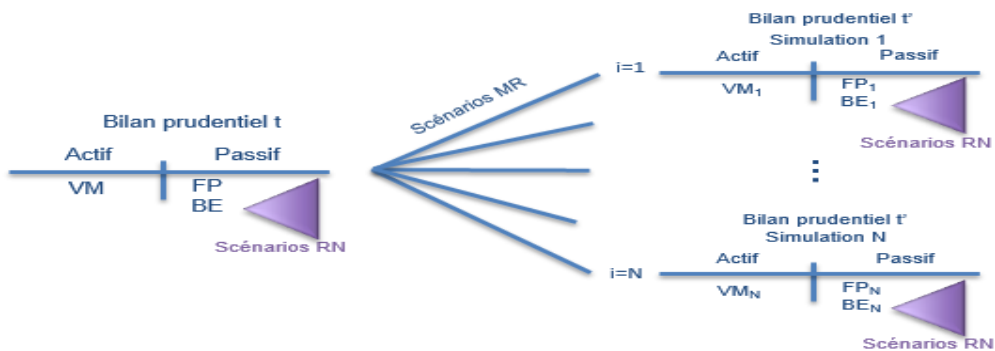


FIGURE 2.1 – La méthode des SdS

2.1.2 La valorisation du bilan prudentiel futur

Obtenir le bilan prudentiel en date t' nécessite une ré-évaluation des éléments d'actif et de passif en date t' . Nous allons à présent détailler les hypothèses que nous prenons dans notre étude pour évaluer le bilan prudentiel à une date future.

Nous souhaitons nous placer dans un cadre suffisamment général pour être compatible avec la plupart des modèles mathématiques existants et permettant de présenter de façon formelle le calcul des éléments de bilan dans le futur (en particulier celui de la provision *Best Estimate*).

Le cadre dans lequel nous nous plaçons est directement inspiré de la valorisation des options américaines et du livre « Introduction au calcul stochastique appliqué à la finance » par Damien Lamberton et Bernard Lapeyre [20]. Un cadre proche est également présenté dans le mémoire de Faris Rouchati « Projection du ratio de solvabilité d'un assureur retraite : les méthodes paramétriques (*Curve Fitting* et *Least Squares Monte-Carlo*) peuvent-elles se substituer à la méthode des Simulations dans les Simulations ? » [27].

Soit $(\Omega, \mathcal{F}, \mathcal{F}=(\mathcal{F})_{0 \leq t \leq T}, \mathbb{P})$ un espace probabilisé filtré avec :

- Ω l'ensemble de tous les événements possibles, muni de la probabilité réelle \mathbb{P} et de la tribu \mathcal{F} ,
- \mathcal{F} une filtration sur l'ensemble des parties de Ω ,
- T la dernière échéance du passif de l'assureur,
- \mathcal{F}_t qui contient toute l'information disponible sur le marché jusqu'à la date t .

Nous assimilons l'état du marché à un processus stochastique multidimensionnel $Y=(Y_t)_{0 \leq t \leq T}$ avec : Y_t \mathcal{F}_t -adapté. Nous considérons que l'état du marché est parfaitement représenté par les variables économiques projetées dans le GSE (courbe des taux, rendement actions, inflation, etc). Les variables non-projetées par le GSE sont moins pertinentes pour notre étude. Par exemple, nous ne projetons pas les rendements immobiliers, en effet il n'y a pas d'immobilier dans le contrat que nous étudions.

Nous supposons, de plus, que Y_t est markovien. Dire qu'un processus $(X_t)_{t \geq 0}$ est markovien signifie que le comportement futur de ce processus après t dépend uniquement de X_t .

Nous faisons également l'hypothèse que le processus Y_t est suffisamment régulier : nous supposons que ses accroissements sont indépendants et que la fonction $t \mapsto Y_t$ est Càdlàg (continue à droite et limitée à gauche). Enfin, nous supposons que les flux futurs de l'actif et du passif de l'assureur sont de carré intégrables.

Nous allons maintenant présenter la re-valorisation du bilan prudentiel à l'instant $t' > t$, vu de l'instant initial t . Nous rappelons que nous supposons l'état du marché en date t connu et que l'état du marché en date t' est obtenu par des simulations univers monde réel effectuées sur le GSE.

• Le re-calibrage du GSE à horizon un an :

Le calibrage du GSE que nous avons effectué à l'instant initial n'est plus valable en date t' . Le GSE peut en effet générer des scénarios de taux très « extrêmes » (notamment des courbes des taux inversées), en ce cas, ne pas changer les paramètres des modèles lors d'une nouvelle utilisation du GSE risquerait de sous-estimer fortement les risques encourus par l'assureur.

Comme nous l'avons vu au chapitre 1, à l'instant initial, nous devons vérifier :

$$\forall T, \quad E^Q[\exp(-\int_0^T r_u du)] = P^M(0, T)$$

avec :

$P^M(0, T)$ la courbe de référence fournie par l'EIOPA.

À l'instant t' nous ne disposons pas de courbe de référence. Nous considérons alors comme référence la courbe des taux reconstituée par la formule fermée du modèle de Vasicek à deux facteurs. Notons, que dans le cadre d'un calcul de SCR en formule standard, nous devons appliquer des chocs à cette courbe lorsque nous voulons calculer le SCR relatif à un choc de taux.

Il nous faut trouver à l'instant futur t' , dans chaque scénario le jeu de paramètres du modèle de Vasicek qui nous permet de nous approcher au mieux de la nouvelle courbe de taux de référence. Si nous n'avons pas de choc de taux à effectuer, ce qui est notamment le cas lorsque l'on calcule un SCR vision modèle interne ou que nous sommes dans le scénario central pour un calcul de SCR vision formule standard, le modèle de Vasicek à deux facteurs nous autorise à éviter un re-calibrage coûteux. Nous pouvons en effet utiliser la simplification suivante : nous conservons les paramètres initiaux à l'exception du taux court initial et du taux long initial qui sont remplacés par ces taux respectifs générés entre 0 et 1 dans chaque scénario.

Cette simplification n'est cependant pas pertinente avec d'autres modèles de taux plus complexes. Il nous est également impossible de l'utiliser lorsque nous effectuons un choc de taux, il sera alors nécessaire de re-calibrer les paramètres du modèle de Vasicek à deux facteurs.

Cette phase de re-calibrage est particulièrement coûteuse et ne peut être effectuée manuellement puisqu'elle doit être réalisée dans chaque scénario. Aussi, il faudra trouver le jeu de paramètres qui minimise l'écart quadratique moyen (par exemple) entre la courbe de référence et la courbe de taux re-générée par les scénarios. Ce re-calibrage pourra être fait en pratique par une fonction d'optimisation. Le test de martingalité est vérifié par construction du modèle pour les actions.

Le test de *Market-Consistency* est effectué sur les taux et les actions de la manière suivante :

- pour les taux : nous re-calculons le prix du put sur obligation avec la courbe des taux tirée,
- pour les actions : nous construisons une fonction de volatilité implicite. La volatilité implicite diminue lorsque la performance du cours de l'année passée augmente et inversement.

Nous n'entrerons pas davantage dans ces détails techniques qui ne sont pas le sujet de notre étude. Nous éveillons cependant l'attention du lecteur sur ces problématiques qui sont essentielles lorsque l'on cherche à projeter un bilan prudentiel dans le futur.

• La re-valorisation de l'actif du bilan prudentiel :

La valeur de marché de la poche action de l'assureur en date t' dans le scénario primaire numéro i est obtenue par la formule suivante :

$$VM_{t'}^i = VM_t \times (1 + per f_{t \rightarrow t'}^i)$$

avec :

- VM_t la valeur de marché des actions en date t ,
- $per f_{t \rightarrow t'}^i$ la performance action entre la date t et la date t' dans le scénario i .

Nous allons à présent nous intéresser à la valorisation de la poche obligataire.

• Quelques rappels sur les instruments obligataires :

Une obligation est un instrument financier versant un coupon annuel à date anniversaire et remboursant le nominal à l'échéance.

Voici ci-dessous une représentation schématique des flux d'une obligation de maturité T , de valeur nominale N délivrant un coupon C :

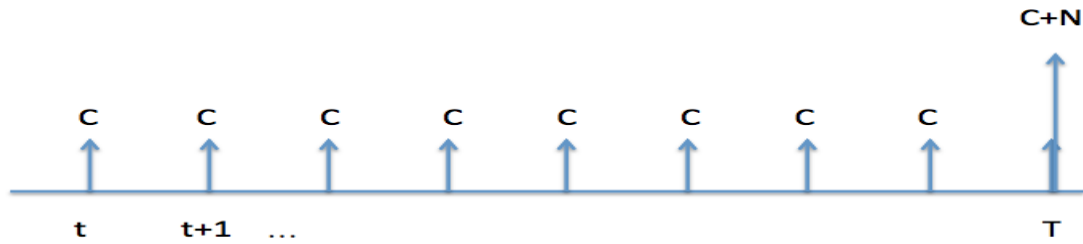


FIGURE 2.2 – Flux d'une obligation

Le taux de coupon et la valeur nominale d'une obligation sont fixés lors de son émission. Une obligation est un instrument financier et doit donc être valorisée en tant que tel. Sa valeur de marché est ainsi la somme de ses flux futurs actualisés :

$$P_0 = \sum_{k=1}^T \frac{C}{(1+i_k)^k} + \frac{N}{(1+i_T)^T}$$

avec :

$\frac{1}{(1+i_k)^k}$ le facteur d'actualisation en date k .

Dans notre cas nous utilisons $\exp(-\int_{t_0}^{t_1} r_t dt)$ comme facteur d'actualisation en date t_0 pour un flux d'échéance t_1 , avec r_t le taux court nominal.

Ce qui donne pour le calcul de la valorisation d'une obligation en date t :

$$P_t = \sum_{k=1}^T C \times \exp(-\int_0^k r_t dt) + N \times \exp(-\int_0^T r_t dt).$$

Si nous nous plaçons en date t' (avec $t < t' < T$), dans le scénario numéro i généré par le GSE nous obtenons alors :

$$P_{t'} = \sum_{k=1}^{T-(t'-t)} C \times \exp(-\int_0^k r_t^i dt) + N \times \exp(-\int_0^{T-(t'-t)} r_t^i dt)$$

avec :

r_t^i le taux nominal de maturité t généré dans le scénario primaire i .

La poche obligataire de l'assureur est investie sur des obligations d'état. Par soucis de simplification, nous projetons une obligation unique reflétant la durée réelle du portefeuille (avec $t < t' < T$).

La durée D est définie par :

$$D = \frac{\sum_{i=1}^n \frac{t(i) \times F_i}{(1+r)^{t(i)}}}{\sum_{i=1}^n \frac{F_i}{(1+r)^{t(i)}}$$

avec :

- F_i les flux futurs,
- t_i l'intervalle de temps en année, séparant la date d'actualisation de la date de flux F_i
- r_i le taux sans risque.

Nous considérons que la valeur de marché de la poche obligataire en date t' est facteur de la valeur de marché de cette obligation.

L'obligation considérée est alors re-valorisée en date t' dans chaque scénario économique généré. La nouvelle valeur de marché de la poche obligataire est ainsi facteur de cette nouvelle valeur de marché à laquelle on ajoute le coupon ou les coupons versés entre la date t et la date t' . Nous allons maintenant présenter la re-valorisation du passif.

• **La re-valorisation du passif du bilan prudentiel :**

Nous avons vu que l'évaluation de la provision *Best Estimate* nécessite la prise en compte des liens actifs-passif et se fait via des projections univers risque neutre. À l'instant initial, nous avons :

$$BE_t = E^Q \left[\sum_{i=1}^T Flux_i \times Actu_i \right]$$

avec :

Q la probabilité risque neutre.

En date t' , le calcul de la provision *Best Estimate* est conditionné par l'état du marché. L'état du marché en date t' est modélisé par le processus (Y_t) que nous avons décrit au début de ce chapitre. Nous obtenons ainsi :

$$BE_{t'} = E^Q \left[\sum_{i=1}^T Flux_i \times Actu_i \middle| Y_{t'} \right].$$

Plaçons nous au sein du scénario primaire numéro i , en date t' . L'état du marché $Y_{t'}^i$ est alors connu. Nous obtenons de ce fait :

$$BE_{t'}^i = E^Q \left[\sum_{k=1}^T Flux_k \times Actu_k \middle| Y_{t'}^i \right] = E^Q \left[\sum_{k=1}^T Flux_k \times Actu_k \right].$$

Pour calculer la provision *Best Estimate* dans le scénario i nous utilisons l'estimateur suivant :

$$\widehat{BE}_{t'}^{K,i} = \frac{1}{K} \sum_{j=1}^K \sum_{k=1}^T Flux_{k,j}^i \times Actu_{k,j}^i$$

$\sum_{k=1}^T Flux_{k,j}^i \times Actu_{k,j}^i$ peut être vu comme une réalisation de la variable aléatoire *Best Estimate* $BE_{t'}$ dans le scénario économique univers risque neutre numéro j obtenu par le GSE.

Et : $Actu_{k,j}^i = \exp\left(-\int_0^k r_t^{i,j} dt\right)$

avec :

$r_x^{i,j}$ est le taux nominal x ans généré dans le scénario risque neutre numéro j conditionné par le scénario monde réel i entre t et t' .

On a de plus : $\lim_{K \rightarrow +\infty} \widehat{BE}_{t'}^{K,i} = BE_{t'}^i$ Q - p - s

Après avoir présenté la méthode des SdS, nous allons maintenant nous intéresser aux différentes utilisations qui peuvent être faites de la méthode des SdS pour répondre à différentes problématiques actuarielles.

2.1.3 Les utilisations classiques de la méthode des SdS :

- L'utilisation de la méthode pour calculer le SCR :

Une utilisation classique de la méthode des SdS est le calcul du SCR en modèle interne.

Le calcul du SCR en modèle interne repose en effet sur la détermination du quantile à 0,5% des Fonds Propres à horizon $t=1$ an. Le SCR est ensuite obtenu par différence entre les Fonds Propres initiaux et le quantiles estimés par la méthode des SdS : $SCR_t = FP_t - q_{0,5\%}(FP_{t+1}) \times F_{actu_{t+1}}$.

Voici un schéma présentant l'utilisation de la méthode dans ce cadre :

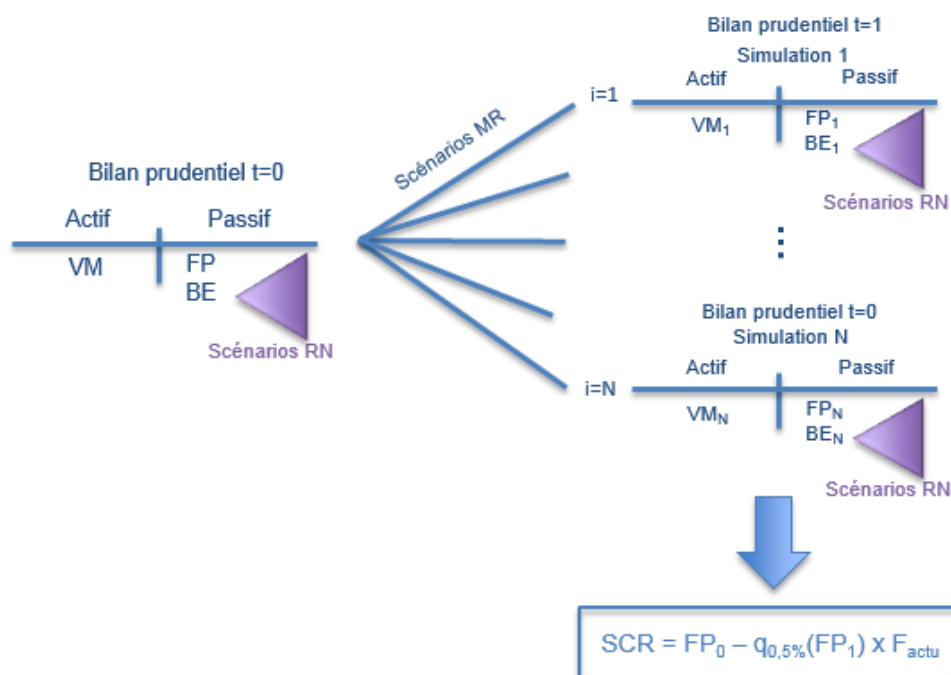


FIGURE 2.3 – La méthode des SdS pour le calcul du SCR

Dans le chapitre suivant, nous proposerons notamment d'utiliser une méthode alternative à la méthode des SdS pour le calcul d'un SCR en vision modèle interne. L'usage des SdS pour le calcul d'un SCR est probablement une des utilisations les plus courantes de la méthode des SdS et de nombreuses méthodes d'accélération des SdS ont été élaborées spécifiquement pour répondre à cette problématique. Cependant, comme nous allons le voir, la méthode des SdS peut également être utilisée pour répondre à d'autres problématiques actuarielles.

- L'utilisation de la méthode des SdS pour estimer la distribution du ratio de solvabilité futur :

Une autre utilisation possible de la méthode des SdS est l'estimation de la distribution des Fonds Propres futurs ou du ratio de solvabilité futur. L'estimation de la distribution des Fonds Propres futurs peut être utilisée pour un calcul de SCR, mais également dans le cadre du choix d'une allocation d'actifs. Par exemple, on peut s'intéresser aux Fonds Propres moyens et à leur volatilité pour différentes

allocations d'actifs. En ce cas le nombre de scénarios primaires peut être éventuellement réduit.

Nous avons expliqué au chapitre précédent l'intérêt que représente l'étude du ratio de solvabilité et en particulier l'étude de la distribution probable de ce ratio dans le futur. La méthode des SdS est également adaptée à l'étude de la distribution du ratio de solvabilité futur t' .

L'obtention du bilan prudentiel en date t' se fait de la manière précédemment décrite. Elle nous permet ainsi d'obtenir une estimation de la distribution des Fonds Propres futurs. Pour obtenir l'estimation de la distribution du ratio de solvabilité futur, il nous faut calculer pour chaque bilan projeté en date t' un SCR. Nous avons vu que le SCR peut être obtenu via l'utilisation d'un modèle interne ou de la formule standard.

L'utilisation d'un modèle interne pour le calcul du SCR en date t' implique la projection de chaque bilan obtenu en date t' en date $t'+1$. Autrement dit, il s'agit de faire des Simulations dans les Simulations dans les Simulations (SdSdS). Cette méthode est particulièrement complexe, puisque l'on rajoute un niveau de simulations. À titre d'exemple, considérer la génération de 1 000 scénarios primaires monde réel entre la date t et la date t' , puis 1 000 scénarios monde réel entre la date t' et $t'+1$ et utiliser 1 000 scénarios risque neutre pour chaque valorisation de la provision *Best Estimate* revient à générer 1 000 000 000 scénarios risques neutre en date $t'+1$. Nous reviendrons plus tard sur le choix du nombre de scénarios selon l'objectif.

L'utilisation de la formule standard pour la résolution de cette problématique pose également des difficultés. Pour calculer le SCR en formule standard, il faudra calculer plusieurs bilans prudentiels à la suite de chaque scénario primaire : un bilan « central » et des bilans choqués, tel que nous l'avons présenté lors du calcul du SCR en formule standard. Pour chacun des chocs que nous devons effectuer, la valeur de marché de l'actif doit être ré-évaluée. Le GSE doit également être re-calibré conditionnellement aux nouvelles conditions de marchés, de nouveaux scénarios risque neutre sont alors générés pour évaluer la provision *Best Estimate*.

Cette opération s'avère complexe, le re-calibrage du GSE après les chocs est particulièrement coûteux en temps de calcul. En effet, l'utilisation du modèle de Vasicek à deux facteurs nous permet d'utiliser une méthode de re-calibrage simplifiée lorsque nous utilisons un modèle interne ou lors du calcul de la provision *Best-Estimate* à horizon un an en scénario central. En revanche, lorsque nous voulons calculer un choc de taux (à la hausse ou à la baisse) en vue de calculer le SCR à horizon un an avec la formule standard, nous ne pouvons pas utiliser cette simplification dans le modèle de Vasicek. Nous devons alors re-constituer par la formule fermée de Vasicek la courbe de référence qui sera choquée avec les coefficients fournis par l'EIOPA. Il nous faut à cet effet, dans chaque scénarios, déterminer le jeu de paramètres initiaux du modèle de Vasicek à deux facteurs qui permet de s'approcher au plus près de la courbe de référence. C'est ce que nous avons fait à l'instant initial. Il nous est impossible de faire cette opération manuellement puisqu'il faudrait le faire dans chaque scénario monde réel. Il faut donc automatiser cette opération. Le re-calibrage des modèles de taux dans les scénarios choqués est ainsi une étape particulièrement complexe.

Dès lors, si le choix de la formule standard semble plus simple de prime abord, elle s'avère également très complexe à mettre en oeuvre du point de vue opérationnel. La mise en oeuvre des SdS vision formule standard a été proposée dans le mémoire de Faris Rouchati, auquel nous renvoyons le lecteur pour plus de détails : « Projection du ratio de solvabilité d'un assureur retraite : les méthodes paramétriques (*Curve Fitting* et *Least Squares Monte-Carlo*) peuvent-elles se substituer à la méthode des Simulations dans les Simulations ? » [27].

Voici un schéma résumant l'utilisation de la méthode pour l'estimation de la distribution du ratio de solvabilité à horizon un an (le calcul du SCR formule standard n'est pas détaillé ici) :

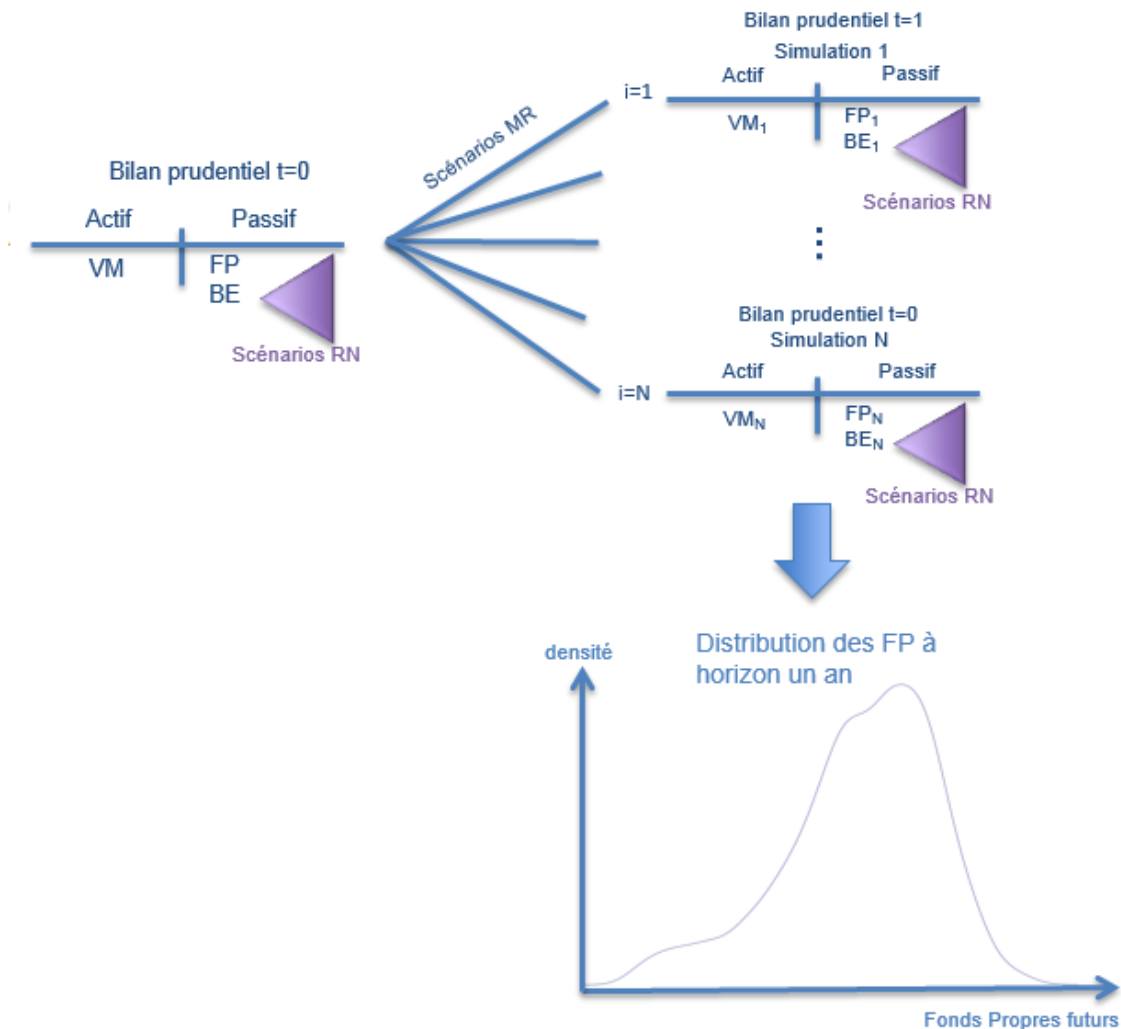


FIGURE 2.4 – La méthode des SdS pour l'estimation de la distribution du ratio de solvabilité à horizon un an

Nous avons vu que l'estimation de la distribution du ratio de solvabilité future est particulièrement intéressante à obtenir pour un assureur. Cette problématique peut également s'inscrire dans le cadre d'un exercice ORSA. En ce cas, on se placera souvent à un horizon de temps de 5 ans. Il s'agira de projeter l'univers économique à horizon 5 ans par des scénarios primaires. Les bilans prudentiels successifs ne seront pas nécessairement re-calculés à horizon 1, 2, 3, 4 ans en faisant l'hypothèse que l'assureur ne fait pas faillite. Conditionnellement à l'information de marché à horizon 5 ans, nous re-dresserons les différents bilans prudentiels.

Nous pouvons également envisager d'étudier l'effet de différentes allocations d'actifs sur le ratio de solvabilité futur. Cela revient à effectuer plusieurs fois la méthode des SdS pour obtenir par exemple le ratio de solvabilité moyen espéré à horizon un an et sa volatilité en fonction des allocations d'actifs considérées. Cette étude permettra alors d'aiguiller le choix de l'allocation d'actifs pour l'assureur.

2.1.4 Le choix du nombre de scénarios primaires

La résolution de ces deux problématiques : calcul de SCR vision modèle interne ou estimation d'une distribution future (des Fonds Propres ou du ratio de solvabilité) amène à des interrogations différentes.

Dans le premier cas, l'estimation du SCR repose sur la détermination d'un quantile à 0,5%. Dès lors, le choix du nombre de scénarios primaires a une importance capitale. Choisir d'utiliser 1 000 scénarios primaires, revient à considérer que le quantile obtenu est le 5^{ème} pire scénario obtenu à horizon un an. Les scénarios générés n'étant pas déterministes, estimer un quantile sur un nombre aussi restreint peut donner un SCR volatile.

Nous nous apercevons qu'estimer un SCR sur 5 scénarios peut être problématique et que le choix du nombre de scénarios primaires doit être élevé dans ces conditions. Proposer 10 000 scénarios primaires revient à estimer le quantile sur le 50^{ème} pire scénario, ce qui améliore la précision, mais augmente également fortement le coût de l'utilisation de la méthode des SdS.

Dans le cours d'agrégation des risques proposé par Alexandre Boumezoued à l'EURIA, le choix du nombre de scénarios primaires pour le calcul du SCR est discuté. Ce choix est délicat à objectiver. Une approche itérative est proposée :

- on effectue plusieurs calculs en utilisant des jeux de simulations primaires de taille croissante,
- dès que l'ajout de simulations primaires ne modifie pas de manière significative l'estimation du capital économique, le processus s'arrête.

Cette méthode montre un écart relatif proche de 1% entre le capital économique de référence et le SCR estimé lorsque 10 000 scénarios primaires sont utilisés. Un écart négligeable est observé pour l'utilisation de 15 000 scénarios primaires (moins de 0,1%).

Cependant, cette étude est difficilement généralisable, puisque la vitesse de convergence d'un SCR dépend de nombreux paramètres : choix des modèles du GSE, type de contrat, politique de revalorisation, allocation d'actifs, etc. Nous y reviendrons lors de l'application numérique.

D'un autre côté, la problématique du nombre de scénarios primaires pour l'estimation de la distribution du ratio de solvabilité futur est différente. Les assureurs s'intéressent souvent au ratio de solvabilité moyen, ce qui pose moins de problème quant au choix du nombre de scénarios. Nous pouvons également nous intéresser à la volatilité du ratio de solvabilité ou à un quantile. Un assureur peut par exemple se demander quel est le quantile à 5% de son ratio de solvabilité futur. Dans ce cas, utiliser 1 000 scénarios primaires revient à estimer le quantile sur le 50^{ème} pire scénario et le choix du nombre de scénarios primaires peut être moins important.

D'autre part, ce choix est également influencé par le calcul du SCR futur qui nécessite soit de faire différents calculs de *Best Estimate* futurs en scénario central et choqués avec le re-calibrage du GSE que cela implique, soit de faire des SdSdS comme expliqué précédemment. Dans les deux cas, la projection du ratio de solvabilité s'avère plus complexe à nombre de scénarios égal que le calcul du SCR en 0 en vision modèle interne. Dès lors, le choix du nombre de scénarios primaires pour la résolution de cette problématique ne peut être aussi élevé que pour le calcul du SCR.

Le choix du nombre de scénario dépend ainsi fortement de la problématique que nous souhaitons étudier.

2.1.5 La problématique du temps de calcul

Comme nous l'avons vu, les SdS permettent de répondre à différentes problématiques actuarielles. Malheureusement, aussi utile que puisse être cette méthode elle s'avère extrêmement complexe à mettre en oeuvre. Pour en prendre conscience, considérons le cas où nous souhaitons estimer le SCR initial avec

une approche modèle interne, avec 15 000 scénarios primaires monde réel et 1 000 scénarios secondaires risque neutre. Cela revient à générer 15 000 000 scénarios économiques en date $t=1$. En particulier, nous devons simuler 15 000 000 courbes des taux allant jusqu'à l'échéance des flux de passifs. Cette complexité rend la méthode des SdS difficile à mettre en oeuvre et de nombreuses méthodes alternatives ont été proposées. Nous allons en présenter rapidement quelques unes et exposer leurs avantages et inconvénients.

2.2 Les alternatives usuelles à la méthode des SdS

2.2.1 Les approches paramétriques

L'objectif de ces approches est d'obtenir une estimation de la distribution des Fonds Propres futurs à l'aide d'une forme paramétrique (formule fermée) [12].

On distingue deux méthodes pour réduire le temps de calcul :

- la réduction du nombre de simulations primaires (univers monde réel),
- la réduction du nombre de simulations secondaires (univers risque neutre).

L'approche *Curve Fitting* utilise la première méthode, l'approche LSMC la seconde. Notons que la complexité de la génération de scénarios monde réel n'est pas nécessairement équivalente à celle de la génération de scénarios risque neutre.

• La méthode *Curve Fitting* :

L'objectif de l'approche *Curve Fitting* est de calibrer une forme paramétrique permettant d'exprimer les Fonds Propres à horizon un an en fonction de différents facteurs de risques.

On génère un jeu de scénarios complet univers monde réel entre $t=0$ et $t=1$. On sélectionne ensuite les scénarios les plus extrêmes par rapport aux facteurs de risques retenus. Pour chaque scénario extrême sélectionné, on calcule les Fonds Propres correspondants via des simulations risque neutre (pour le calcul du BE). On calibre ensuite une forme paramétrique sur les scénarios primaires sélectionnés.

Enfin, on utilise la forme paramétrique calibrée sur le jeu complet de scénarios monde réel pour obtenir une estimation de la distribution des Fonds Propres en $t=1$ et en déduire le SCR.

Le calibrage de la méthode *Curve Fitting* nécessite ainsi un faible nombre de simulations monde réel suivi d'un jeu complet de simulations risque neutre.

• La méthode *Least Square Monte Carlo (LSMC)* :

La méthode LSMC se base sur l'idée que la méthode des SdS apporte des informations redondantes [19]. Deux environnements économiques générés proches conduisent à des simulations secondaires ayant des lois quasi identiques. La méthode LSMC propose alors de générer une unique simulation secondaire par simulation primaire. Il faut cependant veiller à calibrer la forme paramétrique sur des scénarios extrêmes (on estimera qu'un scénario est extrême en fonction de facteurs de risques sélectionnés).

2.2.2 Les *Replicating Portfolios*

Un « portefeuille répliquant » (ou « *replicating portfolio* ») est un portefeuille d'actifs qui reproduit les *cash flows* de passifs pour chaque simulation d'univers économique futur [29].

On a l'équation : $\forall t \geq 0, V_t^{RP} = V_t^{BE}$ où V_t^{RP} (resp V_t^{BE}) est la valeur du portefeuille répliquant (resp de la provision *Best Estimate*) à l'instant t . Les portefeuilles répliquants permettent ainsi de calculer le SCR via la génération d'un unique niveau de simulation : les simulations monde réel.

Les actifs financiers composant le portefeuille répliquant sont choisis par des experts. Un calibrage est généralement fait via une régression des moindres carrés.

2.2.3 Les avantages et inconvénients de ces méthodes

Après avoir présenté brièvement quelques méthodes d'accélération des SdS, nous pouvons noter qu'elles ont pour objectif d'optimiser le temps de calcul en travaillant sur les axes suivants :

- réduire le nombre de simulations primaires,
- réduire le nombre de simulations secondaires,
- réduire la complexité liée au re-calibrage du GSE,
- réduire la complexité liée à plusieurs des axes précédents (par exemple réduire le nombre de simulations primaires et secondaires).

Un inconvénient des méthodes qui visent à réduire le nombre de scénarios primaires est qu'elles sont difficilement adaptable à d'autres problématiques que celles d'un calcul du SCR. En réduisant le nombre de simulations primaires pour se concentrer sur la zone quantile nous perdons de l'information sur le reste de la distribution. Or, cette information peut s'avérer très utile lors d'une problématique de choix d'une allocation d'actifs par exemple. D'autre part, quelle que soit la méthode d'accélération retenue, sa mise en place effective peut être délicate. Il peut, effectivement, s'avérer délicat de justifier à l'ACPR un calcul de SCR sans passer par le calcul complet obtenu via la méthode des SdS. Dans ces conditions, une méthode alternative trouve davantage d'application pratique lorsqu'elle est utilisée à des fins de choix d'allocation d'actifs. En effet, lors du pilotage de l'allocation d'actifs, il est alors possible de concevoir un outil interne de substitutions aux SdS, afin d'éclairer un choix stratégique. Une fois le choix retenu, il convient alors d'effectuer les calculs complets, via la méthode des SdS pour justifier les résultats. C'est ce que nous allons proposer.

Dans ces conditions, les méthodes de réduction du second niveau de simulation semblent plus intéressantes, car davantage adaptées à une problématique de choix d'allocation d'actifs, tout en gardant un intérêt pour le calcul du SCR. Notons cependant, que ces méthodes alternatives nécessitent généralement un calibrage, ou des avis d'experts et ne sont donc pas nécessairement généralisable à plusieurs allocations d'actifs. Il faut également se poser la question de l'adaptation de ces modèles pour la projection d'un ratio de solvabilité.

Nous allons proposer une méthode basée sur des modèles de *machine learning* qui permette de combler à ces manques. Nous chercherons ainsi à mettre en place un outil capable d'obtenir une estimation robuste du SCR vision modèle interne, tout en étant adaptable à une problématique de choix d'allocation d'actifs et ne nécessitant pas un calibrage des modèles trop complexe.

Chapitre 3

Le machine learning : alternative à la méthode des SdS

« Tous les modèles sont faux, mais certains sont utiles. »

George Box

3.1 Le machine learning

3.1.1 Qu'est ce que le *machine learning* ?

Le *machine learning* est défini par Arthur Samuel en 1959 comme le champ d'étude visant à donner la capacité à une machine d'apprendre sans être explicitement programmée. Cette définition a été complétée par Tom Mitchell en 1997 : « *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E* ». Le développement des capacités informatiques et des données disponibles amènent à un attrait vers cette discipline. Les données sont de plus en plus présentes notamment depuis le développement d'internet et de nombreuses entreprises ont compris l'intérêt financier lié à l'utilisation de ces données. En parallèle, le séquençage de l'ADN humain donne également naissance à de gigantesques bases de données et promet de nombreuses applications à cette discipline dans les années à venir, dans des domaines très variés.

• Les données :

Le *machine learning* consiste à apprendre, sans être explicitement programmé, à partir de données. Ces données contiennent généralement une variable à expliquer et des variables explicatives. Les modèles de *machine learning* apprennent la variable à expliquer en fonction des variables explicatives.

Une fois la phase d'apprentissage effectuée, les modèles sont capables de prédire la variable à expliquer sur une nouvelle base de données contenant les variables explicatives seules.

Nous pouvons chercher à expliquer des données très variées. Il peut s'agir, par exemple, de déterminer si un client d'une banque va rembourser un prêt, quel est le risque qu'un assuré donné ait un accident de la route ou encore si un patient est atteint d'une tumeur. Nous distinguons deux principaux types de données :

- les données quantitatives,
- les données qualitatives.

Les données qualitatives décrivent des qualités ou des caractéristiques. Par exemple, le sexe d'un individu, une couleur de véhicule, etc. Ces valeurs ne sont pas des nombres mais un ensemble de modalités. Nous n'aurons pas à utiliser ce type de données dans notre mémoire.

Les données quantitatives, quant à elles, sont des valeurs qui décrivent une quantité mesurable. Nous pouvons en faire des calculs (moyenne, somme, etc), mais également des comparaisons (égalité, infériorité, etc). On distingue parfois les variables quantitatives discrètes des variables quantitatives continues. Les premières pouvant prendre une infinité de valeurs dans un ensemble de valeurs défini (par exemple le prix d'un appartement) tandis que les secondes ne peuvent en prendre qu'un nombre limité (par exemple le nombre de pièces d'un appartement). Nous n'utiliserons que des variables quantitatives continues dans notre mémoire.

La qualité des modèles de *machine learning* est très fortement corrélée avec la qualité des données utilisées pour la phase d'apprentissage. Il convient donc de faire des études préalables quant à la qualité des données, il faut notamment éviter les valeurs aberrantes et erreurs, mais il faut également veiller au fait d'avoir une base de données suffisamment représentative de la réalité que l'on souhaite modéliser. Il est également important de proposer des variables explicatives pertinentes au modèle et d'éviter les informations redondantes ou inutiles. De nombreux problèmes de *machine learning* sont avant tout des problèmes de traitement des données.

3.1.2 L'apprentissage supervisé

On distingue deux types d'algorithmes : les algorithmes supervisés et non supervisés.

Les algorithmes supervisés extraient de la connaissance à partir de couples variables explicatives / variable à expliquer, ou plus simplement entrées/sortie. Les sorties sont donc déjà connues. Le fonctionnement de ces modèles suppose un lien entre les valeurs d'entrée et de sortie. L'algorithme supervisé vise à apprendre ce lien via différents exemples, pour ensuite être capable de prédire une valeur de sortie à partir des données d'entrées seules.

Les algorithmes non supervisés n'intègrent pas cette notion d'entrée/sortie. Ils visent plutôt à créer des groupes homogènes de données. Ces algorithmes sont souvent moins performants.

Dans ce mémoire, nous allons nous intéresser exclusivement à des méthodes d'apprentissage supervisé. Celles-ci présentent un coût car il faut obtenir une base de données contenant des couples entrées/sortie.

Le principe des algorithmes d'apprentissage supervisé est de supposer qu'il existe une fonction lien f entre les données de sorties Y et les données d'entrée $X = (X_i)_{0 \leq i \leq n}$ avec $\forall i, X_i$ un vecteur de même dimension que Y . On a alors : $Y = f(X) + \epsilon$ avec ϵ un bruit aléatoire centré. On a notamment la relation :

$$E[Y|X_1, \dots, X_n] = f(X_1, \dots, X_n).$$

Les algorithmes d'apprentissage supervisé cherchent à modéliser la fonction lien f . Une des difficultés de l'apprentissage supervisé est d'apprendre la fonction lien, sans apprendre le bruit aléatoire, c'est ce que nous allons présenter avec le dilemme biais-variance.

3.1.3 Le dilemme biais-variance

Les modèles d'estimation commettent deux types d'erreurs lorsqu'ils cherchent à prédire une variable de sortie :

- le biais,
- la variance.

Le biais correspond à l'erreur entre la variable observée et la variable prédite par le modèle. La variance est liée à la généralisation du modèle sur d'autres données.

Un algorithme ayant un biais fort et une variance faible aura tendance à être facilement généralisable à d'autres jeux de données. Il obtiendrait ainsi un taux d'erreur relativement stable quelque soit les données fournies, même si son erreur sera forte (du fait de son biais élevé).

À l'inverse, un algorithme ayant un biais faible et une variance forte obtiendra une capacité de prédiction forte sur le jeu de données qu'on lui aura présenté. Mais si on utilise le modèle sur d'autres données ses performances ne seront pas stables.

À titre d'illustration nous proposons ce schéma issu de l'excellent article « *Understanding the Bias-Variance Tradeoff* » [15] :

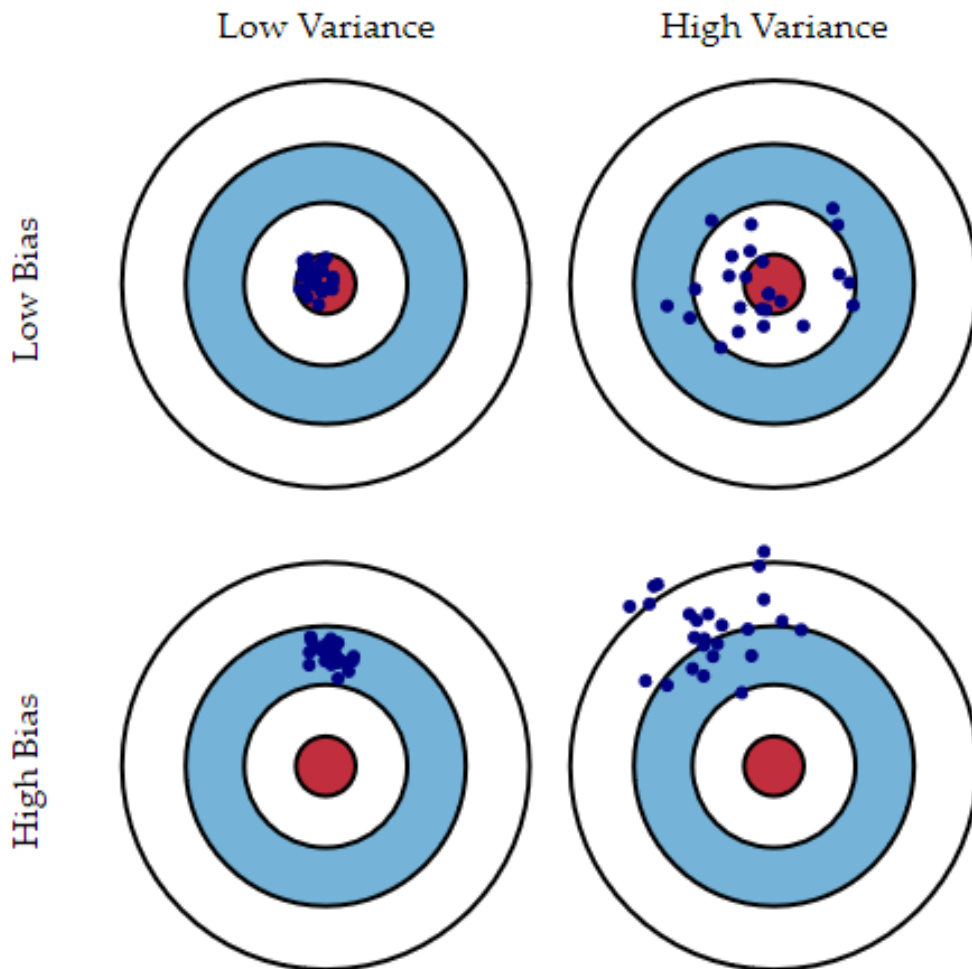


FIGURE 3.1 – Représentation du biais et de la variance d'un modèle

Cette problématique est centrale dans les modèles d'apprentissage supervisé. Généralement, diminuer la variance se fait au prix d'une augmentation du biais et inversement. Il s'agit alors de trouver un compromis acceptable entre les deux : on parle de « compromis biais-variance ».

Supposons qu'il existe une fonction f liant la variable de sortie Y aux données d'entrées X de la manière suivante :

$$Y = f(X) + \epsilon$$

avec :

ϵ un vecteur aléatoire centré de variance $Id \times \sigma^2$.

Notons \hat{f} la fonction de prédiction obtenue par un modèle de *machine learning*. Si on donne une matrice d'entrée X au modèle, il nous prédit ainsi la valeur $\hat{f}(X)$ en sortie. L'erreur commise par le modèle est l'écart entre cette valeur et $f(X)$.

Mathématiquement, nous avons alors la relation suivante :

$$\begin{aligned} E[(f(X) - \hat{f}(X))^2] &= E[f(X)^2 + 2 \times f(X) \times \hat{f}(X) + \hat{f}(X)^2] \\ &= E[f(X)^2] + E[\hat{f}(X)^2] - 2 \times f(X) \times E[\hat{f}(X)] \\ &= Var(f(X)) + E[\hat{f}(X)]^2 + Var(\hat{f}(X)) + E[\hat{f}(X)]^2 - 2 \times f(X) \times E[\hat{f}(X)] \\ &= \sigma^2 + Var(\hat{f}(X)) + Biais(\hat{f}(X))^2 \end{aligned}$$

avec :

$$\begin{aligned} - Biais(\hat{f}(X)) &= (f(X) - \hat{f}(X)), \\ - Var(\hat{f}(X)) &= E[(\hat{f}(X) - E[\hat{f}(X)])^2] = E[\hat{f}(X)^2] - E[\hat{f}(X)]^2. \end{aligned}$$

Ainsi, l'erreur quadratique moyenne commise par le modèle est bien fonction d'une variance et d'un biais tels que décrits précédemment. À cela s'ajoute σ^2 qui est « l'erreur incompressible » que nous allons ignorer.

Nous allons présenter plus en détail le phénomène de « sur-apprentissage ». Cette problématique apparaît souvent lorsque l'on cherche à diminuer le biais des modèles. Nous présenterons également les solutions pour faire face au compromis biais-variance.

3.1.4 Le sur-apprentissage

Nous parlons de sur-apprentissage ou d'« *over-fitting* » pour un modèle supervisé lorsque le modèle obtient d'excellents résultats de prédiction sur les données d'apprentissage fournies, mais qu'il s'avère être mauvais lors de la prédiction de nouvelles données. En étant « trop » proche des données fournies, le modèle se généralise mal : le modèle a un biais faible mais une variance trop élevée.

La notion d'*over-fitting* est le dilemme biais-variance appliqué au *machine learning*. Les modèles supervisés de part leur complexité peuvent obtenir un biais très faible (voire nul) sur les données d'apprentissage.

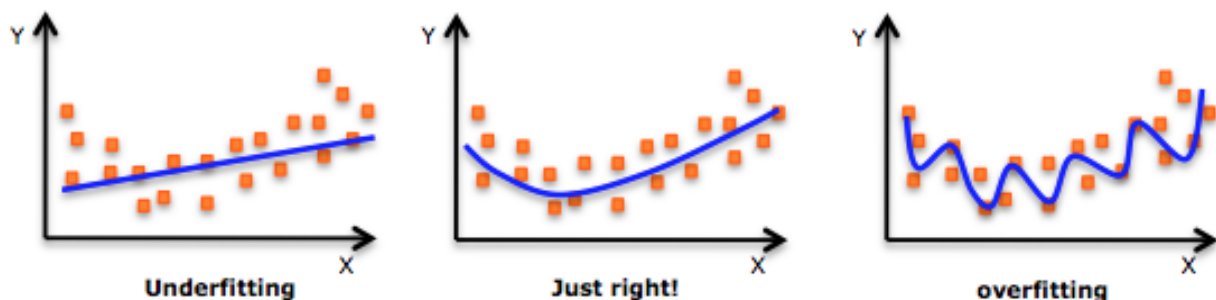


FIGURE 3.2 – Le sur-apprentissage ou « *over-fitting* »

Il s'agit alors de trouver un bon compromis entre la performance obtenue lors de l'apprentissage et la possibilité de généraliser le modèle.

L'augmentation de la variance en *machine learning* va souvent de pair avec l'augmentation de la complexité du modèle :

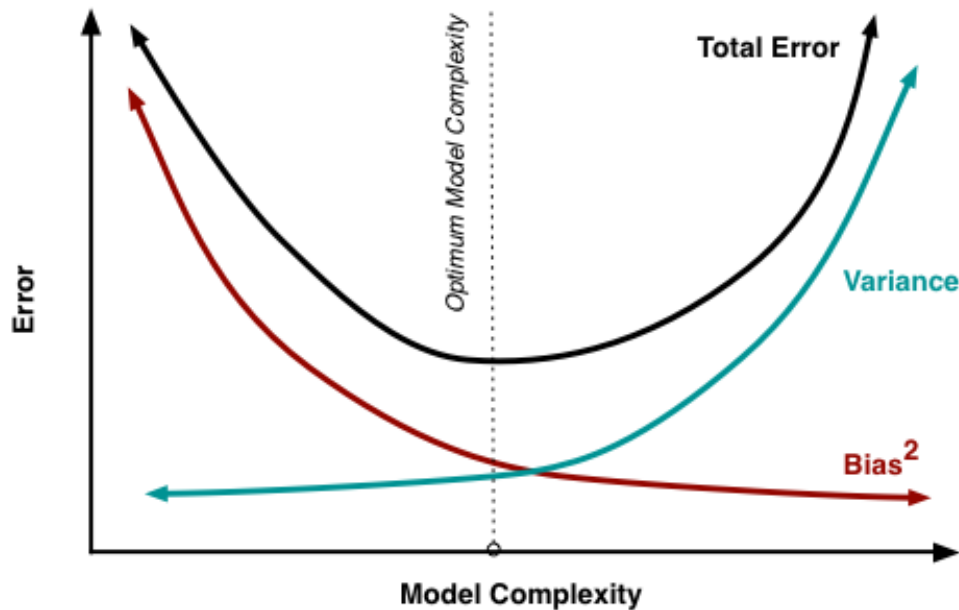


FIGURE 3.3 – Le compromis biais-variance [15]

Pour trouver le point de complexité optimal, il faut tester le modèle sur d'autres données que celles utilisées lors de l'apprentissage afin d'en déterminer sa variance. Le biais peut en effet être mesuré sur les données d'apprentissage et peut généralement être rendu très faible en augmentant la complexité du modèle.

Nous construisons alors deux bases de données : une base d'apprentissage et une base de test. Généralement, nous avons une base de données que l'on divise en deux : Base de test / Base d'apprentissage. Une répartition classique est de 30% des données pour le test du modèle et 70% pour l'apprentissage. Cette proportion peut différer selon le type de problème traité.

L'apprentissage des modèles supervisés se fait sur la base d'apprentissage. Ensuite, les performances sont testées sur la base de test. La base de test contient également des variables entrées/sorties, il s'agit alors de comparer la sortie prédite par le modèle à la variable sortie de la base de test. La mesure de la performance du modèle peut être faite de différentes manières.

- **Un mot sur les indicateurs de performance :**

Un indicateur souvent utilisé est la « MSE » (*Mean Squared Error*) ou « EQM » (Erreur Quadratique Moyenne) en français, il s'agit de la moyenne du carré des erreurs. Si l'on note y_i la valeur observée d'une série à prédire et \hat{y}_i la valeur prédite par le modèle, alors :

$$MSE = \frac{1}{m} \times \sum_{i=1}^m (y_i - \hat{y}_i)^2.$$

On utilise parfois la « RMSE » (*Root Mean Squared Error*) qui est la racine carrée de la MSE. Ces indicateurs présentent l'avantage de pénaliser plus fortement les fortes erreurs (via le carré) que d'autres mesures de performance. Nous utiliserons souvent la MSE durant ce mémoire, mais il existe d'autres indicateurs de performance.

Le choix de la mesure de performance du modèle est particulièrement important. Se contenter d'une mesure de la MSE peut amener à de mauvaises conclusions. Par exemple, dans des problèmes de *scoring* bancaire il s'agit de déterminer si un client est à même de rembourser un emprunt. Les banques choisissent leur client avec attention lorsqu'elles accordent un prêt, le taux de mauvais payeurs qu'elles ont dans leur base de données est ainsi très faible. Considérons une banque qui a dans son historique de clients un taux de mauvais payeurs de l'ordre de 1% et qu'elle décide de mettre en place un modèle de *machine learning* afin de déterminer si un client va être un mauvais payeur potentiel. Refuser un prêt à un bon payeur ou en accorder un à un mauvais payeur présentent des coûts que l'on cherche à éviter. Si l'on se base sur la MSE, un modèle qui prédirait systématiquement que le client est un bon payeur obtiendrait un taux d'erreur de l'ordre de 1% (puisque c'est la répartition de mauvais payeurs dans la base de données de la banque). Cependant, un tel modèle s'avèrerait délétère puisqu'il serait inutile et risquerait d'amener à une forte anti-sélection. À l'inverse, un modèle permettant de déterminer dans 80% des cas si un client est un mauvais payeur et dans 90% des cas si un client est un bon payeur obtiendrait une MSE beaucoup plus élevée que le modèle précédent, mais permettrait de se prémunir bien plus efficacement des mauvais payeurs. On utilise ainsi d'autres mesures de performances dans ce type de problèmes telles que l'AUC par exemple.

De la même manière, dans notre étude la MSE peut présenter un intérêt important, mais également des limites. Par exemple, un modèle qui permet de prévoir les Fonds Propres futurs avec une MSE faible peut être intéressant, mais si l'on cherche à estimer un quantile il faudra considérer d'autres critères : la distribution des Fonds Propres futurs, la MSE dans les scénarios « extrêmes », etc. Nous y reviendrons lors de l'application.

Nous avons vu que l'utilisation de deux bases distinctes pour l'apprentissage et le test du modèle permet de répondre à la problématique du sur-apprentissage. Nous allons présenter la méthode de « *cross validation* » ou validation croisée, qui permet d'optimiser l'apprentissage et les paramètres du modèles afin que celui-ci soit le plus apte à s'adapter à de nouvelles données et obtienne ainsi de meilleures performances sur la base de test.

3.1.5 Utilisation de la validation croisée dans les méthodes d'apprentissage supervisé

L'objectif de la validation croisée est de permettre au modèle d'être le plus généralisable possible. Nous considérons que le modèle se généralise bien s'il obtient une MSE faible sur la base de test.

Une première approche pourrait alors être la suivante : étant donné un modèle, on teste successivement différents jeux de paramètres et nous retenons celui qui permet d'obtenir le meilleur score sur la base de test. Mais cette approche serait alors biaisée, puisque l'on aurait utilisé les données de test pour le choix des paramètres d'apprentissage.

Une réponse à cette problématique est de considérer une troisième base dite « base de validation ». Nous obtenons cette base en divisant la base d'apprentissage en une base d'entraînement et une base de validation. Cette base de validation permet de déterminer le choix du jeu de paramètres « optimal » d'un modèle.

Ainsi, le calibrage d'un modèle peut se faire de la manière suivante :

- pour chaque jeu de paramètre testé :
 - nous effectuons l'apprentissage du modèle sur les données d'entraînement,
 - nous calculons la MSE du modèle sur la base de validation ;
- nous retenons le jeu de paramètres ayant obtenu la MSE la plus faible sur la base de validation et nous faisons apprendre le modèle avec ce jeu de paramètres sur la base d'apprentissage ;
- nous testons la performance de ce modèle sur la base de test.

La validation croisée reprend cette idée pour le choix du jeu de paramètres. Cependant, celle-ci est plus exhaustive. Au lieu de considérer une base d'entraînement et de validation fixées, la validation croisée considère différentes bases d'entraînement et de validation. Cette approche vise à optimiser la généralisation du modèle.

Il existe différents algorithmes de validation croisée, nous allons maintenant en présenter quelques-uns. Nous supposons que la base d'apprentissage contient m couples entrée/sortie. Voici quelques méthodes de validation croisée.

- **La méthode LOOCV (*Leave One Out Cross Validation*) :**

Cette méthode consiste à construire m bases d'entraînement et de validation. La base de validation ne contiendra qu'un unique couple entrée/sortie. Ce couple va successivement être chacun des éléments de la base d'apprentissage. La base d'entraînement sera constituée de la base d'apprentissage à laquelle on aura retiré l'élément de la base de validation. L'erreur du modèle est calculée sur chaque base de validation. Ces m erreurs peuvent être utilisées pour juger de la performance du modèle en validation croisée.

- **La méthode LKOCV (*Leave k Out Cross Validation*) :**

Cette méthode est la même que la précédente, sauf que la base de validation contiendra k couples entrées/sorties et la base d'entraînement correspondante sera la base d'apprentissage à laquelle on aura retiré les k couples de la base de la validation. Le processus sera répété jusqu'à avoir obtenu toutes les bases d'entraînement et de validation possibles en retirant k éléments.

Voici un schéma représentant la méthode LOOCV, qui est un cas particulier de LKOCV pour $k=1$:

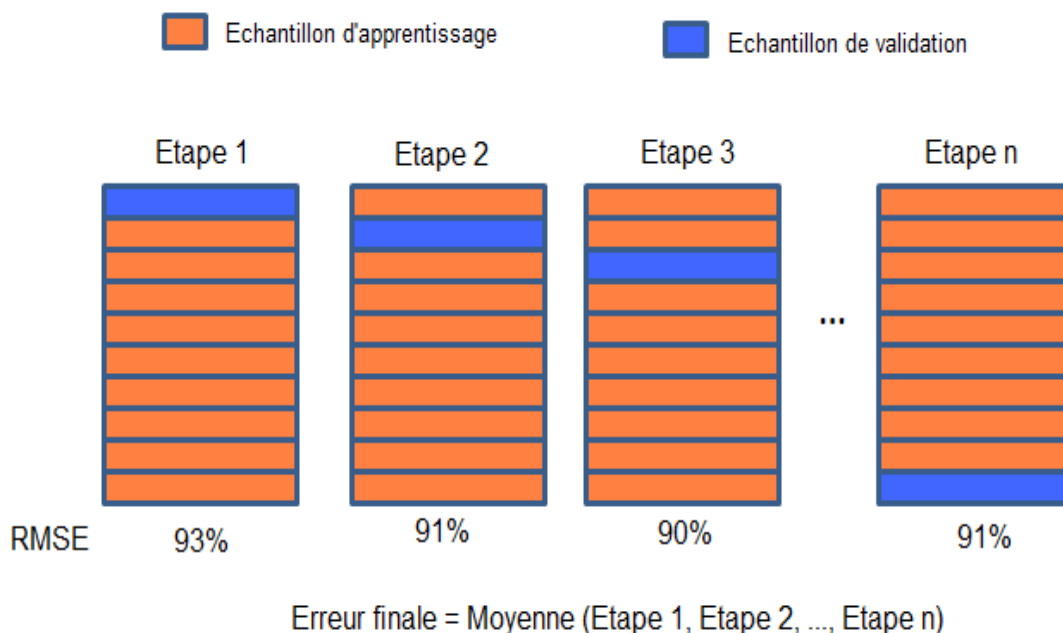


FIGURE 3.4 – Méthode LOOCV

- **La méthode *k-fold Cross Validation* :**

La base d'apprentissage est divisée en k sous échantillons de taille égale, de façon aléatoire. Un de ces échantillons est utilisé en base de validation et les $k-1$ autres servent de base d'entraînement. On répète ce processus k fois. L'erreur du modèle est calculée sur chacune des k bases de validation utilisées.

La méthode LKOCV (notons que LOOCV est un cas particulier de LKOCV) est exhaustive, c'est-à-dire qu'une fois terminée, la méthode a divisé le jeu de données en fonction de toutes les combinaisons possibles. Alors que la méthode k -fold n'est pas exhaustive, car elle ne sépare le jeu de données qu'en un nombre limité de sous-échantillons possibles. La méthode k -fold présente néanmoins l'intérêt d'être plus rapide à mettre en place qu'une méthode exhaustive.

3.1.6 Quelles méthodes d'apprentissage choisir ?

Il est difficile de déterminer a priori quels méthodes d'apprentissage vont s'avérer les plus performantes pour notre étude. Nous allons cependant essayer de sélectionner des méthodes dont nous avons l'intuition qu'elles vont être adaptées à notre problème. Le coût d'apprentissage d'un modèle est relativement faible et nous pourrions donc effectuer ce choix a posteriori. Cependant, la phase de calibration des paramètres de chaque modèle reste coûteuse, nous préférons donc nous concentrer sur certains modèles. Cette phase de calibration s'effectuera par validation croisée (méthode *k-fold cross validation* que nous venons de présenter).

La validation croisée nous permet de déterminer des paramètres « optimaux » pour chaque modèle. Cependant, le jeu de paramètres à tester doit être fourni en entrée par l'utilisateur. Il va donc s'agir de bien comprendre l'influence des différents paramètres sur les performances de chaque modèle. Nous présenterons ci-après les différentes méthodes retenues et nous présenterons les différents paramètres de chacune.

Dans notre étude, nous y reviendrons, nous cherchons à apprendre la provision *Best Estimate* en fonction des paramètres de l'univers économique et de l'allocation d'actifs. Les liens existants entre la provision *Best Estimate* et l'univers économique et l'allocation d'actifs sont complexes. Pour faire face à une telle complexité, nous proposons différentes approches.

Tout d'abord, celle d'une méthode relativement complexe, le *Support Vector Machine* (SVM)¹. Cette méthode permet de modéliser de fortes interactions non linéaires entre les variables explicatives et la variables à expliquer. Elle présente également l'avantage d'être peu sensible au nombre de données présente dans la base d'apprentissage. Il s'agit cependant d'un modèle complexe, dont la compréhension des résultats peu s'avérer délicate.

La seconde approche est la réponse offerte à un problème complexe par une agrégation de modèles simples. Nous parlons alors de méthode ensembliste. Cette approche a été largement plébiscitée récemment : en 2015 sur 29 compétitions « Kaggle »² 17 ont été gagnées par un l'algorithme ensembliste de Boosting : XGBoost [9]. Nous allons débiter le paragraphe suivant par la présentation de cette approche.

Enfin, rappelons que la performance d'une méthode est liée à l'utilisation que l'on souhaite en faire. Par exemple, nous préférons une méthode performante à estimer les provisions *Best Estimate* dans les scénarios extrêmes lorsque nous cherchons à déterminer un SCR en modèle interne, quitte à

1. Le SVM est un modèle de classification. Nous utiliserons sa version adaptée à la régression : *Support Vector Regressor Machine* (SVRM). Il s'agit d'une généralisation du modèle SVM à une problématique de régression, aussi nous pourrions parfois utiliser le terme SVM pour désigner le SVRM. Il n'y a pas de confusion possible lors de notre étude nous utiliserons uniquement la version adaptée à la régression.

2. Kaggle est un site proposant de résoudre différents problèmes liés à l'utilisation de données pour des banques et assureurs principalement.

obtenir de moins bonnes performances dans les scénarios centraux. En revanche, si nous cherchons à déterminer une allocation d'actifs en fonction des Fonds Propres moyens futurs et de leur volatilité, le critère de performance pourra être différent et nous pourrions préférer une autre méthode.

• Les méthodes ensemblistes :

L'idée des méthodes ensemblistes est qu'une agrégation de modèles peut s'avérer plus performante qu'un unique modèle complexe.

À titre d'illustration cette idée est notamment mise en place dans les jurés d'assise où l'on préfère avoir plusieurs jurés tirés au sort n'ayant potentiellement aucune notion de droit plutôt que l'avis d'un unique spécialiste.

Les méthodes ensemblistes permettent de répondre au dilemme biais-variance. Il existe deux approches distinctes. On peut partir d'un modèle simple ayant un biais élevé et une variance faible, on agrège ensuite un second modèle afin de réduire le biais du modèle précédent, et ainsi de suite. Ou alors on peut partir de modèles plus complexes au biais faible et à la variance élevée et chercher à diminuer la variance en agrégeant de manière indépendantes plusieurs modèles.

Les méthodes de boosting reposent sur la première idée : plusieurs « weak learner »³ peuvent devenir un « *strong learner* » si on les agrège. Au début des années 1990, Robert Schapire et Yoav Freund ont réalisé des travaux sur les « *weak learner* ». Schapire a notamment démontré qu'il était possible de « booster » un algorithme faible en un méta-algorithme robuste.

Le *Random Forest* quant à lui s'inspire de la seconde idée et vise à agréger plusieurs modèles complexes indépendants afin d'en réduire la variance.

Les deux approches nous semblent intéressantes pour répondre à notre problématique et nous allons donc proposer un algorithme de Boosting : XGBoost et un algorithme de *Random Forest*. Les deux algorithmes agrègent des arbres de décision. XGBoost va les construire en série : chaque arbre aura connaissance de son prédécesseur, alors que le *Random Forest* va construire des arbres en parallèle de manière indépendante. Après avoir expliqué notre choix, nous allons maintenant présenter les modèles et méthodes utilisés plus en détail.

3.2 Présentation des méthodes d'apprentissage supervisé retenues

Tout d'abord, nous allons présenter un modèle très simple : les arbres de décisions. Ce modèle ne sera pas implémenté lors des applications numériques, mais étant à la base de plusieurs modèles que nous allons utiliser, il nous a semblé utile de l'introduire.

3.2.1 Les arbres de décision à la base des méthodes ensemblistes

Les arbres de décisions permettent d'expliquer une variable aussi bien quantitative que qualitative. Dans le cas de variables quantitatives on parle d'arbre de régression.

Ces méthodes présentent de nombreux atouts : elles sont performantes, non linéaires, non paramétriques et permettent généralement de bien visualiser et comprendre les résultats ce qui présente un avantage indéniable sur d'autres modèles plus complexes. Les arbres de décision peuvent cependant avoir une forte dépendance aux données d'apprentissage et donc une variance élevée.

3. Un « *weak learner* » est un classifieur ou régresseur faible. Il s'agit généralement d'un modèle simple qui aura un biais élevé et une variance faible. On l'oppose à « *strong learner* » qui est un modèle ayant un biais faible.

Les arbres de décisions permettent de partitionner les variables explicatives en groupes homogènes en fonction de la variable à prédire. Ces groupes sont formés en prenant en compte une hiérarchie basée sur la capacité prédictive des variables explicatives.

L'algorithme d'un arbre de décision fonctionne selon le principe suivant :

- On divise les individus en k classes (souvent $k=2$) pour expliquer la variable de sortie. La première division est obtenue en choisissant la variable explicative qui fournit la meilleure explication de la variable de sortie. L'échantillon initial est alors divisé en k sous-échantillons de populations. Ces sous-populations définissent des noeuds de l'arbre. À chaque noeud est associé une mesure de proportion, qui permet d'expliquer l'appartenance à une classe ou la signification d'une variable de sortie.
- On recommence : chaque sous-population est à nouveau divisée selon la variable la plus pertinente.
- On continue jusqu'à ce que plus aucune séparation ne soit possible. On obtient alors des noeuds terminaux, appelés « feuilles » de l'arbre.

Différentes mesures de proportions existent, citons à titre d'exemple l'indice de Gini qui est souvent utilisé (notamment dans l'algorithme « CART » *Classification and Regression Tree* inventé par Breiman et al. en 1984).

Dans le cas d'arbre de régression, l'objectif est de minimiser la variance intra-classe de la variable de sortie. Par exemple, considérons l'ensemble A contenant le couple entrée/sortie (X, Y) . On cherche alors à minimiser $G(A) = V^A(Y)$ où :

$$V^A(Y) = \frac{1}{\#A} \times \sum_{i|x_i \in A} (Y_i - \bar{Y}_A)^2$$

avec :

$$\bar{Y}_A = \frac{1}{\#A} \times \sum_{i|x_i \in A} Y_i.$$

Voici un exemple d'arbre de régression issu du livre « Data Science : Fondamentaux et études de cas » par Eric Biernat et Michel Lutz [2] :

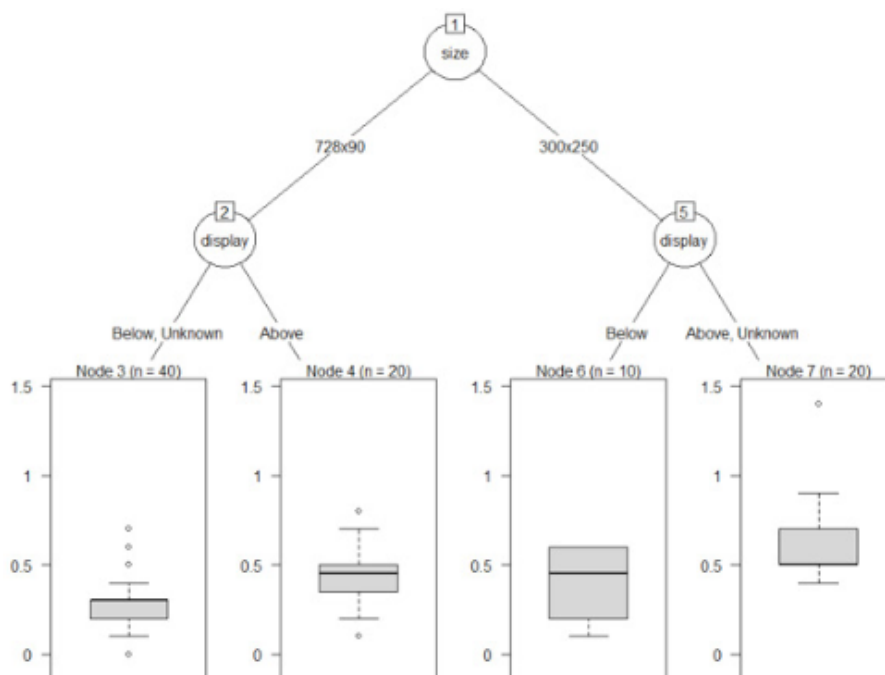


FIGURE 3.5 – Exemple d'arbre de décision pour des données quantitatives

Cet arbre représente la distribution du taux de clic sur une bannière publicitaire en fonction de différentes règles : la taille de la publicité et la position de la publicité.

Si nous laissons un arbre de décision grandir sans critère d'arrêt il peut y avoir une feuille par donnée de sortie. En ce cas la base d'apprentissage est parfaitement décrite, le biais est nul, mais l'arbre est fortement dépendant des données d'apprentissage et la variance est très forte. Pour éviter cette situation, des méthodes d'« élagage » ont été élaborées. Il est possible de choisir un critère d'arrêt, ou d'élaguer l'arbre (c'est à dire supprimer des feuilles en regroupant les données) a posteriori.

Le but de l'élagage est ainsi d'obtenir un compromis biais-variance acceptable. Nous allons présenter brièvement l'algorithme issu des travaux de Breiman [6] : il s'agit de l'algorithme CART (*Classification and Regression Tree*). Cet algorithme a notamment été adapté dans le mémoire « Évaluation et optimisation de la rentabilité d'un portefeuille automobile » [14], il y est présenté de façon plus exhaustive. Nous renvoyons également le lecteur à l'article original de Breiman [5] pour plus d'informations sur cet algorithme.

L'algorithme CART procède en plusieurs étapes.

Tout d'abord, CART construit l'arbre de taille maximale : chaque feuille de cet arbre ne contient alors qu'une unique observation. Pour ce faire, il divise à chaque étape les données en deux sous-échantillon. La règle de division est choisie selon un critère d'homogénéité : à chaque étape on choisit la règle de division qui permet d'obtenir les groupes de données les plus homogènes. On cherche en particulier à ce que dans chaque sous-échantillon les données soient le moins dispersées possibles. Une fois cette étape terminée, l'arbre final ne contient qu'une unique donnée par feuille et le biais est alors nul.

CART propose ensuite un élagage de l'arbre saturé. Il vise à supprimer des séparations afin de regrouper des données, en partant des feuilles vers les racines. Une nouvelle mesure d'erreur est mise en place : il s'agit de « l'erreur complexité ».

• La notion de fonction objectif :

Les algorithmes supervisés visent à obtenir la meilleure performance possible lors de leur construction selon une « fonction objectif ». Par exemple, tel que présenté précédemment, l'arbre de décision minimise la fonction objectif qui est le critère de Gini. Cette mesure ne tient pas compte de la complexité du modèle et amène à une variance potentiellement trop élevée.

Pour en tenir compte, les fonctions objectif ont deux composantes : une fonction de coût α et un paramètre de régularisation β . Ainsi, la fonction objectif Ω est définie de la manière suivante :

$$\Omega(\Theta) = \alpha(\Theta) + \beta(\Theta)$$

avec :

Θ le jeu de paramètres retenu.

L'objectif est de déterminer les paramètres Θ du modèle qui minimisent la fonction objectif. En complexifiant le modèle on diminue souvent la fonction de coût au détriment du paramètre de régularisation.

Pour l'erreur complexité, le paramètre de régularisation est le suivant : $\alpha \times$ nombre de feuilles.

Ainsi, si α est nul, l'arbre maximal minimise l'erreur complexité. À l'inverse, un α élevé va favoriser des arbres « simples » contenant peu de subdivisions et donc de feuilles. Le paramètre α va ainsi pénaliser plus ou moins le biais au profit de la variance.

L'algorithme CART détermine alors avec cette mesure d'erreur complexité une suite de sous arbre optimaux minimisant l'erreur complexité pour α qui varie en partant de 0. Dans cette suite d'arbre optimaux, CART détermine l'arbre optimal en le testant sur une base de validation. L'arbre qui minimise l'erreur sur la base de validation est alors retenu.

Nous allons maintenant présenter deux méthodes ensemblistes basées sur des agrégations d'arbres de décision. Le premier est le *Random Forest* (ou forêts aléatoires) il repose sur des stratégies aléatoires. Tandis que le second, XGBoost, repose sur une stratégie adaptative. Ces méthodes permettent de combler l'un des principaux défauts des arbres de décision : la dépendance de l'arbre de décision aux données d'apprentissage. L'approche est cependant différente, dans le cadre de forêts aléatoires on utilisera des arbres développés de faible biais et nous chercherons à réduire la variance, tandis que pour XGBoost, nous agrégerons des arbres plus simples de faible variance et nous chercherons à diminuer le biais du modèle.

3.2.2 Les *Random Forest*

Les *Random Forest* consistent à utiliser plusieurs arbres de décision pour en faire des « forêts ». Cette méthode peut être décomposée en deux parties : « *Tree Bagging* » et « *Feature sampling* ». Nous allons tout d'abord nous concentrer sur le « *Tree Bagging* ».

• Le *Tree Bagging* :

Le « *Bootstrap AGGREGatING* » ou « *Bagging* » a été introduit par Leo Breiman en 1994. Il s'agit d'un méta-algorithme visant à augmenter la précision des modèles tout en réduisant leur variance.

Considérons une base d'apprentissage constituée des couples entrée/sortie (X, Y) . Avec X une matrice $m \times n$ et Y un vecteur de taille m . Le *Tree Bagging* consiste à générer un nombre B d'arbres de décision, de la manière suivante :

- tirer aléatoirement avec remise B échantillons de (X, Y) , notés (X_b, Y_b) ,
- pour chaque échantillon (X_b, Y_b) , entraîner un arbre de décision.

Nous obtenons ainsi B arbres de décision a priori différents (car entraînés sur des données différentes). Lorsque l'on utilise la méthode de *Bagging* pour prédire la variable de sortie sur d'autres données, la valeur prédite est alors la moyenne de la valeur prédite par chacun des B arbres.

• Le *Feature sampling* :

Le *Feature sampling* est proche du *Bagging*, toutefois au lieu de proposer des observations aléatoires à chaque arbre, il s'agit de proposer des variables explicatives aléatoires. Ainsi, le *Feature sampling* consiste à construire différents arbres de décision en ne donnant à chaque modèle qu'un nombre restreint de variables explicatives sélectionnées aléatoirement.

Le *Random Forest* peut ainsi être vu comme une méthode suivant les principes du *Bagging* et du *Feature sampling*. Chaque arbre généré par le *Random Forest* a ainsi une vision parcellaire du problème traité avec des variables explicatives et des observations différentes. Ceci implique notamment un travail au niveau du traitement des variables explicatives. En effet, dans le cadre de la mise en place

d'un *Random Forest* il faudra veiller à éviter de fournir au modèle des variables non pertinentes.

Le *Bagging* et le *Feature sampling* permettent de réduire la variance. Si l'on considère B variables identiquement distribuées, de variance σ^2 avec une corrélation ρ des variables deux à deux, alors la variance des B variables est :

$$\rho \times \sigma^2 + \frac{1 - \rho}{B} \times \sigma^2$$

Si ρ est élevé, le premier terme va être limité via le *Feature sampling*. Si ρ est faible, le second terme sera limité par le *Bagging*.

• Le *Random Forest* en pratique :

Sous Python, le *Random Forest* est implémenté dans le package *scikit-learn*. Nous utilisons le *Random Forest Regressor* adapté, comme nous l'avons vu, à des problématiques de variables explicatives quantitatives, mais il existe également un *Random Forest Classifier* pour la classification. Son implémentation fait appel à 14 paramètres qui seront optimisés par la fonction *grid-search* qui met en place une validation croisée. Cependant, il n'est pas possible de tester une infinité de jeux de paramètres différents, il faut donc comprendre l'impact de chaque paramètre sur la méthode avant de proposer une liste de paramètres à tester lors de la validation croisée. Nous allons nous contenter de présenter les paramètres les plus importants lors de l'implémentation d'un modèle *Random Forest*. Les autres paramètres sont présentés plus en détail via l'aide du package Python.

Voici les principaux paramètres :

- `n_estimators` : le nombre d'arbres à entraîner,
- `max_depth` : la profondeur maximale de chaque arbre,
- `min_sample_split` : le nombre minimal d'observations par feuille, il s'agit d'un critère d'arrêt,
- `max_features` : le nombre de variables tirées aléatoirement pour chaque arbre lors du *Feature sampling*, par défaut il vaut \sqrt{n} ,
- `criterion` : le critère utilisé pour couper les feuilles de chaque arbre en cours de construction, par défaut ce critère est la MSE, nous ne modifierons pas ce paramètre.

3.2.3 Les méthodes de *Boosting*

Le *Boosting* comme le *Random Forest* se base sur une agrégation d'arbres. Contrairement au *Random Forest*, le *Boosting* génère des arbres en séries, c'est à dire que chaque arbre généré (excepté le premier) a accès à son prédécesseur et notamment à l'erreur commise par son prédécesseur. Le nouvel arbre aura alors pour objectif de combler les lacunes de son prédécesseur en donnant plus de poids aux données mal prédites. Le *Boosting* concentre ainsi ses efforts sur les observations les plus difficiles à ajuster tandis que l'agrégation de l'ensemble des modèles permet d'éviter le sur-apprentissage.

Il existe différents algorithmes de *Boosting*, qui diffèrent selon leur manière de pondérer pour renforcer l'apprentissage des données mal ajustées, leur objectif (notamment classification/régression), la fonction de perte pour mesurer l'ajustement et la façon d'agréger les modèles successifs.

• Le *Gradient Boosting* :

Nous allons présenter le principe de la construction d'un modèle de *Gradient Boosting*. Cette méthode consiste à agréger des modèles, ici nous allons nous intéresser à une agrégation d'arbres.

Nous notons B_k l'arbre construit à l'étape k , $f_k = \sum_{j=1}^k B_j$ le modèle à l'étape k . On considère la base d'apprentissage composée des couples entrée/sortie (X, Y) . On notera par simplification X_i toutes les

données de la ligne i de la matrice X . Ainsi, à l'étape k , notre modèle nous donne l'estimation $f_k(X)$ de Y . Nous prenons comme fonction de coût la fonction des moindres carrés définie de la manière suivante :

$$J = \sum_{i=1}^m j(Y_i, f(X_i))$$

avec :

$$j(a, b) = \frac{(b-a)^2}{2}.$$

Le gradient de J par rapport à $f(X_i)$ est :

$$\frac{\partial J}{\partial f(X_i)} = \frac{\partial \sum_{l=1}^m j(Y_l, f(X_l))}{\partial f(X_i)} = f(X_i) - Y_i.$$

Le modèle *Gradient Boosting* se fait alors de la manière suivante :

- Nous construisons un premier arbre de décision B_1 , visant à prédire Y à partir de X . On a alors : $B_1(X_i) = Y_i + (B_1(X_i) - Y_i)$. On appelle résidus la valeur observée des insuffisances du modèle. Pour B_1 , on a : $res_1 = -\frac{\partial J}{\partial f(X_i)} = Y_i - f_1(X_i)$.
- Pour combler les manques du premier arbre, on construit B_2 qui vise à prédire les résidus de l'étape précédente, ie : $-\frac{\partial J}{\partial f(x_i)}$. Une fois le deuxième arbre construit, notre modèle est donc égal à : $f_2(X) = B_1(X) + B_2(X)$. Nous avons $B_2(X_i) = (Y_i - f_1(X_i)) + (f_2(X_i) - Y_i)$. Les résidus de l'étape 2 sont donc : $res_2 = -\frac{\partial J}{\partial f(X_i)} = Y_i - f_2(X_i)$.
- Pour combler les manques du deuxième arbre, on construit B_3 qui vise à prédire res_2 .
- On itère cette procédure, suivant la descente de gradient. À l'étape k , on construit B_k qui cherche à prédire $res_{k-1} = -\frac{\partial J}{\partial f(X_i)} = Y_i - f_{k-1}(X_i)$.

• Une variante : XGBoost

XGBoost est une version « extrême » du *Gradient Boosting* que nous avons présenté⁴. Il s'agit d'un algorithme particulièrement populaire à l'heure actuelle. Il obtient, en effet, d'excellentes performances dans un nombre très varié de situations, en classification comme en régression. En 2015, sur 29 compétitions Kaggle, 17 vainqueurs utilisaient XGBoost.

Cet algorithme a été présenté en 2015 par Tianqi Chen dans l'article « Greedy Function Approximation : A Gradient Boosting Machine » [9]. Une version simplifiée présentant cet algorithme est également disponible sur le site internet dédié à XGBoost [10]. Nous nous basons donc sur ces travaux.

XGBoost permet de généraliser le Gradient Boosting à d'autres fonctions que des arbres de régression. Nous allons cependant présenter XGBoost dans le cadre d'arbres de régression avec comme fonction de perte l , la MSE. On considère les couples entrée/sortie (X, Y) . De la même manière que dans le Gradient Boosting, XGBoost construit des arbres en séries en vue de minimiser le biais, tout en contrôlant la variance.

Nous notons B_k l'arbre construit à l'étape k . Nous avons ainsi à l'étape t , la prédiction du modèle pour : $\widehat{Y}_i^{(t)} = \widehat{Y}_i^{(t-1)} + \eta B_t(X_i)$. η étant un paramètre permettant de diminuer le phénomène de sur-apprentissage lorsque le nombre d'arbres est élevé, on peut diminuer la valeur de ce paramètre. Nous allons présenter XGBoost dans le cas particulier où $\eta = 1$.

4. Le nom XGBoost est l'acronyme pour eXtreme *Gradient Boosting*.

À chaque étape, l'arbre à ajouter est l'arbre qui va minimiser la fonction objectif. Nous avons vu qu'elle se décompose en une fonction de coût plus un paramètre de régularisation. Nous avons donc à l'étape t :

$$\Omega^{(t)}(\Theta) = \alpha^{(t)}(\Theta) + \beta^{(t)}(\Theta) = \sum_{j=1}^n l(\widehat{Y}_i^{(t)}, Y_i) + \sum_{j=1}^K \beta(B_j).$$

Pour XGBoost, puisque nous prenons la MSE comme fonction de coût l , on utilise :

$$\alpha^{(t)}(\Theta) = \sum_{j=1}^n l(\widehat{Y}_i^{(t)}, Y_i) = \sum_{j=1}^n (\widehat{Y}_i^{(t)} - Y_i)^2.$$

La fonction de régularisation utilisée est la suivante :

$$\beta(B_t) = \gamma \times T + \frac{1}{2} \times [\alpha \times \sum_{j=1}^T |c_j| + \lambda \times \sum_{j=1}^T c_j^2]$$

avec :

- γ, α, λ des paramètres de pénalisation de la complexité du modèle (nous chercherons à optimiser le choix de ces paramètres lors de la phase de calibration par validation croisée),
- T le nombre de feuilles de l'arbre B_t ,
- c_j la valeur associée à chaque feuille de l'arbre.

Ainsi, à chaque étape, il s'agit d'ajouter un arbre afin de minimiser la fonction objectif. Nous nous plaçons dans le cas particulier où le paramètre de pénalisation α est nul. À l'étape t , la fonction objectif est :

$$\Omega^{(t)} = \sum_{i=1}^n (\widehat{Y}_i^{(t-1)} + B_t(X_i) - Y_i)^2 + \beta(B_t).$$

En pratique, on passe par un développement de Taylor à l'ordre 2 afin d'approcher la fonction objectif :

$$\Omega^{(t)} \approx \sum_{j=1}^n [g_i \times B_t(X_i) + \frac{1}{2} \times h_i \times B_t^2(X_i)] + \beta(B_t)$$

avec :

- $g_i := \frac{\partial l(\widehat{Y}_i^{(t-1)}, Y_i)}{\partial Y_i^{(t-1)}} = 2 \times (\widehat{Y}_i^{(t-1)} - Y_i)$
- $h_i := \frac{\partial^2 l(\widehat{Y}_i^{(t-1)}, Y_i)}{\partial^2 Y_i^{(t-1)}} = 2.$

D'où :

$$\Omega^{(t)} \approx \sum_{j=1}^n [2 \times [(Y_i - \widehat{Y}_i^{(t-1)}) \times B_t(X_i) + B_t^2(X_i)] + \beta(B_t).$$

On définit $I_j = \{i \text{ tel que } q(X_i) = j \text{ ie tel que } X_i \text{ appartienne à la feuille numéro } j\}$: l'ensemble des éléments appartenant à la feuille j . Sur cet ensemble, la fonction objectif devient [9] :

$$\Omega^{(t)} \approx \sum_{j=1}^T [G_j \times w_j + \frac{1}{2} \times (H_j + \lambda) \times w_j^2] + \gamma \times T$$

avec :

- w_j la valeur associée à chaque feuille,
- $G_j = \sum_{i \in I_j} g_i$,
- $H_j = \sum_{i \in I_j} h_i$.

Les w_j sont indépendants les uns par rapport aux autres et $G_j \times w_j + \frac{1}{2} \times (H_j + \lambda) \times w_j^2$ est une forme quadratique. Pour une structure d'arbre q fixe, on a alors la solution suivante :

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$

$$\Omega^* = -\frac{1}{2} \times \sum_{j=1}^t \frac{G_j^2}{H_j + \lambda} + \gamma \times T.$$

La dernière équation permet de mesurer à quel point la structure $q(x)$ est performante. Cependant, il est impossible de comparer toutes les structures d'arbres possible (il y en a une infinité). En pratique, nous débutons avec un arbre de profondeur nulle et ajoutons des noeuds de manière récursive. Lorsque l'on ajoute un noeud à une feuille, nous découpons alors la feuille en deux feuilles. La baisse de la fonction objectif est alors la suivante :

$$\frac{1}{2} \times \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma$$

avec :

- $\frac{G_L^2}{H_L + \lambda}$ (resp $\frac{G_R^2}{H_R + \lambda}$) le score sur la partie gauche (resp droite) de la séparation,
- $\frac{(G_L + G_R)^2}{H_L + H_R + \lambda}$ le score sur la feuille originale de l'arbre, avant séparation,
- γ le surplus de complexité du modèle induit par la séparation.

Nous pouvons donc mesurer cette baisse pour chaque variable. Pour chaque variable, un parcours linéaire permet de déterminer la meilleure séparation. On ré-itére cette procédure sur tous les noeuds tant qu'il est possible d'améliorer la fonction objectif.

• XGBoost en pratique :

Dans ce mémoire, nous utiliserons l'algorithme « XGBRegressor » du package « xgboost » de Python, qui est l'algorithme que nous venons de présenter. Lors du calibrage nous chercherons à optimiser les paramètres suivants :

- `n_estimators` : nombre d'arbres,
- `learning_rate` : correspond au paramètre η présenté plus haut,
- γ : baisse minimale de la fonction objectif pour faire une subdivision supplémentaire,
- `max_depth` : profondeur maximale d'un arbre,
- `min_child_weight` : nombre minimum d'observations par feuille,
- les paramètres suivants sont optionnels, leur valeur est 1 par défaut, ils permettent de tirer aléatoirement des observations et variables (de la même manière que dans le *Random Forest*) :
 - `subsample` : ratio d'échantillonnage des observations,
 - `colsample_bytree` : ratio d'échantillonnage du nombre de variables tirées aléatoirement pour chaque arbre,
 - `colsample_bylevel` : ratio d'échantillonnage du nombre de variables tirées aléatoirement pour chaque séparation de noeud,
- `reg_lambda` : paramètre de régularisation λ présenté plus haut,
- `reg_alpha` : paramètre de régularisation α présenté plus haut.

• Paramètre d'importance des variables :

Les deux méthodes ensemblistes que nous allons utiliser le *Random Forest* et XGBoost permettent de visualiser les variables les plus discriminantes. Cette information est précieuse car ces modèles sont capables de détecter des interactions entre variables qu'un test univarié d'importance n'aurait pas nécessairement révélé. Nous y prêterons une attention particulière lors de l'étude des résultats de nos

modèles. Il pourra également être intéressant d'étudier si les variables les plus discriminantes sont les mêmes pour les deux méthodes. Dans le cas où elles différeraient, nous pourrions imaginer qu'un modèle pourrait être plus à même de prédire le BE dans certaines situations et inversement.

3.2.4 Les SVM

Les *Support Vector Machine* (SVM) souvent traduits par Séparateurs à Vaste Marge ou encore Machine à Vecteur de Support en français sont une classe d'algorithmes développés dans les années 90. Ils sont principalement issus de la théorie de Vapnik-Chervonenskis. Les SVM sont des classificateurs permettant de traiter des problèmes de discrimination non linéaire. En 1996, Vapnik, Drucker, Kaufman et Smola proposent une méthode pour adapter les SVM aux problèmes de régression. On parle alors de *Support Vector Regressor Machine* (SVRM) ou plus simplement SVR. Nous allons nous intéresser à ce modèle dans le cadre de ce mémoire.

Les SVM présentent de nombreux avantages, mais sont plus complexes que les algorithmes présentés précédemment. Pour faciliter la compréhension des modèles, nous présenterons tout d'abord l'algorithme des SVM pour la classification dans un cadre simple : celui d'un espace linéairement séparable.

Pour présenter cette classe d'algorithmes complexe, nous nous basons sur différents travaux. Nous pouvons ainsi citer le cours « Apprentissage Statistique : une approche connexionniste » [32], le livre « Data Science : Fondamentaux et études de cas » [2], le mémoire « Utilisation des supports vecteurs machines pour l'accélération du calcul du capital économique » [26]. Enfin, les travaux originaux de Vapnik vers lesquels le lecteur curieux pourra se tourner pour davantage de précision.

• La dimension Vapnick-Chevronenskis (VC) :

La dimension Vapnick-Chevronenskis (VC) permet de mesurer la complexité d'un problème. Un classifieur SVM ayant une dimension VC élevée aura un biais faible ou nul mais une variance plus élevée. Inversement, un classifieur de faible dimension VC aura une variance plus faible. Nous pouvons assimiler la dimension VC au degré de liberté de notre modèle.

On dit qu'un modèle de classification C de paramètre Θ pulvérise un ensemble de données (x_1, x_2, \dots, x_n) si pour tout étiquetage de cet ensemble de données, il existe Θ tel que f ne fasse aucune erreur dans l'évaluation de cet ensemble de données. On a alors la dimension VC notée d_{VC} qui vaut :

$$d_{VC} = \max\{\text{card}(E) \text{ tel que } f \text{ pulvérise } E\}.$$

La dimension VC nous permet de quantifier l'erreur de généralisation du modèle par la formule suivante :

$$E[\epsilon_{test} \leq \epsilon_{train} + \sqrt{\frac{d_{VC} \times (\log(\frac{2m}{d_{VC}}) + 1) - \log(\frac{\eta}{4})}{m}}] = 1 - \eta$$

avec :

- ϵ_{train} (respectivement ϵ_{test}) le pourcentage d'erreur commis sur la base d'apprentissage (respectivement de test),
- m est la taille de l'échantillon d'apprentissage,
- d_{VC} est la dimension VC.

N.B : Cette formule n'est valable que lorsque $d_{VC} < m$.

La dimension VC est une mesure de la capacité d'un algorithme de classification. De manière générale, la capacité d'un modèle correspond à sa complexité. Plus la capacité d'un modèle est faible plus

sa variance sera faible [3].

Nous allons maintenant présenter le SVM pour un problème de classification. Nous considérons une base d'apprentissage $B_t = (x^r, y^r) \in \mathbb{R}^n \times \{-1, 1\}$, $r = 1, \dots, p$. Nous nous plaçons ainsi dans le cadre d'une classification binaire, où y appartient à deux classes : positif ou négatif, selon qu'il vaut 1 ou -1 (ces deux classes seront représentées respectivement par des « croix » ou des « ronds » dans nos représentations graphiques).

Un hyperplan H de \mathbb{R}^n est défini par l'équation :

$$H : \langle w, x \rangle + b = 0$$

avec :

$\langle w, x \rangle$ le produit scalaire de \mathbb{R}^n .

• **Les SVM dans le cadre d'un problème linéairement séparable :**

Plaçons nous tout d'abord dans le cas où B_t est linéairement séparable. B_t est linéairement séparable s'il existe $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ tels que $\forall 0 \leq r \leq p$:

$$\begin{cases} \langle w, x^r \rangle + b \geq 1, & \text{si } y^r = 1 \\ \langle w, x^r \rangle - b \leq -1, & \text{si } y^r = -1 \end{cases}$$

Ou de manière équivalente : $y^r(\langle w, x^r \rangle + b) \geq 1$.

Le SVM cherche à construire un hyperplan qui va permettre de séparer nos deux classes. Lorsque la base d'apprentissage est linéairement séparable, il existe une infinité d'hyperplans permettant de faire une telle séparation. Prenons un exemple avec un jeu de données composé de deux classes : « rond » et « croix » :

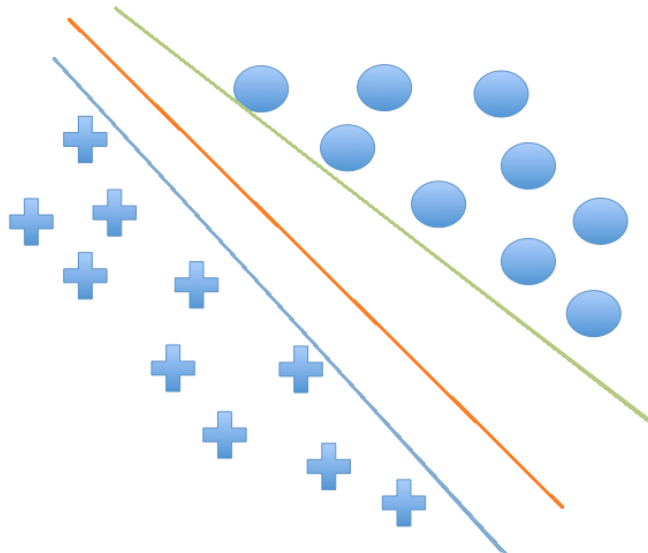


FIGURE 3.6 – Exemple d'hyperplans permettant de classifier des données

Nous cherchons un hyperplan permettant de séparer les deux classes représentées par des croix et des ronds. Nous pouvons effectuer une infinité de séparations de ce type, nous en avons représenté trois. Aucun des hyperplans ne commet d'erreur sur la base d'apprentissage. Cependant, il semble plus naturel de porter notre choix sur l'hyperplan orange. On imagine facilement que le modèle vert risque

d'être moins performant sur d'autres jeux de données pour prédire des « ronds » et l'hyperplan bleu moins performant à prédire des « croix ».

Pour répondre à cette problématique, nous cherchons à déterminer l'hyperplan de marge maximale. Vapnik a, en effet, montré que la capacité des classes d'hyperplans séparateurs diminue lorsque leur marge augmente ce qui augmente leur capacité de généralisation. Il a également montré qu'il existe un unique hyperplan optimal défini comme l'hyperplan qui maximise la marge [31].

La marge est la somme de la marge positive d_+ et la marge négative d_- . La marge positive est la plus petite distance entre l'ensemble des exemples positifs (des « croix » dans notre cas) et l'hyperplan. La marge négative est la plus petite distance entre l'ensemble des exemples négatifs (des « ronds » dans notre cas) et l'hyperplan. La distance d'un point x au plan H d'équation $\langle w, x \rangle + b = 0$ est :

$$d(x, H) = | \langle x - w_1, w_0 \rangle |$$

avec :

- w_1 l'unique point de H tel que w_1 soit co-linéaire au vecteur w ,
- $w_0 = \frac{w}{\|w\|}$.

On a : $w_1 \in H$ ie $\langle w_1, w \rangle + b = 0$

d'où : $\langle w_1, w_0 \rangle = \frac{-b}{\|w\|}$.

On a alors : $d(x, H) = | \langle x, w_0 \rangle + \frac{b}{\|w\|} | = \frac{|\langle x, w \rangle + b|}{\|w\|}$.

L'hyperplan séparateur le plus proche des exemples positifs, noté H_+ a pour équation :

$$H_+ : \langle w, x \rangle + b = 1.$$

L'hyperplan séparateur le plus proche des exemples négatifs, noté H_- a pour équation :

$$H_- : \langle w, x \rangle + b = -1.$$

La marge positive est donc la distance entre H et H_+ et la marge négative la distance entre H et H_- . L'hyperplan optimal est l'hyperplan qui maximise la marge. Il se trouve à mi-distance de H_+ et H_- avec a distance entre H_+ et H_- la plus grande possible.

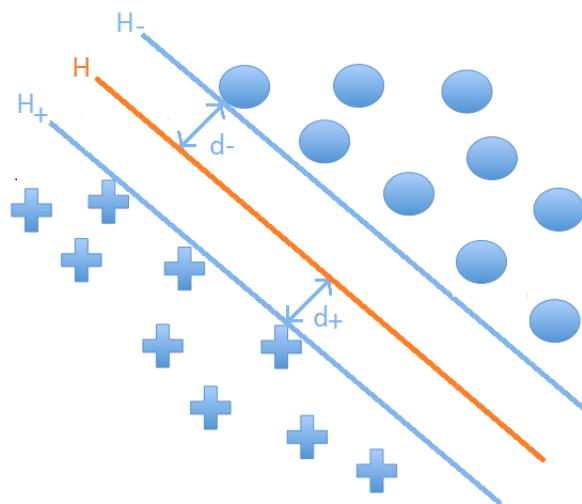


FIGURE 3.7 – Hyperplan optimal

La marge vaut : $d = d_+ + d_- = \frac{2}{\|w\|}$.

Le problème de maximisation de la marge revient donc à maximiser $\frac{2}{\|w\|}$ ce qui revient à minimiser $\frac{\|w\|^2}{2}$ sous les contraintes : $y^r(\langle w, x^r \rangle + b) \geq 1, \quad \forall 1 \leq r \leq p$.

Nous cherchons donc à résoudre le problème suivant :

$$\begin{cases} \min_{w,b} (f(w,b)) \text{ avec } f(w,b) = \frac{\|w\|^2}{2} \\ g_r(w,b) \leq 0, \quad \forall 1 \leq r \leq p \end{cases}$$

avec :

$$g_r(w,b) = -(y^r(\langle w, x^r \rangle + b) - 1).$$

Pour ce faire, nous utilisons la méthode classique des multiplicateurs de Lagrange. Cette méthode permet de transformer un problème d'optimisation sous contraintes en un problème d'optimisation sans contrainte. Le Lagrangien est donné par :

$$L(w,b,\lambda) = f(w,b) + \lambda \times g(w,b) = \frac{\|w\|^2}{2} - \sum_{r=1}^p \lambda_r (y^r(\langle w, x^r \rangle + b) - 1)$$

avec :

- λ le vecteur des multiplicateurs de Lagrange,
- $\lambda_i \geq 0, \quad \forall i$.

La solution du problème est déterminée par le point selle du lagrangien : (w^*, b^*, λ^*) . Le lagrangien $L(w,b,\lambda)$ doit être minimisé par rapport à (w,b) et maximisé par rapport à λ .

Le théorème de Kuhn-Tucker nous permet de déduire que si $x^* = (w^*, b^*)$ est solution du problème d'optimisation, il vérifie :

$$\begin{cases} 1) \sum_{r=1}^p \lambda_r g(x^*) = 0 \\ 2) \nabla f(x^*) + \sum_{r=1}^p \lambda_r \nabla g(x^*) = 0 \end{cases}$$

On déduit de 2) :

$$\begin{cases} \text{a) } \frac{\partial L}{\partial w}(w^*, b^*, \lambda) = w^* - \sum_{r=1}^p \lambda_r y^r x^r = 0 \\ \text{b) } \frac{\partial L}{\partial b}(w^*, b^*, \lambda) = - \sum_{r=1}^p \lambda_r y^r = 0 \end{cases}$$

Et, en remarquant que $\lambda_i \geq 0 \quad \forall i$ et $-(y^r(\langle x^r, w^* \rangle + b^*) - 1) \geq 0$, le point 1) nous donne :

$$\lambda_i (y^r(\langle x^r, w^* \rangle + b^*) - 1) = 0, \quad \forall i = 1, \dots, p.$$

On en déduit que : $\lambda_i = 0$, lorsque $y^r(\langle x^r, w^* \rangle + b^*) > 1$.

C'est à dire que seuls les points vérifiant l'égalité $y^r(\langle x^r, w^* \rangle + b^*) = 1$ interviennent dans le choix de l'hyperplan optimal. Ces points sont appelés vecteurs support. Ce résultat est particulièrement intéressant, puisqu'il implique qu'ajouter des observations à l'échantillon d'apprentissage qui ne sont pas des vecteurs supports n'a aucune influence sur la solution finale. Ce principe permet un calcul efficace et une complexité raisonnable malgré une augmentation de l'échantillon d'apprentissage, d'où

une popularité des SVM pour traiter des problèmes contenant un très grand nombre de données.

En substituant ces conditions dans le Lagrangien, on obtient :

$$\begin{aligned} L(w^*, b^*, \lambda) &= \frac{\|w^*\|^2}{2} - \sum_{r=1}^p \lambda_r y^r \langle w^*, x^r \rangle - \sum_{r=1}^p \lambda_r y^r b^* + \sum_{r=1}^p \lambda_r \\ &= \sum_{r=1}^p \lambda_r - \frac{\langle w^*, w^* \rangle}{2} \\ &= \sum_{r=1}^p \lambda_r - \frac{1}{2} \times \sum_{r=1}^p \sum_{s=1}^p \lambda_r \lambda_s y^r y^s \langle x^r, x^s \rangle. \end{aligned}$$

Nous devons alors résoudre le problème :

$$\begin{cases} \max_{\lambda} (L(w^*, b^*, \lambda)) \\ \sum_{r=1}^p \lambda_r y^r = 0 \\ \lambda_r \geq 0 \quad \forall r = 1, \dots, p \end{cases}$$

Finalement, l'hyperplan optimal a pour équation $H : \langle w^*, x \rangle + b^* = 0$

avec :

- $w^* = \sum_{r|\lambda_r^* \neq 0} \lambda_r^* y^r x^r$,
- $b^* = \frac{-1}{2} \times (\langle w^*, x^+ \rangle + \langle w^*, x^- \rangle)$,
- x^+ (resp x^-) le vecteur support associé à la classe d'observation positive (resp négative).

Pour une nouvelle observation x , le modèle prédit la classe négative -1 (resp. positive +1) si le signe de $\langle w^*, x \rangle + b^*$ est négatif (resp. positif).

• La notion de marge floue :

Lorsque les observations ne sont pas séparables par un hyperplan, il est nécessaire d'assouplir les contraintes par l'introduction d'un terme d'erreur ϵ , autorisant des observations mal classées dans la base d'apprentissage. Il est également possible d'introduire ce terme d'erreur alors que l'espace est linéairement séparable.

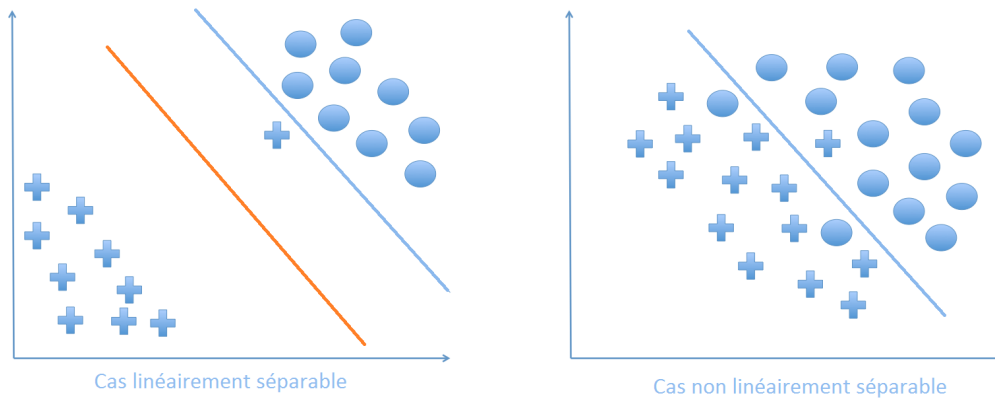


FIGURE 3.8 – Exemples de marge floue

À droite, l'introduction du terme d'erreur permet d'obtenir un hyperplan séparateur dans le cas d'un espace non linéairement séparable. À gauche, l'introduction du terme d'erreur permet de passer de l'hyperplan bleu à l'hyperplan orange. Malgré une erreur plus élevée de ce dernier, l'introduction

du terme d'erreur rend le modèle davantage généralisable.

Avec l'introduction du terme d'erreur $\epsilon > 0$, la contrainte $y^r(\langle w, x^r \rangle + b) \geq 1, \forall r = 1, \dots, p$ est remplacée par $y^r(\langle w, x^r \rangle + b) \geq 1 - \epsilon^r, \forall r = 1, \dots, p$ ⁵.

Le problème d'optimisation devient alors :

$$\begin{cases} \min_{w,b} (f(w, b, \epsilon)) \\ y^r(\langle w, x^r \rangle + b) \geq 1 - \epsilon^r \quad \forall r = 1, \dots, p, \quad \epsilon^r \geq 0, \quad \forall r = 1, \dots, p. \end{cases}$$

avec :

$$f(w, b, \epsilon) = \frac{1}{2} \times \|w\|^2 + C \times \sum_{r=1}^p \epsilon^r.$$

C est une constante positive ou nulle pénalisant les termes d'erreur. Il s'agit d'un paramètre essentiel pour l'optimisation des SVM, on y accordera une attention particulière lors du calibrage du SVM. Lorsque C est faible, les erreurs sont faiblement pénalisées, le modèle a donc un biais plus élevé et une variance plus faible. Lorsque C est élevé la pénalisation des erreurs va amener à un biais plus faible, mais à une variance potentiellement plus élevée.

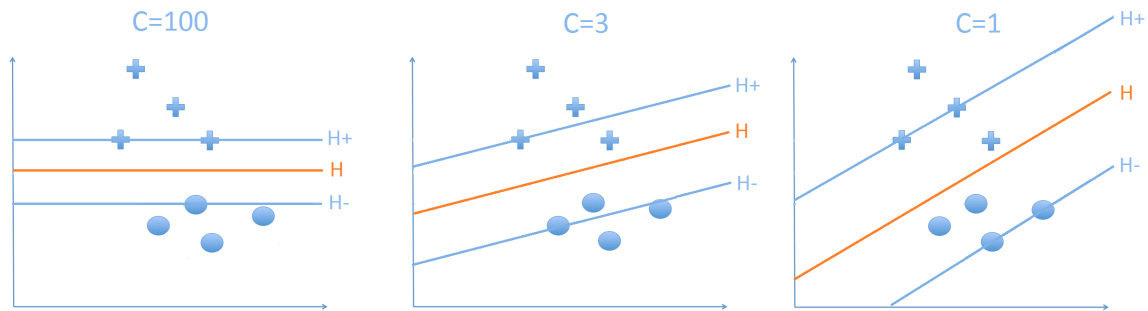


FIGURE 3.9 – Influence du paramètre C dans le choix de l'hyperplan séparateur

Nous illustrons ci-dessus l'impact du paramètre C dans le choix de l'hyperplan séparateur en orange. On remarque que les vecteurs supports définissant la marge sont plus ou moins éloignés de la surface de décision. Un C faible utilisera des vecteurs supports qui ne seront pas nécessairement les plus proches de la surface de décision.

• L'astuce du noyau :

Dans certain cas, on peut transformer un problème qui n'est pas linéairement séparable en se plaçant dans un espace de dimension supérieure. À titre d'illustration nous proposons l'exemple ci-dessous issu du livre « Data Science : Fondamentaux et études de cas » [2].

Nous avons une base d'apprentissage avec les couples entrée/sortie $(X, Y) = (X_i, Y_i)_{i \geq 0}$, où chaque $X_i = (x_{1,i}, x_{2,i})$ est un vecteur de dimension 2. Selon les valeurs de X , Y appartient à deux classes différentes représentées ci-après à gauche par les couleurs bleu et rouge.

L'espace n'est pas linéairement séparable comme nous pouvons le constater. Cependant, une transformation simple nous permet de résoudre un tel problème. On utilise la fonction :

$$\begin{aligned} \phi : \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (x_{1,i}, x_{2,i}) &\mapsto (x_{1,i}, x_{2,i}, x_{1,i}^2 + x_{2,i}^2). \end{aligned}$$

5. Lorsque ϵ^r est strictement supérieur à 1, le modèle attribue une mauvaise classe au vecteur x^r correspondant.

Nous représentons alors les données transformées ci-dessous à droite (avec les deux classes représentées par les couleurs vertes et rouge). Si l'espace n'est pas linéairement séparable en dimension 3, les performances obtenues par un SVM dans ce cas sont nettement améliorées. L'hyperplan d'équation $x_3 = 0$ dans la figure 3.10 à droite revient à tracer un cercle dans l'espace de départ ($x_1^2 + x_2^2 = 0$).

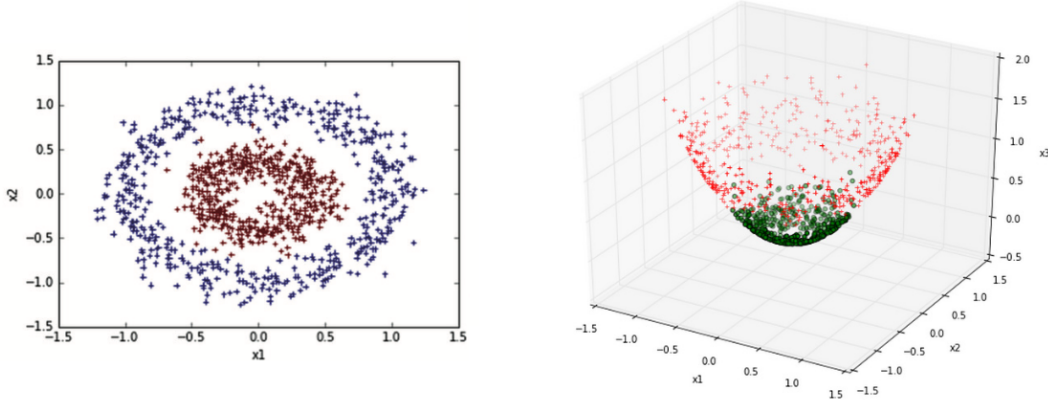


FIGURE 3.10 – Astuce du noyau

Généralisons ce principe. Nous définissons une fonction ϕ :

$$\begin{aligned}\phi &: E \rightarrow F \\ X &\mapsto \phi(X)\end{aligned}$$

avec :

- E l'espace initial,
- F un espace de dimension supérieure.

Nous cherchons alors à déterminer l'hyperplan séparateur H d'équation : $H : \langle w, \phi(x) \rangle + b = 0$.

Nous sommes ainsi ramené au problème suivant :

$$\begin{cases} \max_{\lambda} (L(w^*, b^*, \lambda)) \\ \sum_{r=1}^p \lambda_r y^r = 0 \\ \lambda_r \geq 0, \forall r = 1, \dots, p \end{cases}$$

Où le lagrangien est alors :

$$L(w^*, b^*, \lambda) = \sum_{r=1}^p \lambda_r - \frac{1}{2} \times \sum_{r=1}^p \sum_{s=1}^p \lambda_r \lambda_s y^r y^s \langle \phi(x^r), \phi(x^s) \rangle.$$

Afin d'éviter un calcul complexe du produit scalaire entre deux vecteurs de dimension potentiellement élevée, on utilise l'astuce du « *Kernel Trick* » ou en français « astuce du noyau ». Cette astuce évite également la difficulté liée à la transformation de chaque observation par la fonction ϕ . Nous remarquons en effet que L dépend du produit scalaire dans F qui peut être défini préalablement via une fonction K (appelée « fonction noyau ») :

$$\begin{aligned}K &: E \times E \rightarrow R \\ (x_r, x_s) &\mapsto \langle \phi(x_r), \phi(x_s) \rangle.\end{aligned}$$

Nous pouvons obtenir la fonction K en définissant ϕ , mais il est également possible de définir K et ne pas expliciter ϕ . Dans ce cas la fonction noyau doit répondre à certaines conditions appelées

conditions de Mercer [24].

En pratique, nous choisissons la fonction noyau K sans expliciter ϕ . Il existe un large choix de fonctions noyau, nous allons en présenter quelques unes telles qu'implémentées dans la librairie `sklearn` de Python :

- la fonction noyau linéaire : $K(x_r, x_s) = \langle x_r, x_s \rangle$,
- la fonction noyau polynomiale : $K(x_r, x_s) = (\gamma \langle x_r, x_s \rangle + c_0)^d$,
- le noyau gaussien ou rbf (*Radial Basis Function*) : $K(x_r, x_s) = e^{-\gamma \|x_r - x_s\|^2}$,
- le noyau sigmoïde : $K(x_r, x_s) = \tanh(\gamma \langle x_r, x_s \rangle + c_0)$.

Nous remarquons que l'utilisation de certains noyaux amènent à choisir des paramètres γ et c_0 , un tel choix sera alors effectué par validation croisée dans un second temps (après avoir choisi la fonction noyau qui semble la plus adaptée à notre problème).

• Les SVR :

En 1996, les SVM ont été généralisés pour les problèmes de régression, nous parlons alors de SVR [13].

Le fonctionnement du SVR utilise de nombreuses notions que nous avons présenté dans le cadre du SVM. Nous avons préféré présenter ces notions dans un cadre plus simple pour commencer. Lorsque l'on est dans un problème de régression, la variable de sortie Y est continue.

Nous cherchons alors à déterminer une fonction \hat{f} telle que $\hat{f}(x) \approx y$. De manière similaire au SVM, \hat{f} est un hyperplan de la forme $\hat{f}(x) = \langle w, x \rangle + b$. Il s'agit cependant ici de minimiser l'écart entre $\hat{f}(x)$ et y . Le SVR propose alors de déterminer l'hyperplan qui se situe dans un hypertube de largeur $2 \times \epsilon$ contenant tous les exemples d'apprentissages. Il peut exister plusieurs hypertubes vérifiant cette condition. L'hypertube optimal est celui qui minimise la distance entre y et l'hyperplan \hat{f} . Cela revient alors à maximiser la distance entre y et l'hypertube. Ce problème se résout de manière similaire à celui présenté dans le cas de la classification.

Voici une illustration en dimension 2 d'un SVR. L'hypertube est représenté par les droites bleues et l'hyperplan optimal est alors la droite orange :

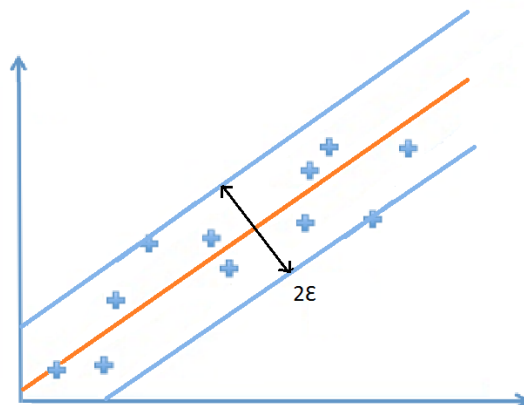


FIGURE 3.11 – Le SVR en dimension 2

Pour résoudre le problème, nous passons au lagrangien :

$$L(w^*, b^*, \lambda_1, \lambda_2) = -\frac{1}{2} \times \sum_{i,j=1}^p (\lambda_{1,i} - \lambda_{2,i})(\lambda_{1,j} - \lambda_{2,j}) \langle x_i, x_j \rangle - \epsilon \sum_{i=1}^p \lambda_{1,i} + \lambda_{2,i} + \sum_{i=1}^p y_i (\lambda_{1,i} - \lambda_{2,i}).$$

Nous obtenons le problème d'optimisation suivant :

$$\begin{cases} \max_{\lambda_1, \lambda_2} (L(w^*, b^*, \lambda_1, \lambda_2)) \\ \sum_{r=1}^p \lambda_{1,r} - \lambda_{2,r} = 0 \\ 0 \leq \lambda_{i,r} \leq C, \forall r = 1, \dots, p, i = 1, 2 \end{cases}$$

L'hyperplan optimal est alors : $\hat{f}(x) = \langle w^*, x \rangle + b^*$

avec :

$$- w^* = \sum_{i=1}^p (\lambda_{1,i} - \lambda_{2,i}) x_i,$$

- b^* défini comme dans le cas de la classification à partir des vecteurs supports de l'hypertube.

De manière similaire au cas de la classification, il est souvent préférable d'utiliser des marges floues, nous introduisons alors deux termes d'erreur supplémentaire : η et η^* qui autorisent des observations à être en dehors de l'hypertube.

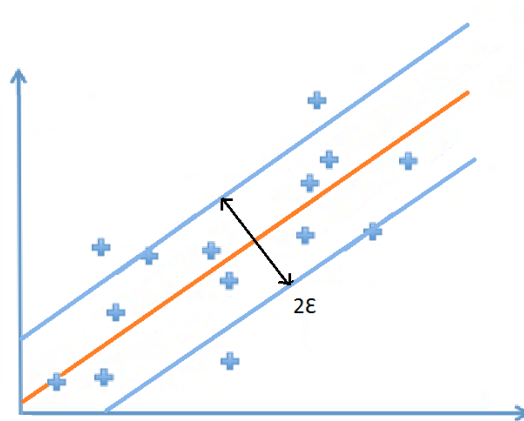


FIGURE 3.12 – Le SVR en dimension 2

Enfin, nous pouvons également utiliser l'astuce du noyau de manière similaire au cas présenté dans le cadre de la classification :

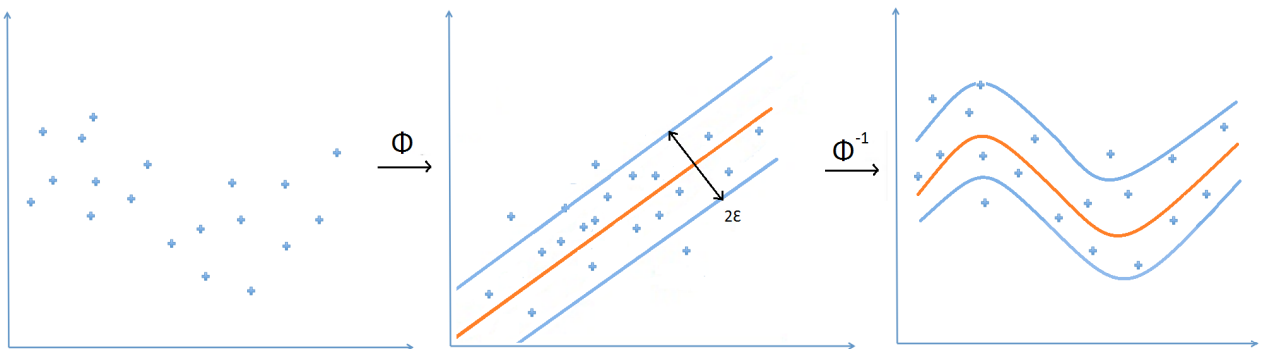


FIGURE 3.13 – Le *kernel trick* pour un problème de régression

• Le SVR en pratique :

Après avoir présenté les SVR nous allons rappeler les principaux paramètres de ce modèle, que l'on cherchera à optimiser lors de la phase de calibrage :

- *kernel* : la fonction K utilisée dans le « *kernel trick* ». Nous choisirons parmi les fonctions « linéaire », « poly », « rbf » ou « sigmoid » que nous avons présenté précédemment. Nous débuterons la phase de calibrage par le choix de ce paramètre. Dans le cas où la fonction polynomiale serait retenue, nous devons déterminer un paramètre « *degree* » correspondant au degré du polynôme à considérer. Nous devons également déterminer un paramètre c_0 si la fonction noyau retenue est polynomiale ou sigmoïde.
- γ : un paramètre des fonctions noyau polynomiales, gaussiennes ou sigmoïdes que nous avons présenté.
- C : le terme de pénalisation des erreurs lors de l'utilisation de marge floue. C élevé pénalise fortement les erreurs et réduit le biais au détriment d'une augmentation possible de la variance.
- ϵ : un paramètre qui va jouer sur la taille de l'hypertube.

Après avoir présenté les différentes méthodes de *machine learning* que nous allons utiliser et justifié le choix de ces méthodes, nous allons maintenant étudier la mise en place de ces modèles pour se substituer aux SdS. Nous allons tout d'abord présenter un historique des travaux réalisés sur ce sujet en mettant en avant les pistes d'améliorations qui font l'objet du mémoire. Nous présenterons ensuite la méthode dans un cadre général. Nous expliquerons enfin, la mise en place pratique de la méthode, dans le cadre d'un calcul de SCR vision modèle interne ou d'une projection du ratio de solvabilité à horizon un an.

3.3 Le machine learning comme alternative aux SdS

« C'est avec la logique que nous prouvons et avec l'intuition que nous trouvons. »

Henri Poincaré

3.3.1 Présentation de la méthode alternative aux SdS

• Introduction et présentation des travaux existants :

De nombreuses méthodes alternatives aux SdS existent. Nous en avons présenté brièvement quelques-unes. Dans ce mémoire nous proposons une méthode basée sur du *machine learning* qui vise à dépasser certains défauts précédemment cités.

En particulier, nous souhaitons développer une méthode qui :

- Puisse être adaptée aux différentes problématiques actuarielles. Nous souhaitons qu'elle puisse se substituer à la méthode des SdS pour le calcul du SCR en modèle interne, mais également à d'autres problèmes actuariels. En particulier, nous souhaitons pouvoir l'adapter à l'estimation de la distribution du ratio de solvabilité futur ou des Fonds Propres futurs, dans le cadre d'une problématique de choix d'allocation d'actifs telle que présentée au chapitre 1.
- Ne présente pas un coût d'entrée trop élevé. Nous avons vu que la mise en oeuvre des algorithmes d'apprentissage supervisés nécessite l'utilisation d'une base de données contenant des couples entrées/sorties avec en sortie la variable que nous souhaitons expliquer. La création de cette base de données peut être obtenue par la méthode des SdS. Dans ce cas, il faudra s'assurer que ce coût d'entrée ne sera pas trop élevé par rapport à d'autres méthodes alternatives.

Différentes utilisations des méthodes de *machine learning* se substituant à la méthode des SdS ont déjà été proposées dans différents mémoires. Nous allons présenter différents articles et travaux réalisés

nous ayant inspiré pour la mise en place de notre méthode.

En 2013, François Bonnin, Frédéric Planchet et Marc Juillard publient un article intitulé « *Best Estimate calculations of savings contracts by closed formulas Application to the ORSA* » [4]. En proposant la mise en place d'une formule fermée pour le calcul de la provision *Best Estimate*, ils confirment l'intuition selon laquelle il existe un lien entre l'environnement économique et la provision *Best Estimate*. Or nous avons vu que l'apprentissage supervisé repose sur l'existence d'une fonction lien entre les variables explicatives et la variable à expliquer. La possibilité d'apprendre la provision *Best Estimate* en fonction de l'environnement économique apparaît alors pertinente et promet des résultats robustes. Pourtant, cette idée ne sera pas immédiatement exploitée.

La même année, deux mémoires d'actuariat ont attiré notre attention. Le premier a été encadré par Laurent Devineau, qui a beaucoup travaillé sur les méthodes alternatives aux SdS. Le mémoire s'intitule « Optimisation du calcul du capital économique d'une compagnie d'assurance vie par la méthode des Simulations dans les Simulations » [23]. Ce mémoire propose notamment de mettre en place un algorithme de forêts aléatoires. L'algorithme permet d'estimer les Fonds Propres à horizon un an en fonction de l'univers économique généré. Cette estimation est peu coûteuse, cependant l'estimation des Fonds Propres dans les cas les plus extrêmes pose problème. Or, il s'agit justement de ces scénarios extrêmes qui nous intéressent pour un calcul de SCR. Les auteurs remarquent cependant que les Fonds Propres estimés par les forêts aléatoires conservent leur rang. C'est-à-dire que les scénarios monde réel qui amènent aux pires Fonds Propres par les forêts aléatoires donnent également les pires Fonds Propres si on réalise une estimation par la méthode classique. Il est alors possible d'estimer les Fonds Propres par les forêts aléatoires et de trier les Fonds Propres estimés du plus faible au plus élevé. Une fois le classement effectué, il suffit de réaliser des simulations de second niveau sur les scénarios qui amènent les Fonds Propres les plus défavorables lors de l'estimation par les forêts aléatoires uniquement. Le quantile est alors estimé sur ces scénarios ce qui permet de réduire grandement le nombre de simulations.

Malgré ces résultats, nous envisageons d'étudier un algorithme de forêts aléatoires. Il est en effet possible qu'un apprentissage des Fonds Propres soit plus complexe qu'un apprentissage de la provision *Best Estimate* que nous pensons plus pertinent. De plus, il peut être intéressant de comparer deux méthodes ensemblistes (Random Forest et XGBoost). Enfin, il est possible que malgré ces résultats décevants sur les quantiles, le Random Forest soit performant dans d'autres situations où d'autres algorithmes seraient moins robustes. Nous pourrions alors envisager une agrégation de plusieurs algorithmes afin de profiter des performances de chacun dans les situations où il est le plus adapté.

Le second mémoire s'intitule « Utilisation des supports vecteurs machines pour l'accélération du calcul du capital économique », rédigé par Marine Niedzwiedz [26]. Ce mémoire propose d'utiliser le modèle d'apprentissage supervisé SVM afin de classer les scénarios primaires générés par le GSE en deux catégories : risqué ou non risqué selon deux facteurs de risques : le risque action et le risque de taux. Cette méthode permet ainsi de réduire le nombre de scénarios primaires lors du calcul d'un SCR en modèle interne. Si l'on est capable d'estimer les scénarios primaires les plus défavorables, il n'est alors plus nécessaire de dresser un bilan prudentiel dans les autres scénarios, puisque nous cherchons à déterminer un quantile. Cette méthode obtient ainsi d'excellents résultats pour le calcul du SCR. Enfin, un apprentissage direct des Fonds Propres via le SVRM est proposé. Cet apprentissage est concentré sur la zone quantile et montre d'excellents résultats en terme de prédiction du SCR. Nous souhaitons alors savoir si ces résultats peuvent être généralisables à d'autres allocations d'actifs et si l'apprentissage doit nécessairement se concentrer sur la zone des scénarios les plus défavorables.

Ces deux mémoires présentent des résultats particulièrement intéressants, mais l'application à la projection du ratio de solvabilité n'est que peu ou pas développée. En 2015, Amélie Roué propose un mémoire intitulé « Les réseaux bayésiens comme technique d'optimisation numérique dans le cadre de l'ORSA » [28]. Elle propose ainsi d'obtenir une distribution du ratio de solvabilité à horizon 1, 2, 3, 4

et 5 ans. Elle étudie notamment un algorithme d'apprentissage supervisé : les réseaux de neurones. Il s'agit d'apprendre le ratio de solvabilité à différents horizons de temps en fonction de différents facteurs de risques. Cette étude est complétée en 2017 par le mémoire « Approximation du bilan économique sous Solvabilité II via des méthodes d'apprentissage automatique et application à l'ORSA ».

L'auteur montre alors une supériorité de la méthode ensembliste XGBoost. Un apprentissage est proposé pour le SCR et pour la provision *Best Estimate*. Comme nous le présentons, l'apprentissage du *Best Estimate* donne de bons résultats. En revanche la prédiction du SCR vision formule standard semble moins robuste, c'est pourquoi nous allons nous concentrer sur le modèle interne.

En apprenant la provision *Best Estimate* nous pourrions en déduire le SCR de la manière suivante :

- calcul de la valeur de marché de l'actif,
- estimation de la provision *Best Estimate* par un modèle de *machine learning*,
- calcul des Fonds Propres par différence entre la valeur de marché de l'actif et la provision *Best Estimate*
- déduction du quantile à 0,5% des Fonds Propres à horizon un an et du SCR.

Notons toutefois que l'apprentissage de la provision *Best Estimate* rend l'analyse des résultats plus complexes. Les scénarios monde réel qui amènent aux *Best Estimate* les plus « extrêmes » ne sont pas nécessairement les mêmes que les scénarios qui amènent aux Fonds Propres les plus « extrêmes ». De plus, un algorithme qui estime convenablement la provision *Best Estimate* en moyenne ne sera pas nécessairement pertinent pour le calcul d'un SCR en vision modèle interne.

Nous reviendrons sur ces points lors de l'application numérique. Nous allons maintenant entrer davantage dans le détail quant à la méthode que nous proposons d'utiliser.

• Présentation de la méthode avec du *machine learning* :

Nous allons utiliser des algorithmes supervisés pour prédire la provision *Best Estimate*.

Comme nous l'avons vu, les algorithmes supervisés permettent de prédire une variable de sortie Y à partir de variables explicatives $X = (X_i)_{0 \leq i \leq n}$ avec X une matrice $m \times n$ et Y un vecteur de taille m . Nous souhaitons pouvoir utiliser notre méthode pour estimer un SCR en vision modèle interne, mais également pouvoir l'utiliser dans le cadre du choix d'une allocation d'actifs. Une utilisation de la méthode de substitution dans ce cadre s'avère particulièrement intéressante. En effet, une des limites importantes de l'utilisation de méthodes alternatives est la justification des modèles retenus auprès de la hiérarchie et des institutions de contrôle. Or, si on utilise un modèle alternatif pour tester différentes allocations d'actifs, il est possible de justifier le choix final de l'allocation en utilisant la méthode des SdS pour l'allocation d'actifs retenue. Ainsi, nous pourrions étudier facilement l'impact d'un grand nombre d'allocations d'actifs différentes sur le bilan futur, comparer chaque allocation d'actifs et choisir celle qui semble la plus pertinente. Puis, pour cette allocation, on effectuera les calculs complets via la méthode des SdS. Si le modèle est suffisamment précis les résultats obtenus ne seront pas sensiblement différents, mais ne poseront plus de problème de justification.

Notons qu'une telle étude avec la méthode des SdS nous est impossible puisque utiliser une fois la méthode des SdS pour une allocation d'actifs est très coûteux. Il n'est donc pas envisageable de tester l'impact de différentes allocations d'actifs sur les bilans futurs sans passer par une méthode alternative.

Enfin, nous souhaitons pouvoir appliquer notre méthode à la projection du ratio de solvabilité à horizon un an. Nous avons vu qu'il s'avère délicat de conserver de bonnes performances en terme de calcul du SCR lorsque nous nous plaçons dans le cadre de la formule standard. Nous resterons donc en vision modèle interne pour le calcul du SCR. Ce faisant, nous transformerons un problème de

« Simulations dans les Simulations dans les Simulations » (SdSdS) de complexité N^3 en un problème de « Simulations dans les Simulations » de complexité N^2 .

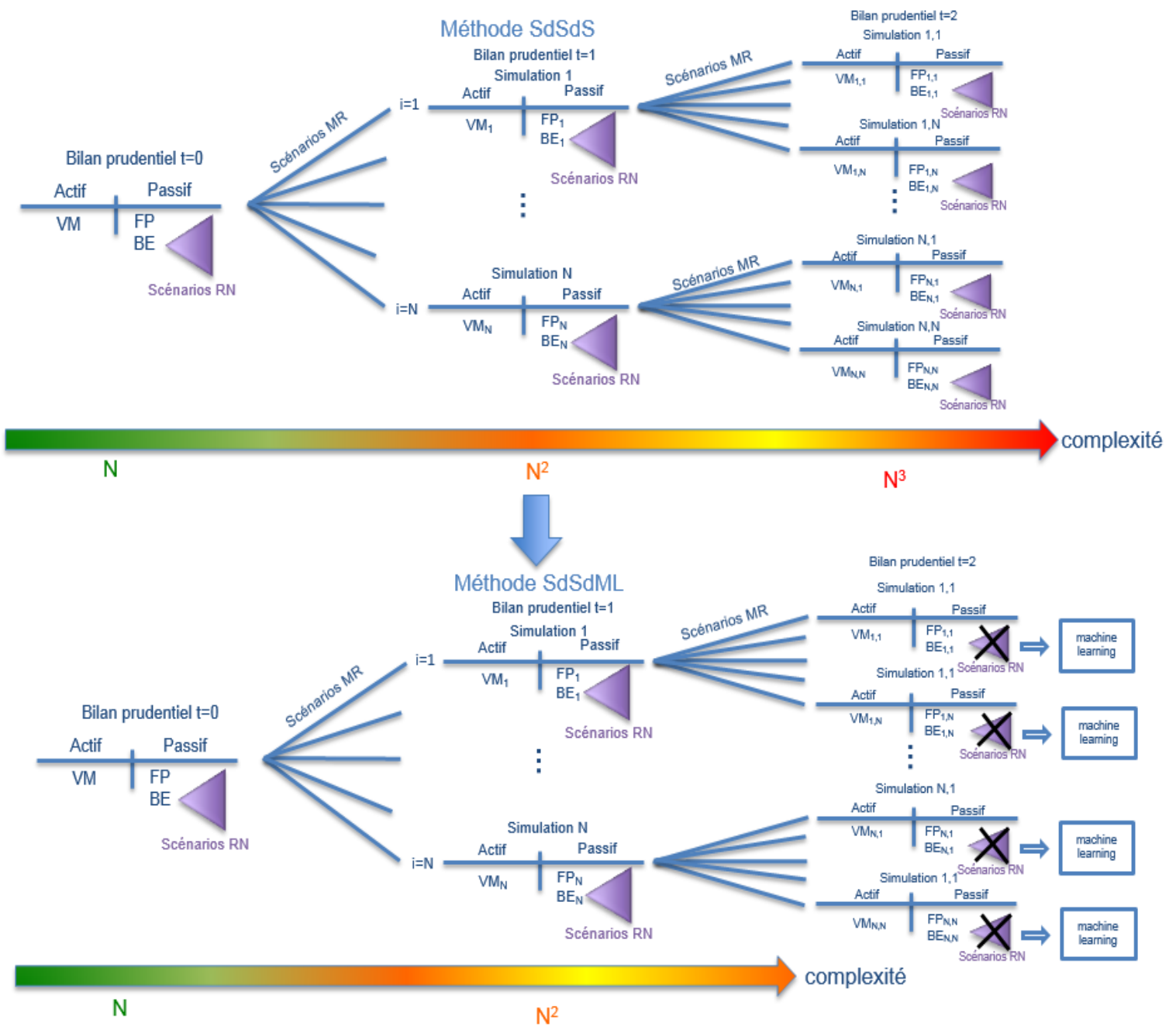


FIGURE 3.14 – La méthode SdSdML, alternative aux SdSdS

Face à une telle complexité, une telle méthode n’a, à notre connaissance jamais été proposée dans un mémoire d’actuariat. Nous parlerons ultérieurement des choix qui permettent de faire face à une telle complexité. Notons également qu’une augmentation des capacités machine ouvre les portes de telles applications auparavant in-envisageables.

Nous proposons donc une méthode permettant pour différentes allocations d’actifs de :

- calculer un SCR vision modèle interne en estimant le BE à horizon un an sans nécessiter de re-calibrage du GSE ni simulations risque neutre,

- obtenir une estimation de la distribution du ratio de solvabilité à horizon un an, via une estimation du BE à horizon deux ans.

Nous reviendrons sur la complexité des deux approches et le gain de temps apporté par la méthode. L'apprentissage du BE va nécessiter deux bases d'apprentissage et de test différentes selon l'horizon de temps (1 an ou 2 an). Nous nous concentrerons tout d'abord sur la robustesse de la méthode dans le cadre de l'estimation d'un SCR via une projection des Fonds Propres à horizon un an. Après avoir démontré la robustesse de la méthode dans le cadre de l'estimation du SCR, nous l'appliquerons à la projection du ratio de solvabilité.

Pour notre étude à horizon un an (resp deux ans), nous considérons les couples (X,Y) de variables entrée/sortie suivants :

- X l'ensemble des variables économiques projetées par le GSE à horizon un an (resp deux ans), ainsi que l'allocation d'actifs à horizon un an (resp deux ans) et la provision *Best Estimate* initiale (resp à horizon un an),
- Y la provision *Best Estimate* dans un an (resp dans deux ans).

Nous allons maintenant détailler l'obtention de la base contenant les couples entrée/sortie (X,Y) .

3.3.2 La construction des bases d'apprentissage et de test

Nous allons utiliser la méthode des SdS pour obtenir une base de données que l'on divisera entre base d'apprentissage et base de test. Il faut cependant veiller à deux points essentiels :

- la base de données doit contenir des allocations d'actifs différentes de façon à ce que le modèle « capte » les liens entre la provision *Best Estimate* et l'allocation d'actif,
- le coût de l'obtention de la base de données ne doit pas être trop élevé, sans quoi l'intérêt de la méthode serait faible.

• Obtention de la base de donnée à horizon un an :

Afin d'obtenir une base d'apprentissage suffisamment diversifiée en terme d'allocations d'actifs, nous procédons de la manière suivante.

Nous considérons 51 allocations d'actifs initiales : de (0% Actions / 100% Obligations) à (50% Actions / 50% Obligations) avec un pas de 1%. Pour chaque allocation d'actif initiale, nous :

- dressons le bilan initial, le calcul de la provision *Best Estimate* est effectué avec 1 000 simulations risque neutre,
- simulons 300 scénarios monde réel entre 0 et 1,
- dressons le bilan prudentiel $t=1$, pour chaque scénario monde réel entre 0 et 1. Le calcul de la provision *Best Estimate* est effectué avec 1 000 simulations risque neutre.

N.B : nous ne re-balançons pas l'allocation d'actifs à horizon un an, afin de garantir davantage de diversité dans notre base de données.

L'obtention de cette base de données nécessite donc la génération de $51 \times 300 \times 1000$ scénarios (plus les 51×1000 scénarios pour le calcul de chaque provision *Best Estimate* initiale). Le calcul est effectué en quelques minutes. Nous obtenons ainsi une base de données contenant 15 300 lignes. Nous décidons de diviser cette base en 30% de données pour le test et 70% pour l'apprentissage qui est une répartition souvent utilisée pour des problématiques d'apprentissage supervisé. Pour chaque allocation d'actifs initiale, 200 des 300 scénarios monde réel sont conservés dans la base d'apprentissage. Les 100

scénarios restants sont conservés dans la base de test. Nous obtenons alors une base d'apprentissage $Base_{app}^{t=1}$ constituée de 10 200 lignes et une base de test $Base_{test}^{t=1}$ constituée de 5 100 lignes.

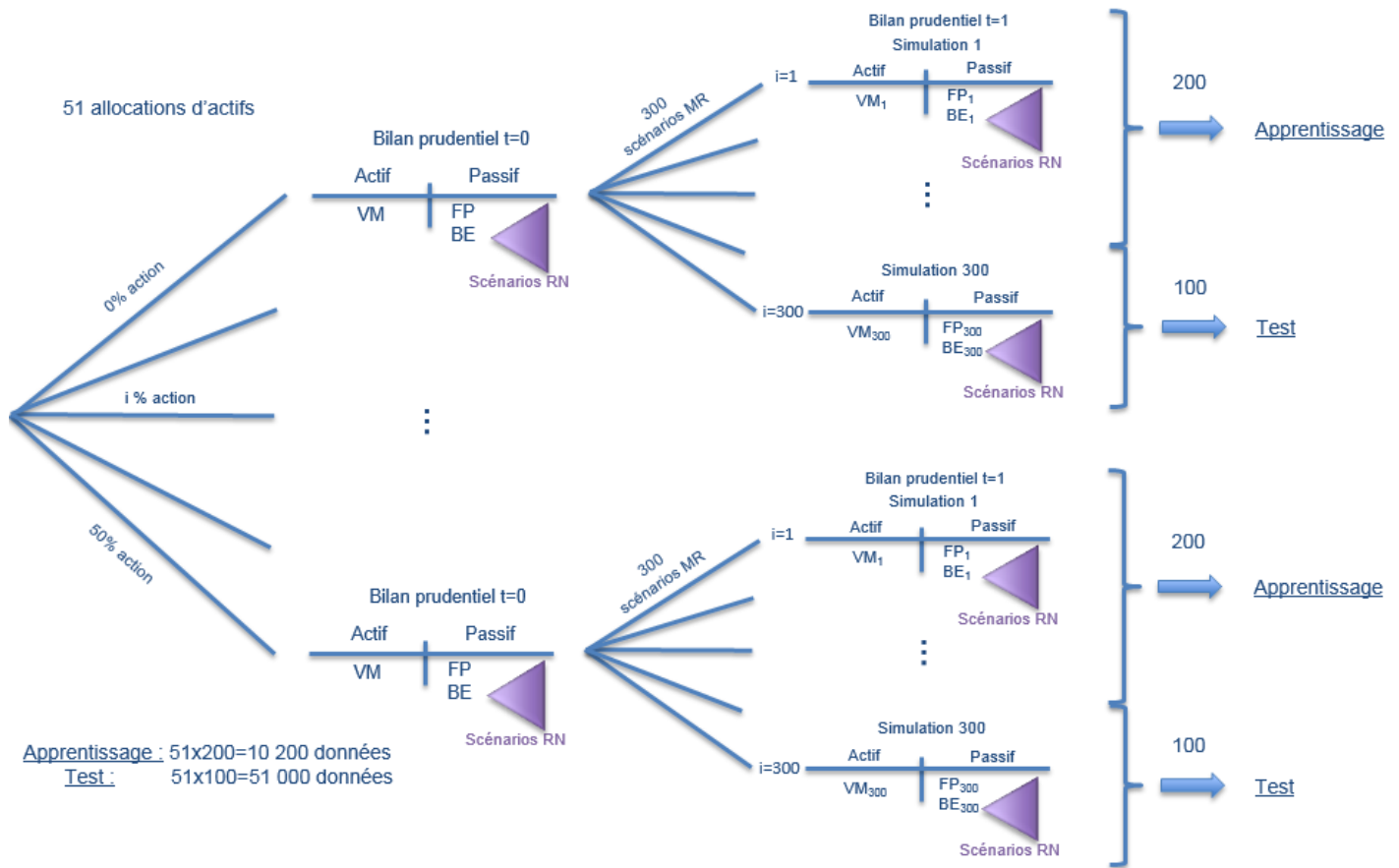


FIGURE 3.15 – Construction des bases d'apprentissage et de test à horizon un an

• Obtention de la base de données à horizon deux ans :

La construction de la base d'apprentissage et de test pour l'estimation de la provision *Best Estimate* à horizon deux ans repose sur les mêmes principes.

Nous considérons 51 allocations d'actifs initiales : de (0% Actions / 100% Obligations) à (50% Actions / 50% Obligations) avec un pas de 1%.

Pour chaque allocation d'actif initiale, nous :

- dressons le bilan initial, le calcul de la provision *Best Estimate* est fait avec 1 000 simulations risque neutre,
- simulons 25 scénarios monde réel entre 0 et 1,
- dressons le bilan prudentiel, pour chaque scénario monde réel entre 0 et 1. Le calcul de la provision *Best Estimate* est effectué avec 1 000 simulations risque neutre.
- simulons 25 scénarios monde réel entre 1 et 2,
- dressons le bilan prudentiel, pour chaque scénario monde réel entre 0 et 1. Le calcul de la provision *Best Estimate* est effectué avec 1 000 simulations risque neutre.

L'obtention de cette base de données nécessite donc la génération de $51 \times 25 \times 25 \times 1000 = 31\ 875\ 000$ scénarios (plus les 51×1000 scénarios pour le calcul de chaque provision *Best Estimate* initiale

et les $51 \times 25 \times 1000$ scénarios pour le calcul de chaque provision *Best Estimate*). L'obtention de cette base d'apprentissage est également rapide : il nécessite 1 h 20 de calcul. Nous reviendrons sur la problématique du temps de calcul ultérieurement. Nous obtenons alors également deux bases : $Base_{app}^{t=2}$ constituée de 21 675 lignes et $Base_{test-1}^{t=2}$ constituée de 10 200 lignes.

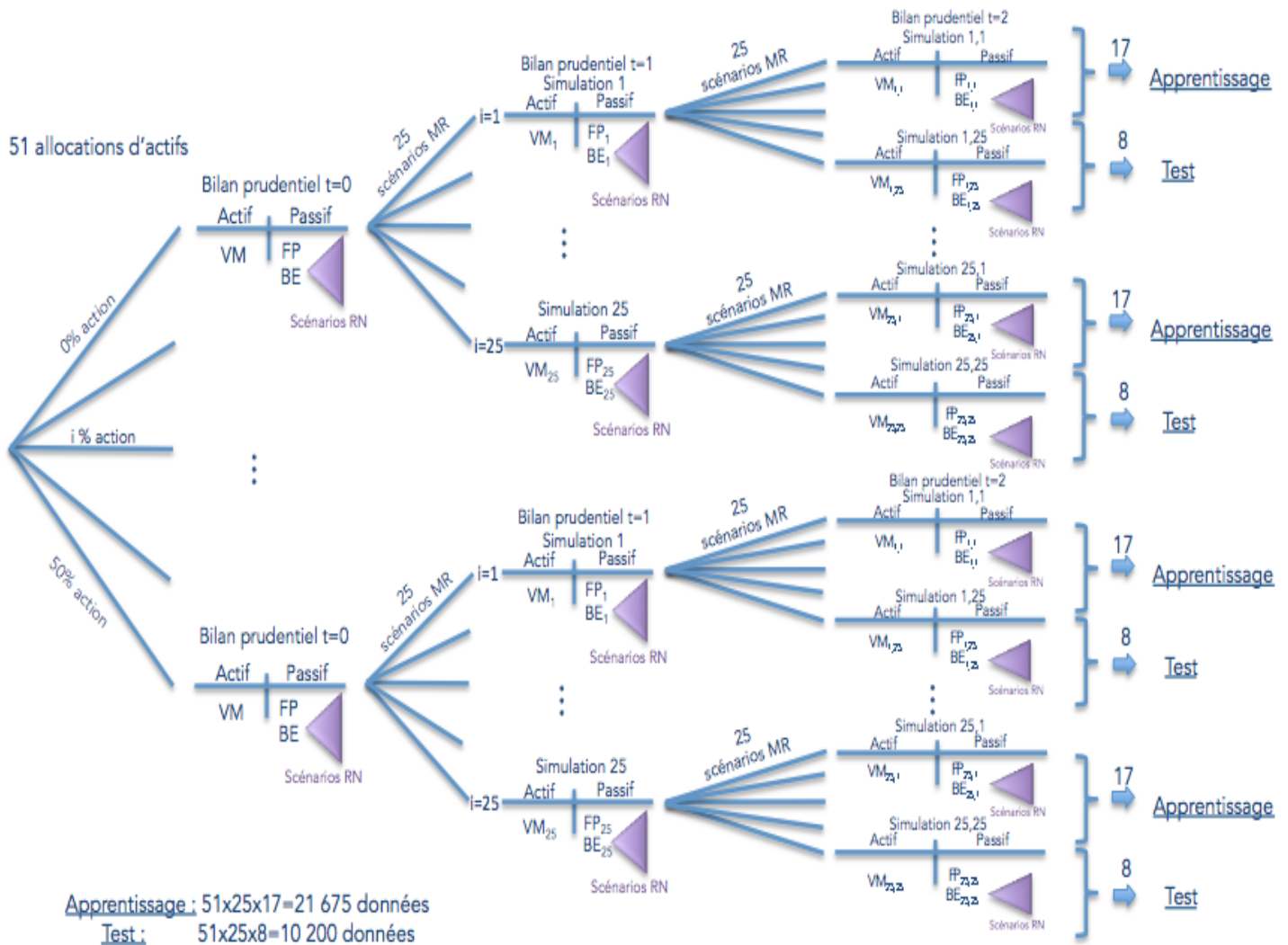


FIGURE 3.16 – Construction des bases d'apprentissage et de test à horizon 2 ans

Cependant, les bases de test à horizon un an et deux ans ne sont pas suffisantes pour déterminer la robustesse de nos modèles. C'est ce que nous allons mettre en lumière dans le paragraphe suivant.

• Utilisation du modèle en pratique :

Les bases de test obtenues vont nous donner des indications quant aux performances des différents modèles. En revanche, $Base_{test-1}^{t=1}$ ne nous permettra pas d'estimer un SCR, puisque nous avons une base contenant des allocations d'actifs différentes. Nous ne pourrons alors savoir si la méthode peut effectivement se substituer à l'utilisation des SdS.

Pour répondre à cette problématique, nous allons chercher à comparer le SCR obtenu par notre méthode sur le bilan initial de l'assureur à celui obtenu par une utilisation complète des SdS. Pour

ce faire, nous générerons 10 000 scénarios monde réel⁶ entre 0 et 1. À la suite de ces scénarios nous calculerons la valeur de marché de l'actif, les provisions *Best Estimate* en générant de 1 000 scénarios risque neutre et nous en déduirons les Fonds Propres.

Nous en déduirons alors le SCR en date 0 obtenu par la méthode des SdS. Cette base de 10 000 scénarios économiques associés à 10 000 bilans prudentiels à horizon un an nous servira de seconde base de test : $Base_{test-2}^{t=1}$.

Nous utiliserons alors nos différents modèles à l'issue des 10 000 scénarios monde réel afin d'estimer la provision *Best Estimate* $t=1$. Nous en déduirons les Fonds Propres, puis le SCR. En utilisant 10 000 scénarios primaires, nous pourrions comparer précisément les résultats obtenus par les différents modèles lors des scénarios les plus défavorables. Afin de confirmer le caractère généralisable de nos modèles, nous estimerons également les SCR obtenus sur des allocations d'actifs de 20% action et 40% action par les modèles que nous comparerons à ceux obtenus par la méthode des SdS avec 10 000 scénarios primaires.

La même problématique se pose pour juger de la performance de nos modèles sur l'estimation de la distribution du ratio de solvabilité à horizon un an. Pour y répondre, nous devons comparer la performance de nos modèles à la distribution théorique obtenue par la méthode des SdSdS. Cependant, nous ne pourrions conserver en mémoire la base de données utilisées ici, et le choix du nombre de scénarios devra être plus faible. Nous utiliserons la méthode des SdSdS avec 1 000 scénarios monde réel entre 0 et 1 et entre 1 et 2, et 1 000 scénarios risque neutre pour l'évaluation des différentes provisions *Best Estimate*.

Nous calculerons alors la MSE des 1 000 000 de provisions *Best Estimate* obtenues à horizon deux ans. Nous nous intéresserons également aux distributions des SCR et ratio de solvabilité à horizon un an obtenus par les modèles de *machine learning* que nous comparerons à ceux obtenus par la méthode des SdSdS.

• Un mot sur la complexité :

Notons RN pour risque neutre et MR pour monde réel.

De prime abord, le lecteur pourrait penser qu'il est équivalent de lancer 10×100 scénarios MR \times 1 000 scénarios RN (utiliser 10 fois la méthode des SdS) ou de lancer 1 000 scénarios MR \times 1 000 scénarios RN. Ce n'est pas le cas. Ceci est lié notamment à la taille des objets générés et à la gestion de la mémoire.

Sans entrer dans des détails techniques, il est plus simple et beaucoup plus rapide d'obtenir 10×100 scénarios MR \times 1 000 scénarios RN plutôt que 1 000 scénarios MR \times 1 000 scénarios RN. À titre d'exemple cette problématique est proche de celle que l'on rencontre lorsque l'on cherche à ouvrir un fichier de taille très importante sur Excel. L'utilisation de l'outil ne pose guère de problème pour générer 1 000 scénarios monde réel suivi de 1 000 scénarios risque neutre. En revanche, passer de 1 000 à 10 000 scénarios monde réel pose beaucoup plus de difficultés. Il est d'ailleurs arrivé plusieurs fois que l'ordinateur ne soit pas à même de terminer les calculs dans ce cas. Notons enfin que, lorsque nous utilisons un plus grand nombre de scénarios monde réel entre 0 et 1, l'impact est encore décuplé par l'enregistrement de données très volumineuses au format csv.

Ainsi la création des bases de données d'apprentissage et de test $Base_{test-1}^{t=1}$ est d'une complexité largement inférieure à celle d'une utilisation complète de la méthode des SdS telle que nous le faisons pour obtenir $Base_{test-2}^{t=1}$, bien que le nombre de scénarios utilisés pour l'obtention de la première base soit supérieur.

6. Nous reviendrons sur le choix du nombre de scénarios durant ce chapitre.

$Base_{test-2}^{t=1}$ est ainsi beaucoup plus complexe à obtenir et nécessite un temps de calcul particulièrement long⁷, là où l'obtention des bases d'apprentissage et de test $Base_{test-1}^{t=1}$ est très court (moins d'une heure).

Notons également qu'en théorie, si les modèles se généralisent bien à chacune des 51 allocations d'actifs testées, nous pourrions calculer 51 SCR (les SCR de chaque allocation entre 0 et 50% d'actions) et donc 51 ratio de solvabilité initial sans simulation risque neutre. Cela reviendrait à générer $51 \times 10\,000$ scénarios monde réel, puis utiliser le modèle. Alors que calculer 51 SCR avec la méthode des SdS nécessiterait la simulation de 1 000 fois plus de scénarios.

L'utilisation de la méthode pour la projection du ratio de solvabilité à horizon deux ans est en revanche plus complexe. Contrairement au cas $t=1$, nous ne conservons pas en mémoire les données générées⁸. L'utilisation de la méthode des SdSdS avec 1 000 scénarios MR et 1 000 scénarios RN en vue de comparer notre méthode aux résultats théoriques nécessite ainsi environ 3 jours de calculs.

3.3.3 Inférence statistique et validation des modèles

• Inférence statistique et choix du nombre de scénarios :

La plupart des modèles utilisés en assurance visent à prédire la réalité. Il est possible de confronter ces modèles à la réalité a posteriori ou encore de les calibrer sur un historique incomplet, puis déterminer l'erreur commise sur le reste de l'historique.

Dans notre étude, nous sommes dans une démarche différente. Nous confrontons notre modèle non pas à « la réalité » mais aux résultats d'un autre modèle (celui de la méthode des SdS). Il s'agit alors d'un problème d'inférence statistique.

Nous considérons alors que « la réalité » face à laquelle nous confronterons nos modèles sera les résultats obtenus par la méthode des SdS. Dès lors, il est important de choisir un nombre de scénarios permettant de justifier une bonne convergence. Il faut également tenir compte de la complexité induite par un nombre de scénarios « trop » élevé.

Dans notre étude, nous considérons deux problématiques essentielles pour le choix du nombre de scénarios :

- le nombre de scénarios risque neutre liés à l'estimation de la provision *Best Estimate*, qui s'apparente à un calcul de moyenne,
- le nombre de scénarios pour le calcul du quantile à 0,5% des Fonds Propres à horizon un an en vue de calculer un SCR en modèle interne.

Le nombre de scénarios à considérer dépend notamment du caractère rare de l'évènement que l'on cherche à estimer. L'estimation d'un quantile à 0,5% nécessite généralement davantage de scénarios pour être stable que l'estimation d'une moyenne. Dans le cours d'agrégation des risques [12], Alexandre Boumezoued obtient une erreur d'estimation du SCR de l'ordre de 1% lorsqu'il génère 10 000 scénarios monde réel. Elle devient négligeable avec l'utilisation de 15 000 scénarios. Ce nombre de scénarios dépend cependant du contrat étudié, de l'allocation d'actifs, des modèles utilisés par le GSE, etc comme nous l'avons expliqué précédemment.

7. $Base_{test-2}$ est obtenue après environ 30 heures de calcul. L'obtention de cette base a nécessité plusieurs essais successifs, l'ordinateur n'étant pas parvenu au bout de l'opération lors des premiers essais.

8. En utilisant 1 000 scénarios MR entre 0 et 1 et entre 1 et 2 nous avons 1 000 000 d'environnements économiques à horizon deux ans qu'il nous est impossible de stocker en mémoire.

Dans notre cas, l'outil que nous utilisons pour mettre en place la méthode des SdS a déjà fait l'objet d'étude de convergence sur notre contrat retraite. L'étude consistait à projeter N fois le bilan initial de l'assureur du 31/12/2016 au 31/12/2017. À l'issue de ces N projections, le SCR initial était calculé. L'opération était alors répétée en faisant varier N de 1 à 10 000, afin d'apprécier la vitesse de convergence du SCR :

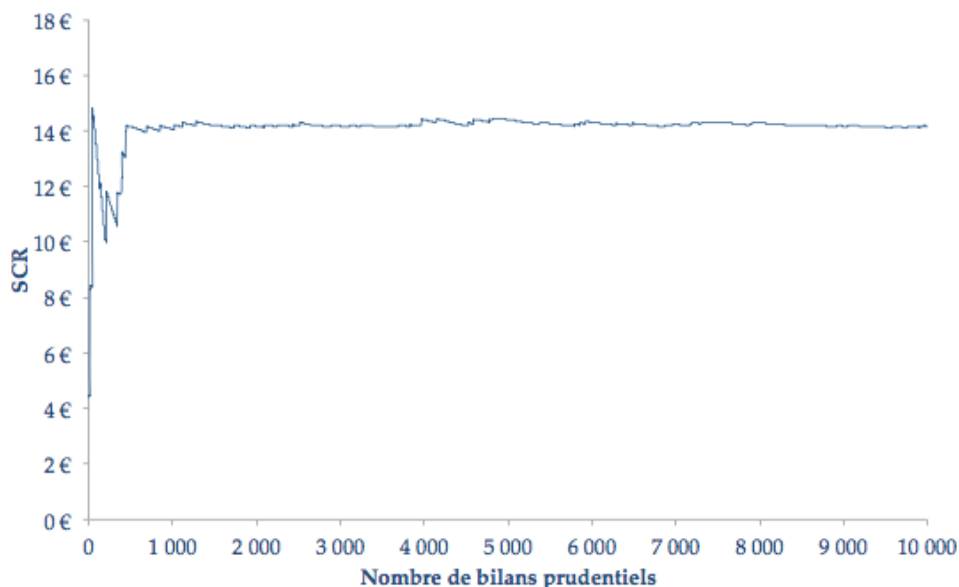


FIGURE 3.17 – Vitesse de convergence du SCR

Une telle étude est particulièrement complexe et longue à réaliser. Nous nous baserons donc sur les résultats obtenus, qui suggèrent que 1 000 à 2 000 scénarios primaires pourraient suffire à obtenir une estimation convenable du SCR dans notre cas.

Dans ce mémoire, nous cherchons tout d'abord à étudier les performances des modèles pour estimer un SCR vision modèle interne pour différentes allocations d'actifs. Ce faisant, nous chercherons à déterminer le modèle le plus performant sur la zone quantile. Nous souhaitons alors avoir accès à une quantité d'information suffisante et préférons utiliser 10 000 scénarios pour $Base_{test-2}^{t=1}$. Le SCR obtenu sera également plus stable et garanti un cadre plus général aux résultats obtenus.

Lorsque l'on cherche à estimer une distribution du ratio de solvabilité futur, nous devons calculer plusieurs SCR en date $t=1$ et le choix d'un nombre de scénarios aussi élevé est impossible. Nous utiliserons donc 10 000 scénarios monde réel pour une estimation ponctuelle du SCR en $t=0$, mais seulement 1 000 scénarios monde réel pour estimer le SCR en date $t=1$ lorsque nous cherchons à estimer une distribution du ratio de solvabilité par la méthode des SdS, nous reviendrons sur ce cas plus en détail au chapitre suivant.

Nous allons maintenant parler du choix du nombre de scénarios RN pour l'estimation de la provision *Best Estimate*. Selon Barrie & Hilbert [25], les assureurs utilisent entre 1 000 et 10 000 scénarios risque neutre pour l'estimation de la provision *Best Estimate*. La volatilité de la provision *Best Estimate* en $t=0$ ne dépasse pas 0,7% lorsque nous générons 1 000 scénarios risque neutre. Nous retenons alors 1 000 scénarios risque neutre pour les calculs de provisions *Best Estimate*.

Notons enfin, que plus le nombre de scénarios risque neutre liés à l'estimation de la provision *Best Estimate* est élevé, plus le recours à notre méthode s'avère intéressant puisqu'elle vise à se substituer à ce calcul.

Nous allons maintenant discuter de la problématique de la validation des modèles.

• Validation des modèles :

La validation des modèles se fera via différents outils. Nous utiliserons notamment les informations suivantes :

- l'erreur entre le BE prédit par les modèles et le BE théorique, l'erreur moyenne est donnée par la MSE, plus elle est faible, plus le modèle est performant,
- la performance des modèles en terme de prédiction moyenne et écart type, nous pourrons notamment estimer :
 - le BE moyen et son écart type à horizon un an,
 - le SCR et ratio de solvabilité moyen à horizon un an, ainsi que leur écart type respectifs,
- les distributions théoriques que nous pourrons comparer aux distributions obtenues pour les différents modèles pour :
 - les Fonds Propres et BE obtenus à horizon un an,
 - les SCR et ratio de solvabilité à horizon un an,
- le qq-plot qui est un outil graphique que nous présentons plus en détail en Annexe B. Il nous permet notamment de comparer deux distributions entre elles. Si les distributions sont proches les points obtenus sont alignés selon une droite. Il s'agit d'un outil précieux pour comparer les queues de distributions de deux distributions.

Après avoir présenté la méthode que nous proposons, nous allons l'appliquer à un contrat retraite afin de déterminer si elle obtient les performances escomptées. Nous proposerons tout d'abord la mise en place de la méthode pour calculer les provisions *Best Estimate* à horizon un an, afin de déterminer si nous obtenons de bonnes performances en terme de calcul de SCR et si ces performances sont généralisables à d'autres allocations d'actifs. Puis, nous appliquerons la méthode à la projection du ratio de solvabilité de l'assureur à horizon un an et nous comparerons les performances obtenues au résultat théorique obtenu par la méthode des SdSdS.

Chapitre 4

Application de la méthode à un contrat retraite

« Ce qui est difficile prend du temps, ce qui est impossible en prend un peu plus. »

Fridtjof Nansen

4.1 Présentation du contrat

Nous allons maintenant présenter le contrat retraite que nous étudions. Il va nous servir de référence pour l'utilisation de la méthode.

4.1.1 Les engagements de l'assureur au 31/12/2016

Le contrat que nous utilisons est un contrat collectif d'assurance retraite re-valorisable. Il vise à verser aux assurés une pension complémentaire lorsqu'ils partiront à la retraite. Le contrat n'offre pas la possibilité d'un rachat anticipé aux assurés. Les prestations sont versées sous forme de rente.

Lors de l'évaluation de la provision *Best Estimate*, le portefeuille est supposé en *run off*. L'échéancier des flux de prestations hors revalorisation peut alors être calculé via l'utilisation de table de mortalité. Les flux de prestations des maturités les plus longues ne sont pas significatifs. L'horizon de projection présente pourtant un coût important lors de l'utilisation du GSE et de la génération de scénarios économiques. Afin de palier à cette problématique, nous arrêtons la projection des flux de prestations à 60 ans. Les flux de prestations ultérieurs sont alors ramenés à la dernière maturité. Cette simplification ne modifie pas les conclusions de l'étude. Nous obtenons alors l'échéancier suivant au 31/12/2016 :

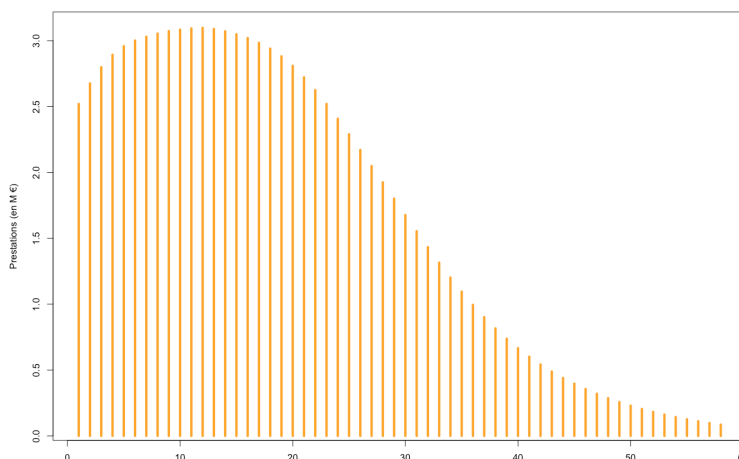


FIGURE 4.1 – Échéancier des flux de prestations non revalorisées vu au 31/12/2016

Les rentes promises par l'assureur sont revalorisées chaque année. Dans notre étude, la politique de revalorisation est modélisée de la manière suivante :

	Rendement < 5%	-5% ≤ Rendement ≤ 0%	Rendement > 0%
Inflation < 0%	0%	0%	0%
0% ≤ Inflation ≤ 4%	0%	50% × Inflation	Max(0%; Min(Inflation ; Rendement -1%))
Inflation > 4%	1%	2%	Max(0%; Min(Inflation ; Rendement -1%))

TABLE 4.1 – La politique de revalorisation de l'assureur

Le taux de rendement est la somme :

- du rendement obligataire sur l'année : on considère qu'il s'agit de la performance obligataire sur l'année à laquelle on ajoute le coupon reçu,
- du rendement action.

4.1.2 Le bilan initial de l'assureur

• L'actif de l'assureur :

L'actif de l'assureur est composé :

- d'une poche action équi-répartie sur l'indice CAC40,
- d'une poche obligataire reflétant la duration réelle du portefeuille.

L'allocation initiale d'actifs est de 30% d'actions pour 70% d'obligations. La valeur de marché de l'actif est de 110M €. Nous allons maintenant calculer le bilan prudentiel simplifié de l'assureur et notamment calculer la provision *Best Estimate* via le GSE.

• Le bilan de l'assureur au 31/12/2016 :

Le bilan initial de l'assureur est le suivant :

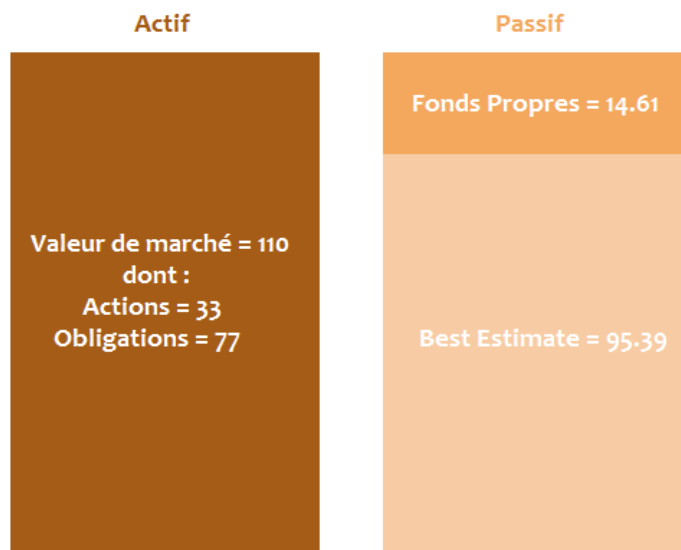


FIGURE 4.2 – Bilan Prudentiel de l'assureur au 31/12/2016

4.2 Utilisation de la méthode pour calculer un SCR vision modèle interne

4.2.1 Apprentissage et calibrage des modèles

• La phase d'apprentissage et de calibration des modèles :

Nous obtenons par la méthode décrite en 3.3.2 une base de données composée de 15 300 lignes et 145 colonnes. Nous divisons cette base en une base d'apprentissage $Base_{app}^{t=1}$ et une base de test $Base_{test-1}^{t=1}$. $Base_{app}^{t=1}$ est composée de 10 200 couples (X,Y). Y est le vecteur des 5 100 provisions *Best Estimate* futures. Nous retirons des variables explicatives le BE futur, ainsi que toutes les variables auxquelles nous n'auront pas accès sans utiliser les SdS (SCR initial, ratio de solvabilité initial, etc). La matrice X des variables explicatives est alors de taille $10\,200 \times 140$. Elle est notamment composée des éléments suivants :

- le BE en $t=0$ ainsi que les Fonds Propres en $t=0$ auxquels s'ajoutent certains paramètres de l'environnement économique initial (inflation initiale, etc),
- l'univers économique en date $t=1$: courbe des taux reconstituée, inflation, flux des prestations futures revalorisées,
- la valeur de marché de l'actif bilan prudentiel en date $t=1$.

• Le choix des variables explicatives :

Nous effectuons un premier apprentissage des modèles sur cette base. Il nous permet de re-traiter les variables explicatives de façon à supprimer des informations non pertinentes ou redondantes.

Les modèles XGBoost et RF permettent notamment de classer les variables explicatives selon leur importance. Les variables ayant le pouvoir explicatif le plus faible ne sont pas nécessairement les mêmes pour chaque modèle. Nous devons donc faire des recoupements et comprendre quelles variables il convient de supprimer. Par exemple, il peut arriver qu'un taux zéro coupon (issu de la courbe des taux reconstituée) de maturité X ait un pouvoir explicatif plus faible qu'un taux zéro coupon de maturité plus éloignée. Pour autant, il nous paraît peu pertinent de délaissier l'information d'une échéance plus faible au profit d'une échéance plus éloignée.

De manière générale, il semble cependant que les taux zéro coupons ont un pouvoir explicatif relativement élevé (plus particulièrement pour les maturités les plus faibles), nous décidons donc de les conserver. Nous pourrions éventuellement envisager de supprimer cette information pour les maturités les plus éloignées, mais comme il n'y a pas de redondance ici, nous préférons nous abstenir de supprimer ces informations. De même l'inflation entre $t=0$ et $t=1$ et l'allocation d'actifs choisie ont un pouvoir explicatif élevé.

Le flux des vecteurs de prestations futur est revalorisé pour chaque maturité. Nous avons donc un vecteur de 60 flux de prestations revalorisés. Ces informations sont redondantes, nous créons alors une variable contenant la somme des *Cash Flows* futurs revalorisés que nous nommons « CF ». Ainsi, en cas de bonne performance de l'actif la somme sera élevée et inversement. Cette variable a un pouvoir prédictif fort et nous supprimons les informations redondantes que représentent les *Cash Flows* futurs revalorisés pour chaque maturité. D'autres variables présentent un intérêt relativement faible, telles que des informations sur l'environnement économique initial (inflation initiale, etc). Après retraitement de ces variables, il nous reste 78 variables explicatives.

Nous n'effectuons pas ce retraitement à la base de données que nous fournissons au SVM. En effet, nous avons vu que ce modèle est davantage capable de tirer parti d'une quantité d'information élevée. De plus, il ne permet pas de visualiser le pouvoir explicatif des différentes variables comme nous pouvons le faire avec XGBoost ou RF.

Une fois ce re-traitement effectué, nous regardons les 20 variables ayant le pouvoir explicatif le plus fort. Il s'agit des variables les plus utilisées par les arbres et le plus « haut » dans les arbres.

Voici les résultats obtenus :

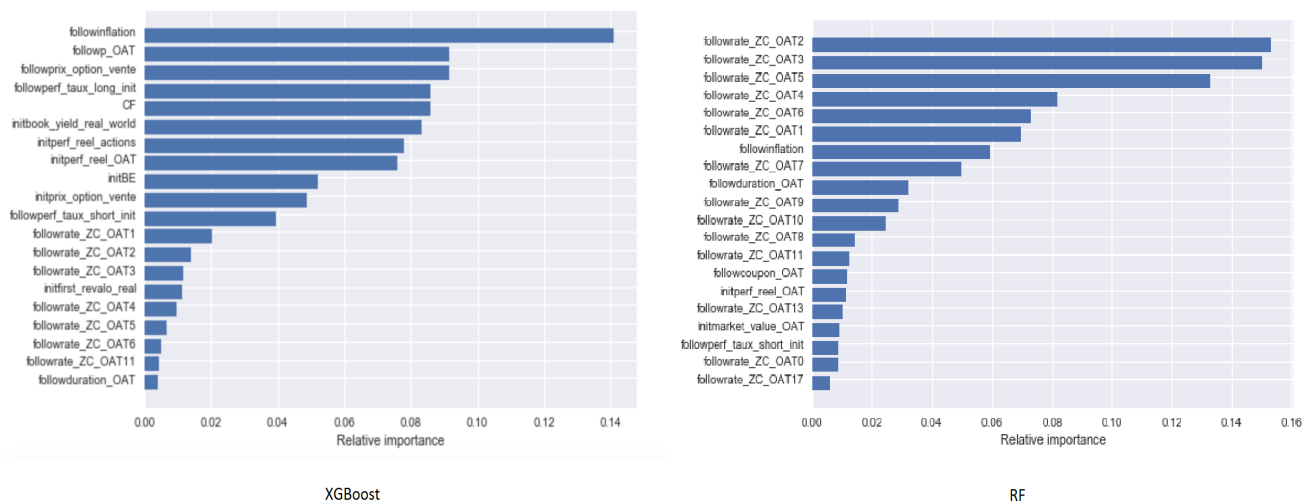


FIGURE 4.3 – Variables explicatives les plus pertinentes pour les modèles XGBoost et RF

Le modèle XGBoost semble ainsi privilégier l'inflation future là où le RF va se concentrer davantage sur les courbes des taux. Le RF accorde néanmoins beaucoup d'importance à l'inflation, ce qui semble logique au vu de la politique de revalorisation présentée plus haut.

XGBoost se concentre davantage sur les flux de prestations, de part l'importance qu'il accorde à la variable CF comprenant la somme des *Cash Flows* futurs revalorisés, mais également via le BE initial. À l'inverse le RF semble tenir peu compte de ces variables.

Nous pouvons imaginer que XGBoost prédit le BE avec un premier arbre basé sur l'inflation future qu'il va ensuite corriger en fonction de la part action en portefeuille et des autres variables de l'environnement économique futur.

Ces résultats nous amènent à nous interroger :

- la *Random Forest* semble tenir davantage compte de la courbe des taux que XGBoost, obtient-il de meilleures performances dans les scénarios de taux extrêmes ?
- la *Random Forest* ne semble pas privilégier l'allocation d'actifs comme variable explicative : a-t-il davantage de difficultés à prédire le BE pour des allocations d'actifs « extrêmes » ?

• Les paramètres retenus pour les modèles :

Après avoir sélectionné les variables explicatives les plus pertinentes, nous optimisons les paramètres de chaque modèle. Cette phase de calibrage s'effectue via validation croisée. La validation croisée nous permet de choisir les paramètres a priori les plus optimaux pour chaque modèle. Cependant cette opération reste coûteuse en temps de calcul et le nombre de possibilités est restreint. Notons également que certains paramètres vont amener à un modèle plus long à utiliser en pratique. Or, lorsque nous cherchons à utiliser le modèle un très grand nombre de fois, il faut tenir compte de cette difficulté. C'est notamment le cas si l'on utilise un nombre d'estimateurs très élevé pour des méthodes ensemblistes

telles que la *Random Forest* ou la descente de gradient.

Voici les paramètres retenus pour le *Ranfom Forest* :

- `max_features` = 70,
- `max_depth`=15,
- `min_sample_split` = 5,
- `min_sample_leaf` = 5,
- `Bootstrap` = True,
- `n_estimators` = 1000.

Voici les paramètres retenus pour XGBoost :

- `n_estimators` = 1000,
- `learning_rate` = 0.02,
- `max_depth`=5,
- `subsample`=0.5.

Voici les paramètres retenus pour le SVM :

- `kernel` = poly,
- `degree` = 2,
- `gamma`=0.1,
- `C`=5,
- `Coef0`=-5.

4.2.2 Comparaison des différents modèles

Une fois la calibration des modèles effectuée, nous testons les différents modèles sur la base de test $Base_{test-1}^{t=1}$. Cette base de test est constituée de 30% des données initiale soit 5 100 lignes et 140 colonnes. Nous effectuons le re-traitement appliqué à $Base_{app}^{t=1}$ afin d'obtenir les couples entrées/sorties (X,Y), avec Y le vecteur des 5 100 provisions *Best Estimate* obtenues par la méthode des SdS. X est la matrice des variables explicatives, elle est de dimension 5 100 × 78. Nous fournissons à chaque modèle la matrice X en entrée et le modèle nous fournit en sortie un vecteur \hat{Y} des provisions *Best Estimate* à horizon un an estimées par le modèle.

• Comparaison des résultats obtenus par les différents modèles :

Nous calculons la MSE obtenue par chaque modèle sur $Base_{test-1}^{t=1}$, le BE moyen prédit et l'écart type des BE obtenus. Nous obtenons alors les résultats suivants :

Modèle	SdS	XGB	RF	SVM
MSE	0	0,83	0,99	1,00
BE moyen	92,79	92,81	92,82	92,84
Écart type BE	7,49	7,41	7,38	7,37

TABLE 4.2 – Performance des modèles sur $Base_{test-1}^{t=1}$

Le modèle XGBoost obtient les meilleures performances en terme de MSE, BE moyen prédit et écart type du BE prédit. Toutefois, malgré une MSE plus élevée, RF et SVM semblent obtenir une prédiction du BE moyen et de l'écart type acceptable.

Nous confirmons ces résultats avec la comparaison des densités reconstituées obtenues par les modèles en rouge à la densité théorique en bleu :

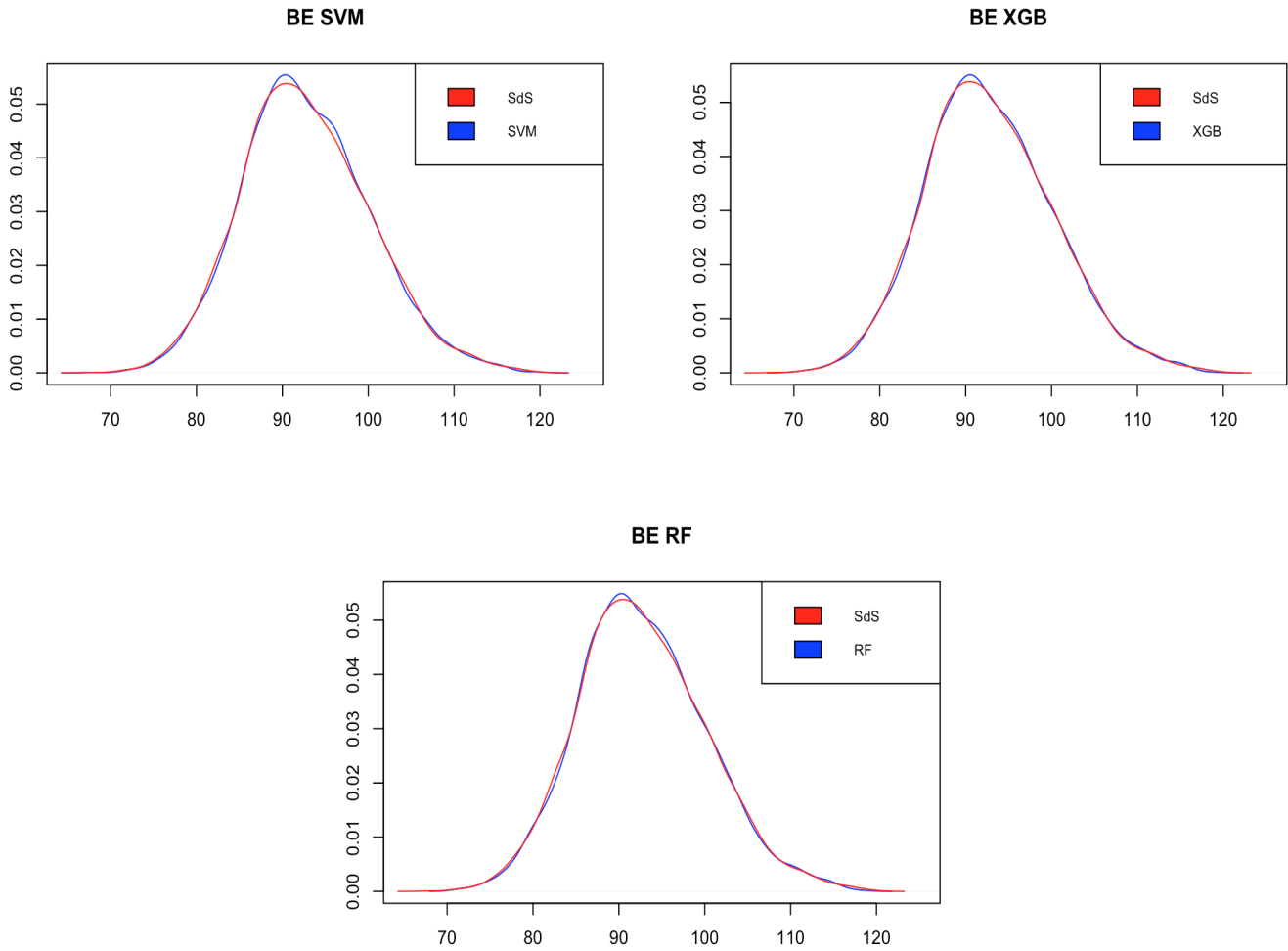


FIGURE 4.4 – Comparaison des densités reconstituées des BE

On note que les densités reconstituées des BE sont proches de la densité théorique pour les trois modèles. RF semble d'ailleurs plus proche de la distribution théorique que le SVM ce qui peut signifier qu'il commet davantage d'erreur au niveau des queues de distribution (puisque la MSE des deux modèles est proche).

Nous complétons cette étude par une mesure de l'erreur en % commise par chaque modèle, afin de regarder comment se répartissent les erreurs d'un modèle à l'autre. Nous calculons cette erreur de la manière suivante :

$$Erreur_{\%} = 100 \times \text{abs}\left(\frac{BE_{\text{modele}} - BE_{SdS}}{BE_{SdS}}\right).$$

Nous regardons la répartition des erreurs entre les différents modèles. Nous obtenons les résultats suivants :

Modèle	XGBoost	RF	SVM
$\max(Erreur_{\%})$	3,80	5,32	3,79
$Var_{99\%}(Erreur_{\%})$	2,49	2,78	2,71
$Var_{90\%}(Erreur_{\%})$	1,64	1,78	1,79

TABLE 4.3 – Répartition des erreurs entre les différents modèles

On note que XGBoost et SVM ont une erreur maximale proche. Cependant, lorsque l'on s'intéresse aux VaR à 95% et 99% des erreurs, XGBoost semble clairement le meilleur modèle. L'écart entre SVM et RF semble également s'atténuer avec une performance relativement proche. Nous complétons alors cette étude par un qq-plot entre les erreurs $Erreur_{\%}$ commises par les différents modèles :

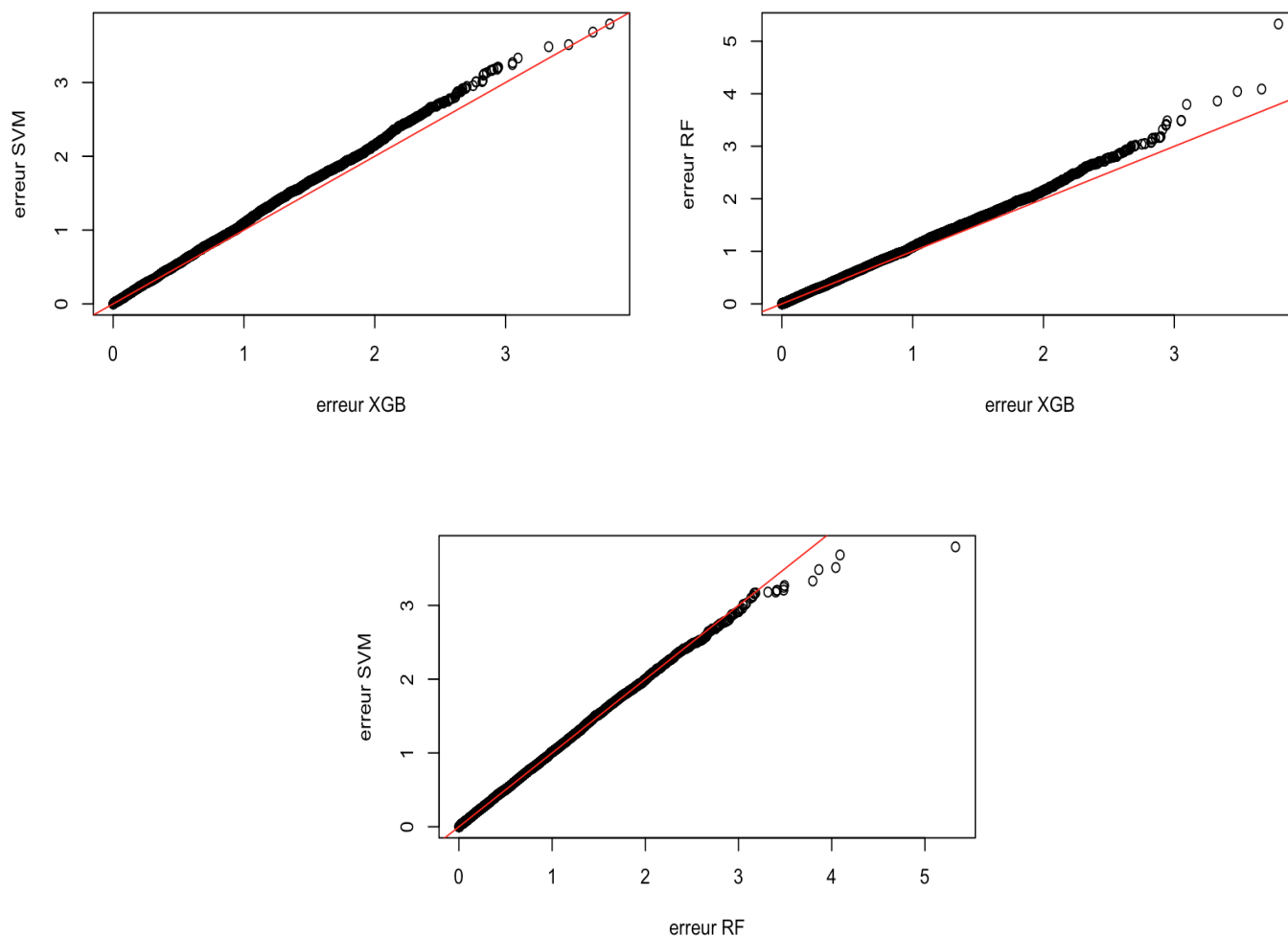


FIGURE 4.5 – Comparaison de la répartition des erreurs commises par les différents modèles

Ces résultats confirment la meilleure répartition des erreurs de la part du modèle XGBoost par rapport aux autres modèles. On note également une sur-performance du SVM sur le RF.

Ainsi, de nos deux méthodes ensemblistes, XGBoost semble sur-performer clairement le RF. Pour autant, il ne s'agit ici que d'erreurs relatives et nous ignorons encore dans quels scénarios se produisent les erreurs les plus élevées. Nous allons à présent poursuivre l'étude en cherchant à déterminer quels scénarios amènent les modèles à faire les plus grosses erreurs.

• Étude des scénarios les plus défavorables pour les modèles :

Différentes questions se posent quant aux scénarios qui amènent aux BE les plus mal prédits :

- ces scénarios sont-ils les mêmes pour chaque modèle ou au contraire certains modèles sont-ils plus performants dans certains cas ?
- les modèles obtiennent-ils des performances uniformes quelle que soit l'allocation d'actifs testée ? Est-ce que certaines allocations d'actifs sont plus complexes à prévoir ?

Nous déterminons les 500 scénarios les plus mal prédits par les différents modèles. Nous obtenons les scénarios menant aux 10% d'erreurs les plus élevées. Cela revient à étudier les scénarios qui mènent à des erreurs supérieures à environ 1,6-1,8 % pour les différents modèles.

Nous regardons sur un qq-plot la répartition de la part action des 500 scénarios qui mènent aux taux d'erreurs les plus élevés par rapport à la répartition de la part actions présente dans la base de test :

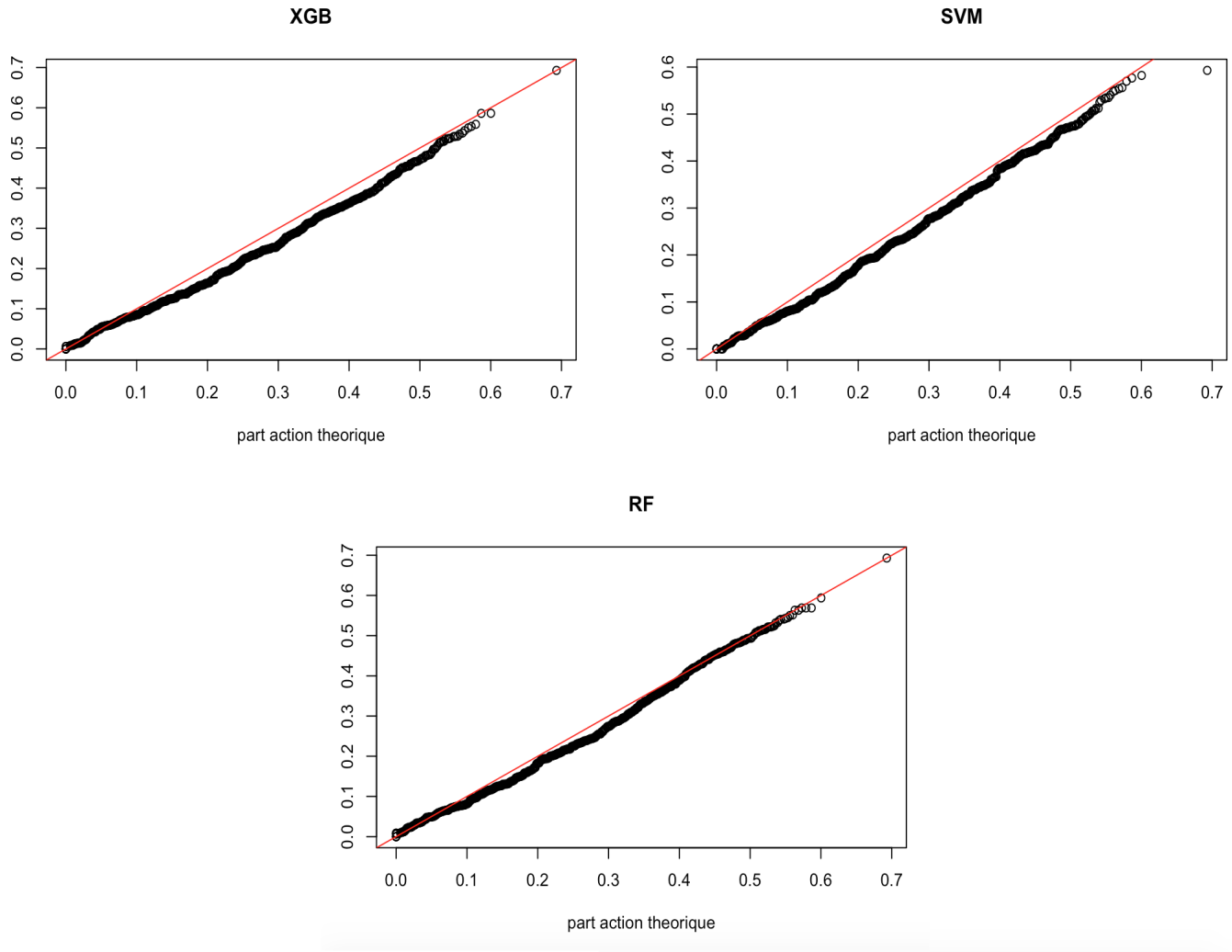


FIGURE 4.6 – Part action des scénarios les plus défavorables aux modèles

Il semble clair que les 500 scénarios qui mènent aux pires erreurs des modèles n'ont pas la même répartition de la part actions que celle présente sur la base de test. Ce phénomène est particulièrement marqué pour le SVM et XGB. On voit notamment sur le qq-plot que les tranches 10%-50% action environ des bases de données contenant les 500 pires erreurs commises par ces modèles sont sous-représentées. Il est alors possible que les plus grosses erreurs commises par ces modèles soient concentrées sur des tranches actions faibles (proche de 0%), bien que les modèles fassent ponctuellement de grosses erreurs sur d'autres tranches.

Afin de mieux comprendre le lien entre l'erreur et l'allocation d'actifs, nous effectuons une régression linéaire (avec la fonction `lm` de R) sur l'ensemble de la base de test (soit 5 100 données) entre la MSE par les modèles et l'allocation d'actifs que nous regroupons par classe. Nous créons 12 classes d'allocations d'actifs, les 11 premières classes sont construites de la manière suivante : la classe $i+1$ regroupe les allocations d'actifs contenant de $5 \times i\%$ à $5 \times i+5\%$ d'action, i allant de 0 à 10. La 12^{ème} classe regroupe les allocations d'actifs entre 60% actions et le maximum d'actions présent dans la base

de test afin de contenir suffisamment d'observations.

Voici les MSE obtenues par les différents modèles sur les différentes tranches actions de notre base de test :

Tranche action	MSE SVM	MSE XGB	MSE RF
0%-5%	1,15	0,91	1,04
5%-10%	1,07	0,96	1,15
10%-15%	1,16	0,95	1,10
15%-20%	0,94	0,88	0,95
20%-25%	0,99	0,83	0,92
25%-30%	1,06	0,90	1,08
30%-35%	0,94	0,79	0,91
35%-40%	0,88	0,78	0,98
40%-45%	0,82	0,69	0,82
45%-50%	1,03	0,77	1,06
55%-60%	0,89	0,67	0,89
60%-100%	0,88	0,80	0,94

TABLE 4.4 – erreur en fonction de la tranche action

Nous remarquons ainsi que l'erreur commise par les différents modèles semble plus forte sur la tranche [0% ; 15%] action quel que soit le modèle. Les différents modèles obtiennent globalement de meilleures performances lorsque la part action en portefeuille augmente. Nous notons également que le modèle XGBoost sur-performe les autres modèles quelle que soit la tranche action étudiée.

Nous effectuons ensuite une régression linéaire entre la MSE obtenue et la part action.

Nous obtenons alors un modèle de la forme :

$$MSE_i = Intercept + Coef f_1 \times ClasseAction_i + \epsilon,$$

avec :

ϵ une variable aléatoire centrée de variance finie (le résidu du modèle).

Sous R, nous testons l'hypothèse $H_0 : Coef f_1 = 0$, si la p-value du test est inférieure à un seuil α , nous rejetons l'hypothèse nulle. Rejeter l'hypothèse nulle revient dans notre cas à estimer qu'il y a un lien entre l'erreur commise en terme de prévision du BE et le pourcentage d'action présent dans le portefeuille. On utilise généralement un seuil α de 5% ou 1%. Plus la p-value est faible plus nous avons une présomption forte contre l'hypothèse nulle.

Si la p-value est élevée, cela signifie que le lien entre la variable à expliquer (l'erreur dans notre cas) et la variable explicative (la part action dans notre cas), n'est pas clairement établi. Une autre information souvent utilisée dans le cadre de la régression linéaire est le R^2 (ou le $R^2_{a,justé}$ qui pénalise la complexité du modèle). Ce coefficient mesure le pourcentage de variance expliquée par la régression. Si les observations sont bien expliquées par le modèle, ce coefficient est proche de 1.

Nous obtenons les résultats suivants :

Modèle	Intercept	$Coef f_1$	p-value	$R^2_{a,justé}$
XGB	0,97	-0,43	0,0007	0,66
RF	1,08	-0,3	0,06	0,24
SVM	1,13	-0,45	0,006	0,55

TABLE 4.5 – Régression linéaire erreur en fonction de la part action en portefeuille

Nous pouvons rejeter l'hypothèse H_0 pour les modèles XGB et SVM. La p-value obtenue est en effet inférieure à 1% et le $R_{a,justé}^2$ est relativement élevé : il y a un lien entre la part action et l'erreur commise par ces modèles. Nous notons que l'hypothèse alternative des trois régressions linéaires propose un $Coeff_1$ négatif. Cela signifie, que la performance obtenue par les différents modèles semble s'améliorer lorsque la part action en portefeuille augmente. Le $Coeff_1$ estimé dans l'hypothèse alternative pour SVM et XGB est le plus faible, ces modèles semblent donc fortement sensibles à la part action en portefeuille. Il faut en tenir compte lors de l'utilisation des modèles.

Le RF obtient une p-value supérieure à 5% et un $R_{a,justé}^2$ relativement faible. Nous ne pouvons donc pas rejeter l'hypothèse nulle. Ce résultat est particulièrement intéressant, puisque a priori nous ne pouvons pas affirmer que les performances du RF seront significativement impactées par la part action en portefeuille. Nous pourrions alors envisager d'utiliser ce modèle pour des part actions faibles. Ce résultat est également à mettre en parallèle avec l'importance accordées à la variable explicative de la part action par les modèles XGB et RF. On avait notamment noté que RF ne privilégiait pas l'information de la part action future en portefeuille. Cela peut expliquer le fait que l'erreur commise par ce modèle ne soit que peu (ou pas) sensible à la part action.

Nous allons maintenant nous intéresser aux performances des modèles pour prédire le SCR de l'assureur retraite.

4.2.3 Le calcul du SCR vision modèle interne

- Quelques résultats préliminaires obtenus par les SdS :

L'obtention de la base de test $Base_{test-2}^{t=1}$ nous amène différentes informations quant aux données de l'assureur.

Nous pouvons notamment nous intéresser aux évolutions respectives de la valeur de marché de l'actif du bilan prudentiel, de la provision *Best Estimate* et des Fonds Propres à horizon un an. Au 31/12/2016 nous avons ainsi une valeur de marché de l'actif de 110M €, un BE initial de 95,39M € et des Fonds Propres de 14,61M €. Au 31/12/2017, la valeur de marché de l'actif moyenne est de 110,29M €, les Fonds Propres moyens s'élèvent à 17,80M € alors que la provision *Best Estimate* moyenne est de 92,78M €.

Nous obtenons les distributions suivantes des Fonds Propres et provision *Best Estimate* dans un an :

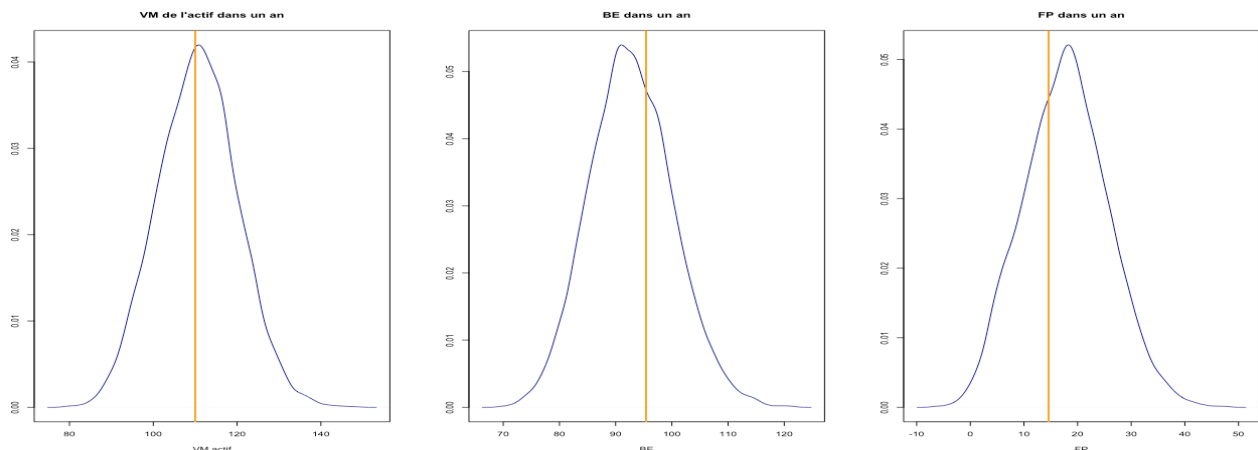


FIGURE 4.7 – distribution des Fonds Propres et *Best Estimate* dans un an

La droite orange représente le niveau initial de la valeur de marché de l'actif, du BE, ou des FP.

On note donc une nette amélioration des Fonds Propres, alors que la valeur de marché de l'actif semble relativement stable en moyenne. Cette hausse des Fonds Propres futurs semble être davantage expliquée par une baisse de la provision *Best Estimate* à horizon un an.

Pour expliquer ces résultats au niveau de la provision *Best Estimate*, nous devons nous rappeler de la politique de revalorisation de l'assureur. Lorsque le rendement de l'assureur est élevé, ou très élevé, la revalorisation est notamment capée par l'inflation. Cela signifie que dans les environnements économiques les plus favorables à l'assureur, la valeur de marché de l'actif augmente fortement, alors que les engagements de l'assureur ne sont que faiblement revalorisés. Lorsque l'environnement économique est moins favorable via un rendement trop faible, ou une inflation basse, les engagements de l'assureur ne sont peu ou pas revalorisés. Notons également que la part action relativement élevée présente dans le portefeuille de l'assureur amène à une valeur de marché de l'actif plutôt volatile, qui explique également en partie l'évolution des Fonds Propres futurs.

Après cette analyse sommaire de la distribution future des Fonds Propres et de la provision *Best Estimate*, nous allons nous intéresser aux performances de la méthode pour estimer le SCR de l'assureur.

• Comparaison des différents modèles :

Nous débutons notre étude par le calcul de la MSE entre les 10 000 provisions *Best Estimate* obtenues par les SdS et par les différents modèles. Nous calculons également le SCR obtenu par les SdS que nous comparons au SCR obtenu via l'utilisation des différents modèles. Nous en déduisons l'écart relatif en % entre le SCR prédit par le modèle et le SCR obtenu par la méthode des SdS. Le SCR obtenu par la méthode des SdS est de 14,89M €.

Voici un tableau présentant les résultats :

Modèle	$(MSE(Base_{test-1}^{t=1}))$	$MSE(Base_{test-2}^{t=1})$	SCR prédit	Écart relatif
XGB	(0,83)	0,81	15,06	1,11%
RF	(0,99)	0,98	15,52	4,20%
SVM	(1,00)	0,95	14,95	0,35%

TABLE 4.6 – Comparaison des différents modèles

Afin de faciliter la lecture, nous rappelons entre parenthèses les résultats obtenus en terme de MSE sur notre première base de test diversifiée. Les modèles obtiennent une MSE légèrement plus faible que ce qu'ils obtenaient sur $Base_{test-1}^{t=1}$, ce phénomène est davantage marqué pour le SVM. Ce résultat est à mettre en parallèle avec l'hypothèse selon laquelle les modèles sont plus performants lorsque la part action augmente.

Les trois modèles obtiennent des performances acceptables en terme de MSE et de capacité à prédire un SCR. La MSE de XGBoost sur cette base est particulièrement bonne, toutefois le SVM est plus précis au niveau de l'estimation SCR, nous allons donc essayer de comprendre cette meilleure performance.

Notons également que les trois modèles surestiment le SCR. Si le SCR estimé est supérieur au SCR réel, cela signifie que le quantile à 0,5% des Fonds Propres estimés à horizon un an est inférieur au quantile réel des Fonds Propres à horizon un an. Il est alors probable que nos modèles tendent à surestimer légèrement la provision *Best Estimate*. Nous allons chercher à déterminer si c'est effectivement le cas.

Pour ce faire, nous traçons les histogrammes des provisions *Best Estimate* en date $t=1$ par chacun des modèles :

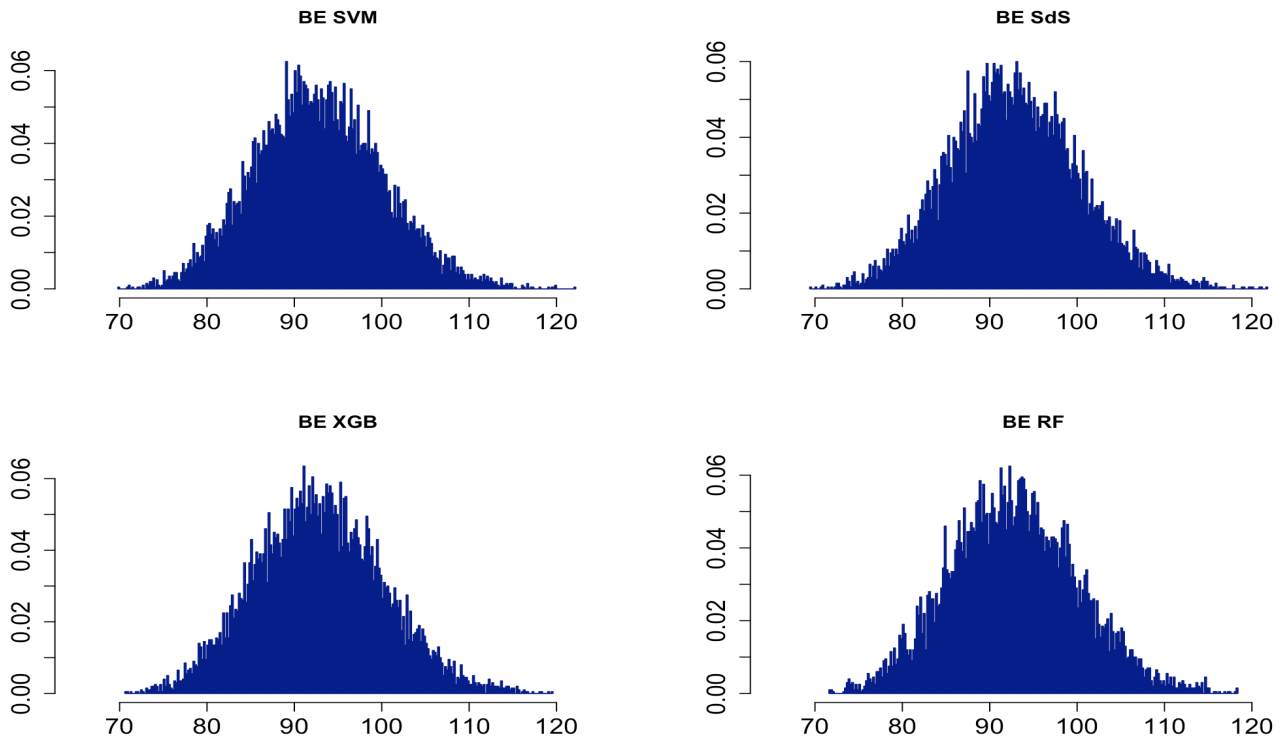
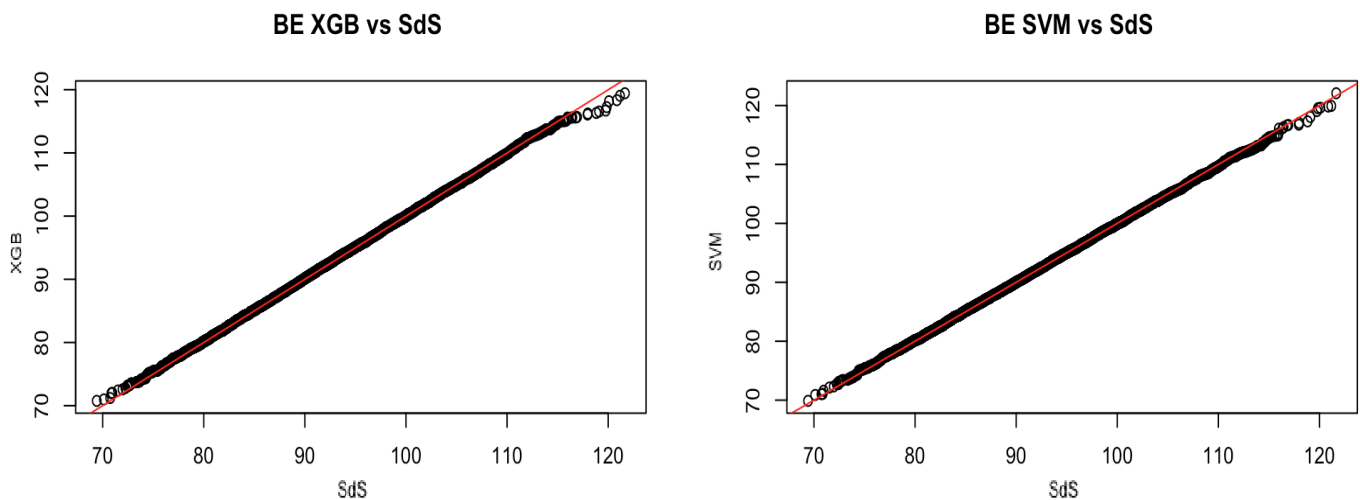


FIGURE 4.8 – Comparaison des provisions *Best Estimate* obtenues

Ce graphique nous apporte des éléments de réponse. Nous voyons en effet que le modèle SVM semble obtenir une distribution des BE futurs plus proche de celle obtenue par les SdS, notamment pour les valeurs extrêmes des provisions *Best Estimate* (le SVM est le seul avec les SdS à obtenir des provisions *Best Estimate* supérieures à 120M €). Ce phénomène permet-il d'expliquer la meilleure performance du modèle SVM en terme de prévision du SCR ?

Pour nous en assurer, nous devons étudier ce qui se passe au niveau des Fonds Propres. Avant de passer à cette étude, nous souhaitons vérifier la répartition des quantiles des provisions *Best Estimate* obtenues par les différents modèles. Nous proposons d'effectuer un qq-plot entre les BE estimés par chaque modèle et les BE théoriques obtenus par les SdS :



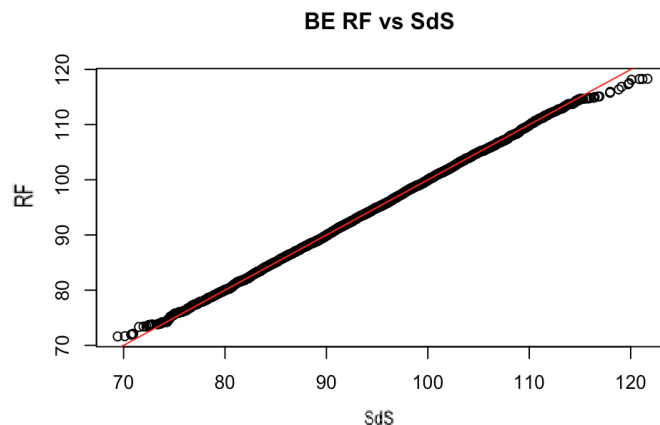


FIGURE 4.9 – qq-plot modèles vs SdS

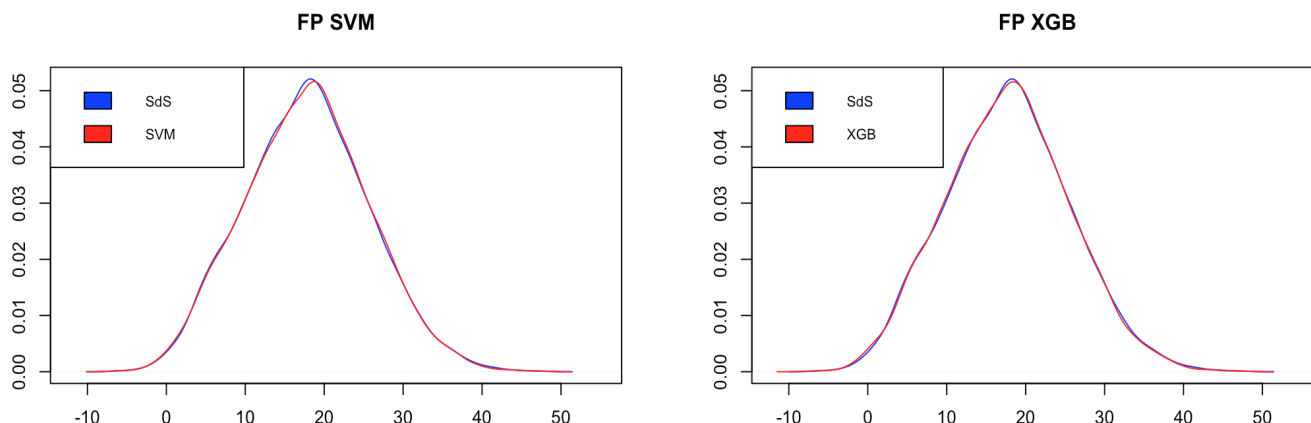
Ces graphiques nous confirment que les queues droites des distributions des BE obtenues par les modèles XGBoost et RF sont plus légères que la queue droite de la répartition théorique. Autrement dit, ces modèles ne parviennent pas à estimer correctement les BE les plus élevés : ils sont sous-estimés par ces modèles.

En plus de cette erreur, le RF a également des difficultés à estimer la queue gauche de la distribution des BE futurs. La queue gauche de la distribution obtenue par le modèle RF est en effet trop légère par rapport à la distribution théorique. Le RF semble donc le moins adapté à prédire des BE « extrêmes ». Nous pouvions déjà noter ce phénomène lors de la comparaison des histogrammes des BE prédits par les différents modèles.

Les scénarios qui mènent à des BE aussi extrêmes sont potentiellement ceux qui nous intéressent lorsque nous cherchons à estimer le quantile, ce qui pourrait expliquer la moins bonne performance du RF pour l'estimation du SCR.

Toutefois, un BE élevé n'implique pas nécessairement une valeur des Fonds Propres faible. En effet, la hausse du BE peut être compensée par une hausse similaire au niveau de la valeur de marché de l'actif. Nous allons donc maintenant nous intéresser aux distributions des Fonds Propres futurs obtenues par les différents modèles.

Nous comparons la densité des Fonds Propres à horizon un an obtenue par les différents modèles (en rouge), à la densité théorique (en bleu) obtenue par les SdS :



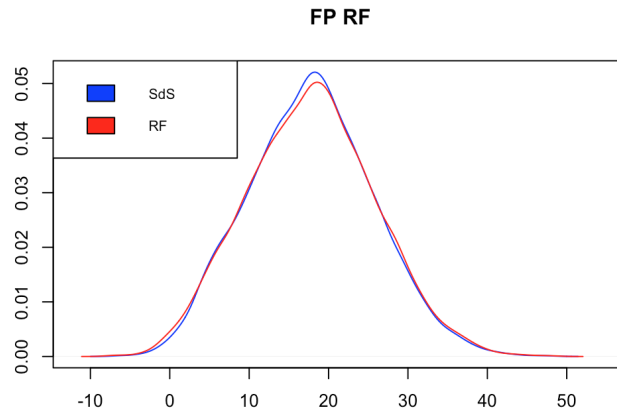


FIGURE 4.10 – Comparaison des densités obtenues par les modèles à la densité obtenue par les SdS

XGBoost est le plus proche de la densité théorique. Ce résultat n'est guère surprenant au vu de la faible MSE qu'il obtient. On note une petite erreur des modèles SVM et RF au niveau du centre de la distribution. Cependant, il ne s'agit pas de la zone qui nous intéresse pour l'estimation du SCR. Le SVM semble en revanche le plus proche de la distribution des Fonds Propres théoriques au niveau de la queue gauche. On note également un « décrochage » du modèle RF assez visible sur la queue gauche.

Nous poursuivons l'analyse sur la queue gauche de la distribution des Fonds Propres. Le SCR étant de 14,89M € pour des Fonds Propres initiaux de 14,61M € le quantile à 0,5% des Fonds Propres à horizon un an est proche de 0M €. Nous effectuons alors un zoom sur cette zone :

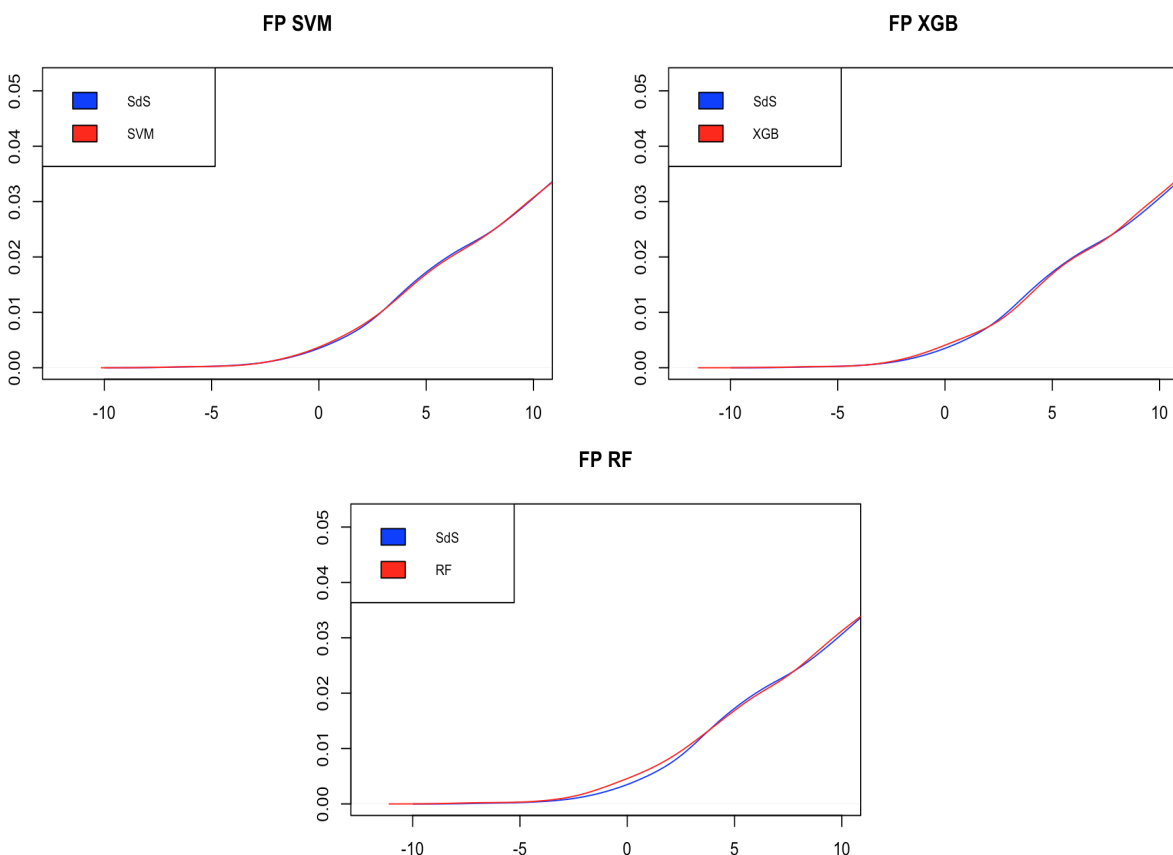


FIGURE 4.11 – Comparaison des densités obtenues par les modèles à la densité obtenue par les SdS
Zoom sur la queue gauche de la distribution

Ce zoom sur la densité des Fonds Propres dans la zone $[-12,12]$ ne montre qu'un faible décrochage de la part du modèle XGBoost. Comme attendu, le SVM est le plus proche de la densité théorique. Ce zoom ne montre toutefois qu'un décrochage assez faible des modèles par rapport à la densité théorique. On note d'ailleurs que RF semble commettre une erreur autour de la zone quantile, mais reste performant pour des Fonds Propres plus élevés. Ces résultats indiquent que la méthode semble robuste et adaptée à un calcul de SCR vision modèle interne.

Nous effectuons un second zoom afin de mettre davantage en évidence ce « décrochage » de la part des modèles RF et XGBoost au niveau de la zone quantile. Nous traçons alors la comparaison de la densité théorique des Fonds Propres (en bleu) à la densité obtenue par les modèles (en rouge) sur la zone $[-12,2]$:

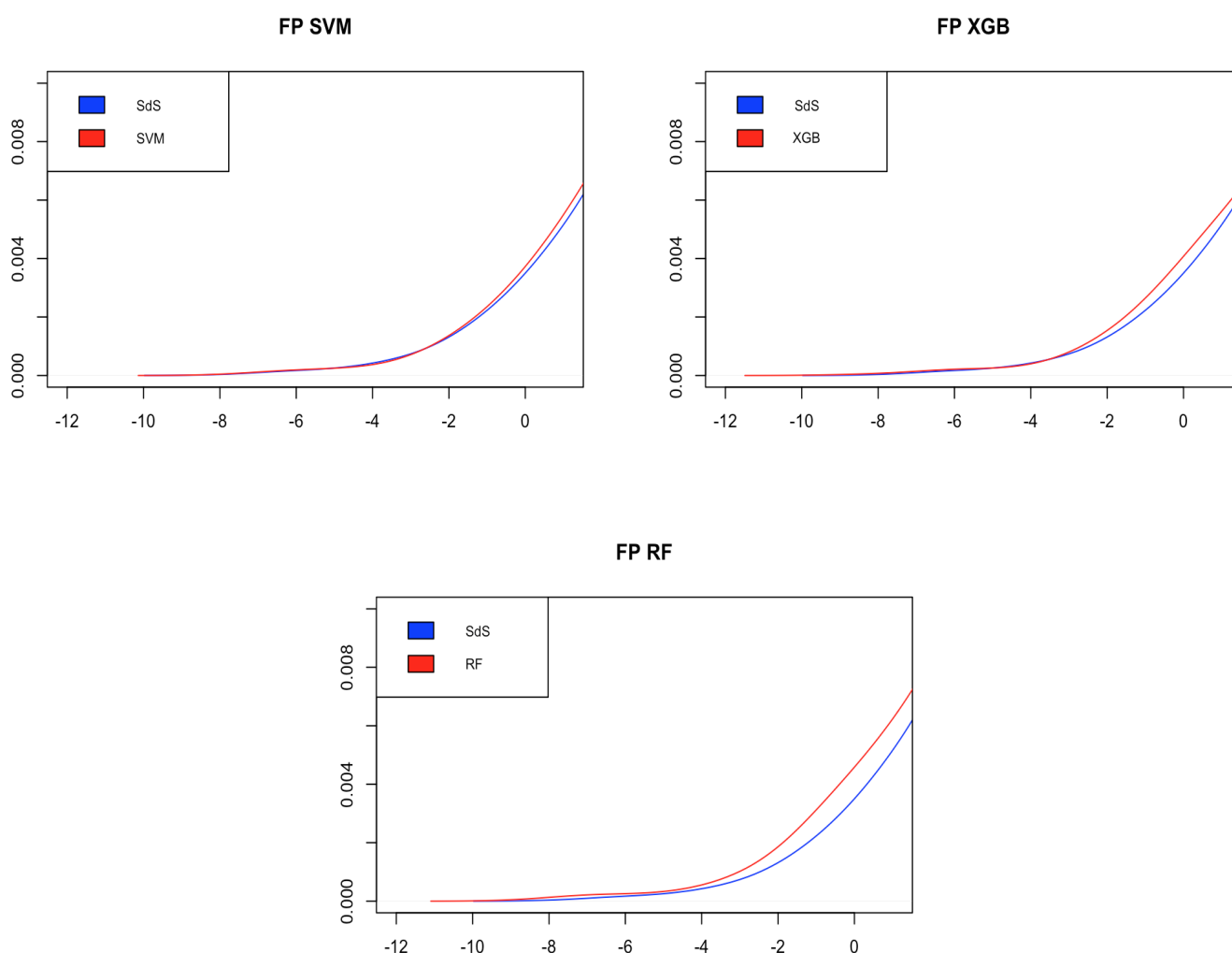


FIGURE 4.12 – Comparaison des densités obtenues par les modèles à la densité obtenue par les SdS
Zoom sur la queue gauche de la distribution

La meilleure performance obtenue par le modèle SVM est d'autant plus flagrante ici. Toutefois la distribution obtenue par XGBoost reste acceptable.

Nous terminons l'analyse des résultats par un qq-plot entre la distribution des Fonds Propres obtenus par les différents modèles et les Fonds Propres théoriques (obtenus par les SdS), en Annexe C.

Ce qq-plot va également dans le sens des résultats précédents : le SVM est le plus proche de la queue gauche de la distribution des Fonds Propres théoriques à horizon un an. On note toutefois, que la performance des modèles RF et XGB reste convenable et que le « décrochage » observé est concentré

autour de la zone quantile et ne concerne pas toute la queue gauche de la distribution.

De plus, si le modèle SVM semble le plus adapté à prédire un SCR nous devons néanmoins nous rappeler de la régression linéaire effectuée sur $Base_{test-1}^{t=1}$ expliquant l'erreur des modèles en fonction de la part action en portefeuille. La MSE du modèle SVM était en effet particulièrement sensible à la part action utilisée et l'erreur semblait diminuer significativement en augmentant la part action.

Nous allons alors nous intéresser aux performances des modèles en terme de prédiction du SCR pour des part action de 20% et 40%.

• **La performance des modèles en terme de prédiction du SCR sur des parts action de 20% et 40% :**

Nous complétons notre étude par un calcul de SCR vision modèle interne pour 20% actions et 40% actions avec 10 000 scénarios monde réel et 1 000 scénarios risque neutre (ces bases sont obtenues de la même manière que $Base_{test-2}^{t=1}$). Nous obtenons alors les résultats suivants :

Modèle	MSE 20%	SCR 20% (10,38)	Écart SCR en %
	MSE 30%	SCR 30% (14,89)	Écart SCR en %
	MSE 40%	SCR 40% (20,20)	Écart SCR en %
XGB	0,85	10,50	1,15%
	0,81	15,06	1,11%
	0,79	20,49	1,43%
RF	0,96	10,40	0,19%
	0,98	15,52	4,20%
	1,07	21,19	4,90%
SVM	0,99	9,93	-4,33%
	0,95	14,95	0,35%
	0,94	19,84	-1,78%

TABLE 4.7 – Résultats des modèles sur les parts 20%, 30% et 40% action

Nous mettons entre parenthèses le SCR théorique obtenu par la méthode des SdS. Comme nous l'avons vu lors de la régression linéaire entre la MSE des modèles et la part action en portefeuille, les modèles XGBoost et SVM obtiennent une MSE plus forte lorsque la part action baisse. Toutefois, on remarque qu'une MSE faible n'est pas nécessairement synonyme de meilleure prédiction de la zone quantile et du SCR.

On note également que le RF obtient la meilleure estimation du SCR lorsque la part action en portefeuille est de 20%. Nous pouvons également mettre en parallèle ces résultats et le tableau de la MSE par tranche actions que nous avons obtenu. Nous avons notamment noté que le RF semblait particulièrement performant sur cette zone. De plus la MSE du RF diminue lorsque l'on diminue la part action. Nous devons nous rappeler qu'il s'agissait du seul modèle pour lequel on ne pouvait affirmer l'existence d'un lien entre la part action et la MSE (on obtenait une p-value supérieure à 5% lors de la régression linéaire). Nous pourrions donc envisager d'utiliser ponctuellement ce modèle sur certaines tranches actions faibles où SVM et XGBoost seront moins performants.

La bonne performance obtenue par le SVM en terme de prédiction du SCR ne semble pas être généralisable lorsque la part action diminue. Elle s'avère plutôt volatile.

À l'inverse, XGBoost est le modèle le plus stable en terme de MSE ou de prédiction du SCR avec un écart de prédiction du SCR très faible (environ 1%) quelle que soit la part action en portefeuille et obtient systématiquement la MSE la plus faible. Il s'agit, de plus, du modèle le plus performant en

terme de temps de calcul (comme nous allons le voir).

Pour ces différentes raisons, il nous semble que c'est le modèle le plus approprié à une problématique de choix d'allocation d'actifs. Il serait néanmoins envisageable de changer de modèle en fonction de l'allocation d'actifs testée. Une autre idée pourrait être de proposer un modèle agrégatif, visant à palier aux manques des différents modèles.

Nous allons conclure l'étude à horizon un an par une comparaison entre les temps de calculs obtenus par notre méthode et la méthode des SdS pour estimer un SCR vision modèle interne.

4.2.4 La performance de la méthode en terme de temps de calcul

Les calculs ont été effectués sur un ordinateur i-mac ayant les caractéristiques suivantes :

- Processeur 2,8 GHz Intel Core i5,
- 16 Go de RAM.

Le calcul du SCR vision modèle interne par la méthode des SdS, avec 10 000 scénarios monde réel suivis de 1 000 scénarios risque neutre a nécessité 30 heures de calcul. Le coût lié à l'utilisation de notre méthode se décompose de la manière suivante :

- 1) le coût d'obtention de la base d'apprentissage,
- 2) la phase d'apprentissage et calibrage des modèles,
- 3) l'obtention des scénarios monde réel nécessaires à l'utilisation des modèles,
- 4) le temps lié à la prédiction des BE par les modèles.

Nous allons débiter par l'analyse des points 1) et 3) qui sont indépendants des modèles de *machine learning* retenus. Nous avons vu que le coût lié à la construction de la base d'apprentissage est très faible (environ une heure). La construction d'une base de 10 000 environnements économiques (correspondant au point numéro 3) nécessite 2 heures de calcul.

La phase d'apprentissage des modèles a un coût compris entre 1 minute pour XGBoost et 10 minutes pour le SVM. Le coût lié au calibrage par validation croisée est difficile à mesurer : nous avons en effet testé différents jeux de paramètres pour nos différents modèles et nous pourrions continuer une telle opération presque indéfiniment. Dans notre cas, ce coût est estimé à environ 2 à 5 heures (en fonction du modèle retenu : 5 heures pour SVM, 2 à 3 heures pour XGBoost et RF). Enfin, la prédiction des 10 000 provisions *Best Estimate* a un coût négligeable (de l'ordre de quelques secondes).

Nous obtenons ainsi les résultats suivants :

- temps de calcul SdS : environ 30 heures,
- temps de calcul pour la méthode alternative : entre 6 et 11 heures.

Ces résultats peuvent sembler décevants. Cependant, l'intérêt de la méthode réside notamment dans le fait que nous sommes à présent capable d'estimer le ratio de solvabilité initial et la distribution des Fonds Propres à horizon un an via l'utilisation du modèle XGBoost pour différentes allocations d'actifs.

Si nous souhaitons obtenir une telle information pour 21 allocations d'actifs entre 20% et 40% actions, nous avons les résultats suivants :

- temps de calcul SdS : 30 heures \times 21 = 630 heures,
- temps de calcul pour la méthode alternative avec XGBoost : 46 heures (1 heure pour la création de la base d'apprentissage, 3 heures de calibrage et apprentissage, auquel s'ajoutent 2 heures pour la génération des scénarios monde réel pour chaque allocation d'actifs testée).

Notons enfin pour conclure, que si nous devons utiliser davantage de scénarios risque neutre pour obtenir une provision *Best Estimate* stable, le coût lié à l'utilisation des SdS va augmenter considéra-

blement, là où celui de notre méthode sera davantage stable.

Après avoir étudié en détail les performances de nos modèles, nous pouvons conclure :

- que notre méthode s'avère adaptée au calcul d'un SCR vision modèle interne. Nous obtenons une prédiction du SCR robuste via l'utilisation des modèles SVM et XGBoost,
- la réduction du temps de calcul par la méthode permet d'en justifier l'utilisation,
- la méthode peut être utilisée pour se substituer au calcul de SCR pour différentes allocations d'actifs entre 20% et 40% action. Mais certains modèles sont plus adaptés à certaines allocations d'actifs.

Nous pensons que XGBoost est le modèle le plus approprié pour répondre à notre problématique. C'est en effet le modèle qui obtient systématiquement la MSE la plus faible et qui est le plus rapide. De plus nous avons vu que la prédiction du SCR obtenue par ce modèle est robuste et plutôt stable quelle que soit l'allocation d'actifs testée. Pour autant, il nous a semblé intéressant de conserver les autres modèles pour l'étude de la projection du ratio de solvabilité à horizon un an. Nous pouvons d'ores et déjà nous demander si la meilleure performance du SVM au niveau de la queue gauche de la distribution des Fonds Propres de l'assureur à horizon un an va amener ce modèle à sur-performer XGBoost lors de la projection du ratio de solvabilité de l'assureur à horizon un an.

4.3 Utilisation de la méthode pour projeter le ratio de solvabilité à horizon un an

4.3.1 Calibrage des modèles

Nous utilisons la base d'apprentissage $Base_{app}^{t=2}$ telle que présentée au chapitre précédent. Elle contient 21 675 lignes. Nous effectuons le même retraitement de données que celui effectué pour la base d'apprentissage $Base_{app}^{t=1}$ pour les modèles RF et XGBoost. Une fois ce re-traitement effectué avec l'ajout de la variable « CF », nous nous intéressons à l'importance relative des variables explicatives des modèles RF et XGB :

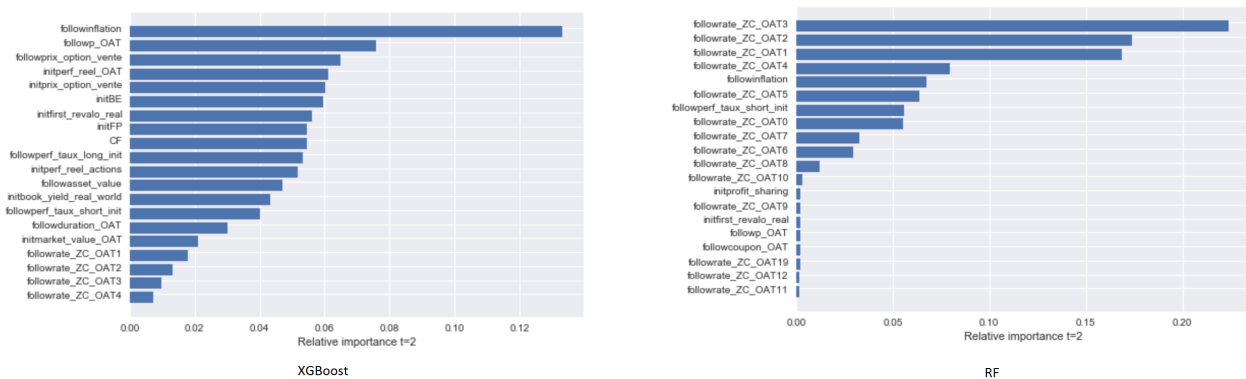


FIGURE 4.13 – Importance relative des variables explicatives pour les modèles XGBoost et RF

Ce premier résultat est surprenant au regard de ce qu'on obtenait en date $t=1$. Le modèle XGBoost semble utiliser davantage de variables explicatives pour estimer la provision *Best Estimate* à horizon deux ans. Ce résultat semble cohérent, puisque l'on s'attend à ce que le nombre de variables explicatives pertinentes soit plus élevé à horizon deux ans. En revanche RF se concentre presque exclusivement sur les premières maturités de la courbe des taux futures et l'inflation future. Dès lors, nous pouvons redouter une baisse de la performance du RF.

4.3.2 Comparaison des modèles

Nous allons maintenant nous intéresser à la comparaison des modèles sur la base de test $Base_{test-1}^{t=2}$ telle que nous l'avons présentée au chapitre 3. Cette base contient 10 200 données. Nous effectuons le même re-traitement que précédemment sur cette base pour RF et XGB.

L'étude des performances des modèles sera plus succincte que celle effectuée à la section précédente, puisque nous avons déjà mis en lumière certaines caractéristiques des différents modèles en terme de performance et que nombre de ces résultats restent identiques ici. De plus, à horizon un an, nous avons une base de test $Base_{test-1}^{t=1}$ de 5 100 couples entrée/sortie alors que nous souhaitons utiliser ensuite la méthode sur 10 000 couples dans le cadre d'un calcul de SCR. À horizon deux ans, nous avons une base de test $Base_{test-1}^{t=2}$ de 10 200 couples entrée/sortie. Or, l'estimation de la distribution du ratio de solvabilité à horizon un an nécessitera l'estimation de 1 000 000 de provisions *Best Estimate*. Dans ces conditions, il semble moins pertinent de s'intéresser à l'erreur commise sur seulement 10 200 couples.

De la même manière que précédemment, nous estimons l'erreur en pourcentage et la MSE des différents modèles, nous obtenons les résultats suivants :

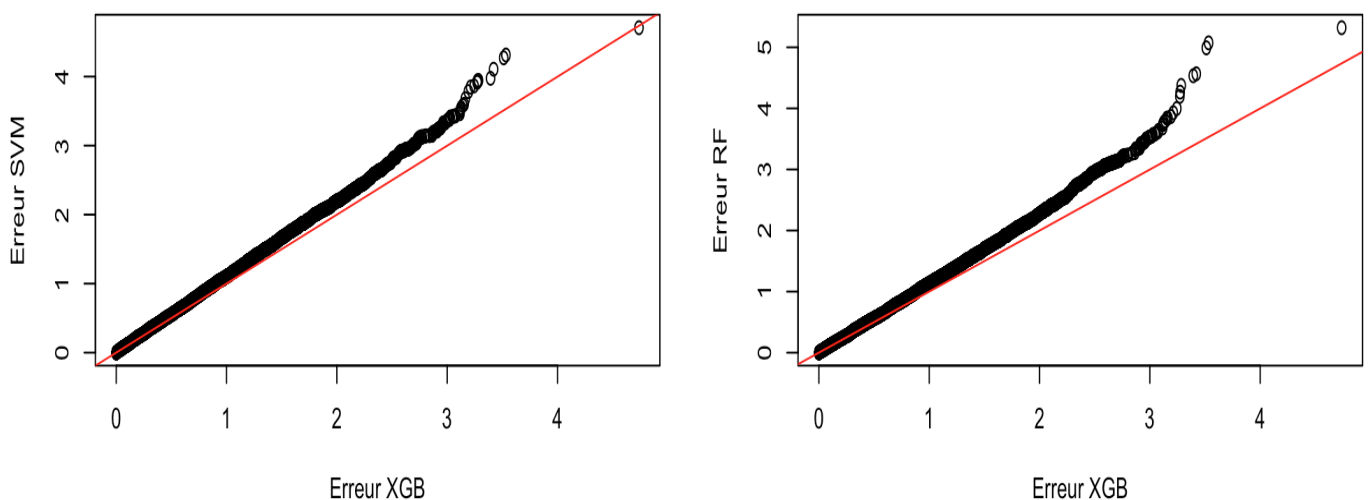
Modèle	XGBoost	RF	SVM
MSE	0,85	1,10	1,04
$max(Erreur\%)$	4,74	5,32	4,70
$Var_{99\%}(Erreur\%)$	2,61	3,07	2,92
$Var_{90\%}(Erreur\%)$	1,67	1,88	1,85

TABLE 4.8 – Répartition des erreurs entre les différents modèles

Les erreurs obtenues par les différents modèles semblent plus conséquentes à horizon deux ans. On note une détérioration assez forte de la MSE pour le RF, alors que la MSE obtenue par XGB et SVM est relativement stable. L'erreur maximale commise par les modèles SVM et XGB augmente en revanche de façon importante (environ 1 point de plus ici).

XGBoost reste le modèle qui obtient la meilleure MSE et les meilleurs quantiles des erreurs sur cette base. On note une erreur maximale légèrement supérieure à celle commise par le SVM comme nous l'avons noté à horizon un an.

Mais cette meilleure répartition des erreurs de la part du modèle XGBoost s'avère d'autant plus parlante en traçant un qq-plot entre les distributions des erreurs obtenues par les différents modèles :



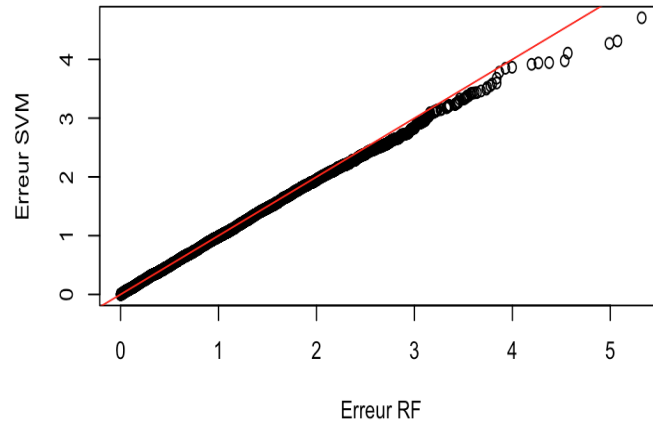


FIGURE 4.14 – qq-plot entre l'erreur commise par les différents modèles

Les résultats sont ici bien plus flagrants que ceux obtenus à horizon un an. On note une très nette sur-performance du modèle XGBoost sur les autres modèles. Toutefois, la densité reconstituée des provisions *Best Estimate* à horizon deux ans, nous montre que les différents modèles restent relativement proches de la distribution théorique :

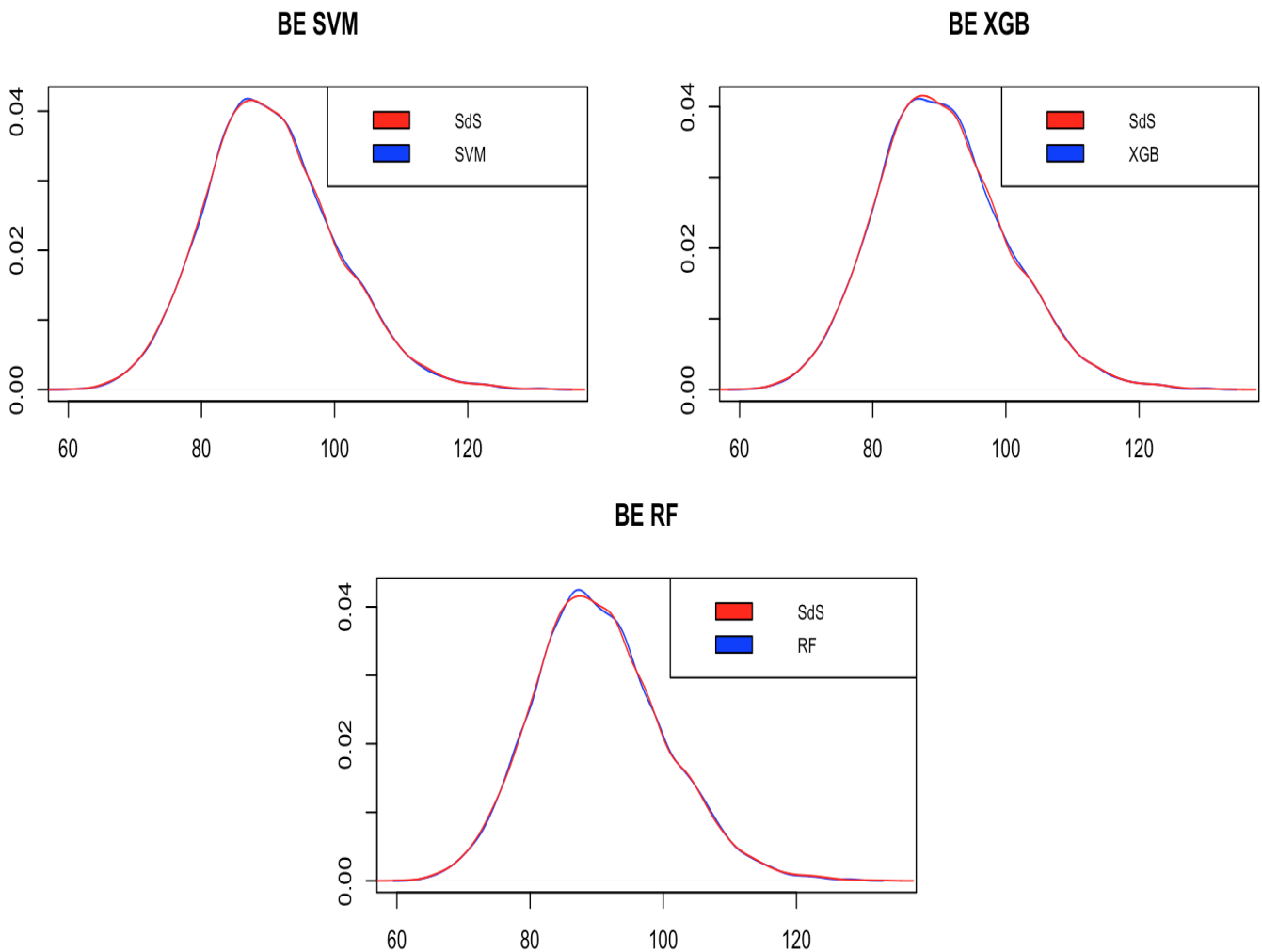


FIGURE 4.15 – Comparaison des densités des BE obtenus à horizon 2 ans par les modèles à la densité théorique

4.3.3 Étude préliminaire sur le bilan de l'assureur et rappels de la méthode

Nous allons maintenant chercher à projeter le ratio de solvabilité à horizon un an. Nous utiliserons alors la méthode des SdSdS comme référence, afin de les comparer à ceux obtenus par notre méthode avec l'utilisation des différents modèles. Nous débiterons toutefois par une analyse sommaire des résultats obtenus par les SdSdS pour le bilan de l'assureur retraite. Nous allons notamment nous intéresser à la progression de son SCR et de son ratio de solvabilité à horizon un an. Avant de présenter ces résultats, il nous a semblé utile de rappeler les principes des SdSdS ainsi que le cadre d'utilisation de la méthode telle que nous allons l'utiliser dans la partie suivante.

• Rappel de la méthode utilisée

Nous allons chercher à comparer la performance de la méthode à celle de l'utilisation des SdSdS pour les différents modèles. Pour ce faire, nous estimons la distribution du ratio de solvabilité de l'assureur à horizon un an par la méthode des SdSdS avec l'utilisation de 1 000 scénarios RN pour chaque calcul de BE et 1 000 scénarios MR pour chaque projection de l'univers économique dans le futur. Nous obtenons alors 1 000 univers économiques et bilans prudentiels à horizon un an. Pour chaque bilan prudentiel à horizon un an, nous avons également 1 000 SCR associés et nous pouvons en déduire 1 000 ratio de solvabilité à horizon un an et donc une distribution du ratio de solvabilité futur. La méthode s'applique lors du calcul du SCR qui nécessite de projeter 1 000 bilans prudentiels à horizon deux ans pour chaque bilan prudentiel en date $t=1$. Notre méthode permet alors d'estimer le BE à horizon deux ans sans effectuer de projections univers risque neutre ni de re-calibrage du GSE à horizon deux ans.

Nous rappelons le principe des SdSdS ci-dessous :

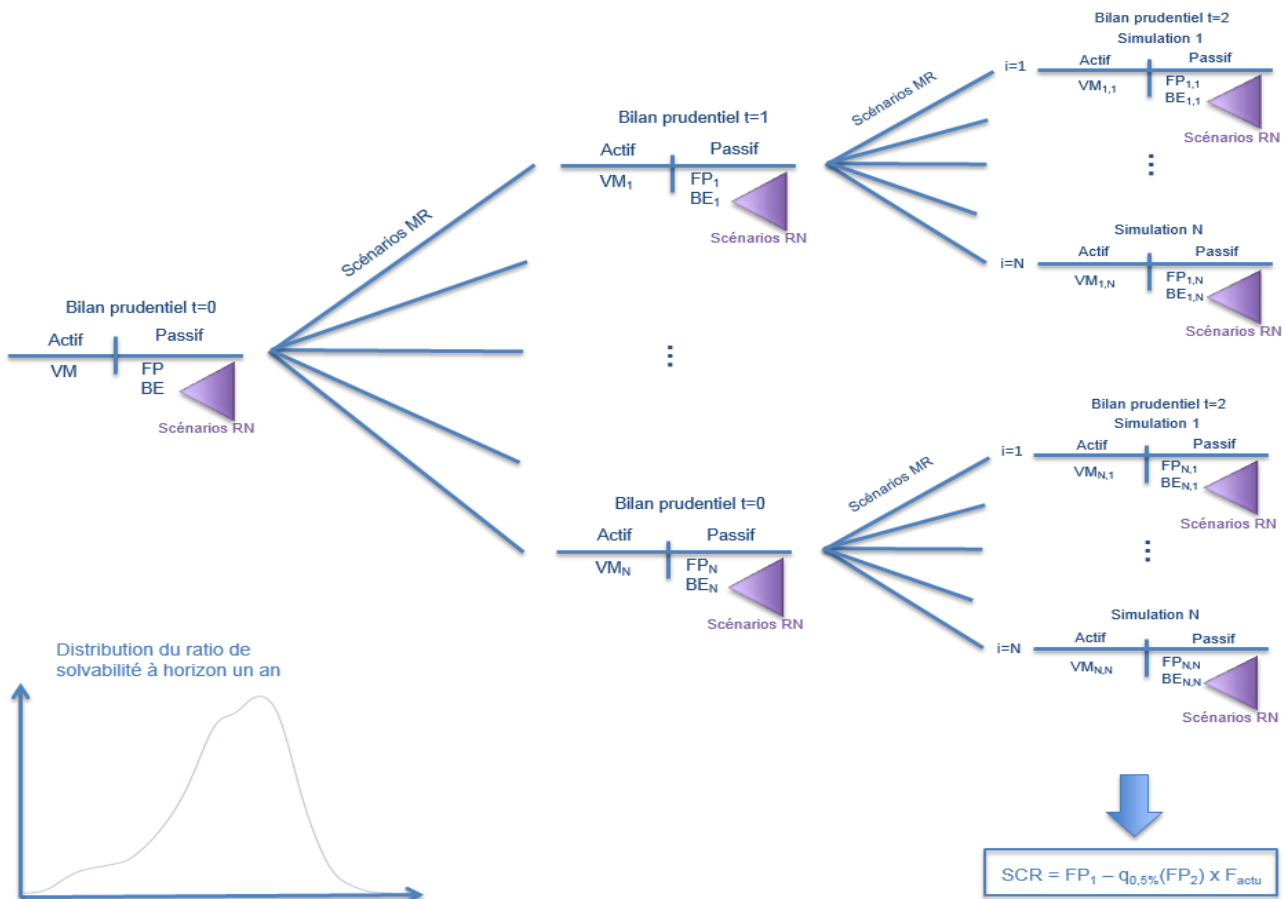


FIGURE 4.16 – La méthode des SdSdS pour l'estimation de la distribution du ratio de solvabilité à horizon un an

Nous avons vu que la méthode des SdS posait de grandes difficultés de part sa complexité. En utilisant 1 000 simulations risque neutre pour les estimations de provisions *Best Estimate*, 1 000 simulations monde réel entre $t=0$ et $t=1$ pour estimer une distribution du ratio de solvabilité, et enfin 1 000 simulations monde réel entre $t=1$ et $t=2$ pour l'estimation du SCR en date $t=1$, nous sommes amenés à générer 1 000 000 000 de scénarios économiques¹.

À notre connaissance face à une telle complexité, une approche de type SdS n'a, à ce jour, jamais été présentée dans un mémoire d'actuariat.

La méthode que nous proposons, permet de contourner cette complexité opérationnelle. Nous parlerons de « SdSdML » lorsque nous utiliserons cette approche : Simulations dans les Simulations suivie de modèle de *Machine Learning* en date $t=2$, pour l'estimation de la provision *Best Estimate*. Nous sommes alors ramené à la génération de $1\,000 \times 1\,000$ scénarios pour l'estimation de la distribution du ratio de solvabilité à horizon un an².

Notre méthode suit alors ce schéma :

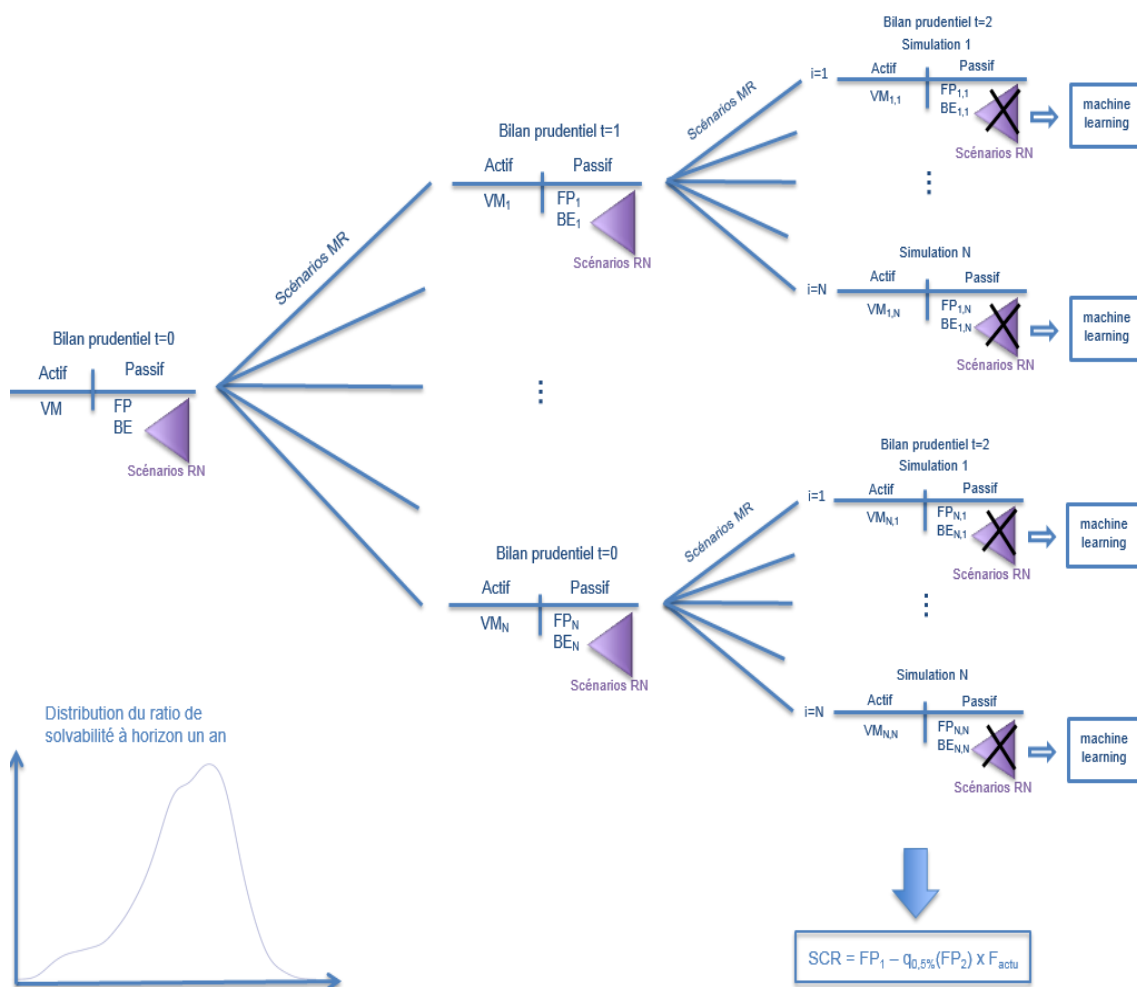


FIGURE 4.17 – La méthode SdSdML

Nous reviendrons sur le gain de temps obtenu via l'utilisation de notre méthode en fin de chapitre. Nous allons à présent nous intéresser aux résultats obtenus par la méthode des SdSdS pour le contrat retraite étudié.

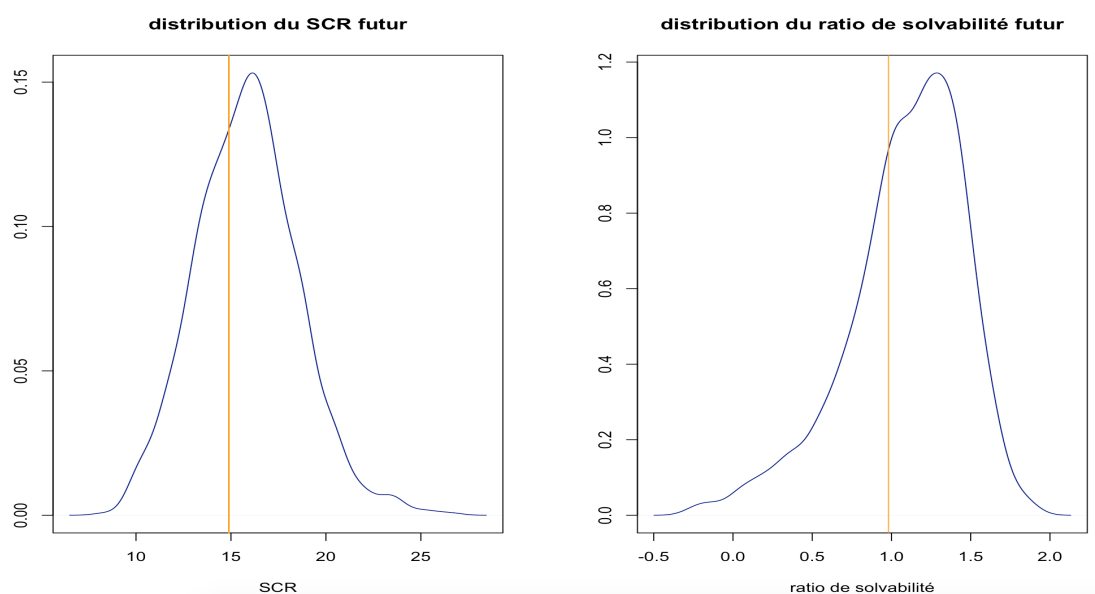
1. Il faudrait également compter le nombre de scénarios liés à l'évaluation des provisions *Best Estimate* en date $t=0$ et $t=1$ que nous négligeons.

2. En négligeant les scénarios liés au calcul des provisions *Best Estimate* à horizon un et deux ans.

- Quelques résultats préliminaires :

Nous allons étudier l'évolution du SCR et du ratio de solvabilité à horizon un an. Il convient tout d'abord de se rappeler des résultats obtenus précédemment concernant la distribution de la valeur de marché de l'actif, des Fonds Propres et de la provision Best Estimate à horizon un an. Nous avons notamment noté une progression des Fonds Propres moyens relativement forte malgré une valeur de marché de l'actif qui augmente peu en moyenne (mais qui est volatile) et une provision *Best Estimate* plus faible en moyenne.

Le SCR au 31/12/2016 était de 14,89M € et nous avons un ratio de solvabilité de 98,12%. Au 31/12/2017, le SCR moyen est de 15,87M € et le ratio de solvabilité moyen de 108,92%. Nous obtenons la distribution suivante (estimée par la méthode des SdSdS) :



Nous traçons en orange le niveau initial du SCR et du ratio de solvabilité. Le SCR moyen augmente ainsi de 5,58% à horizon un an, alors que les Fonds Propres moyens augmentent de 21,83%. On a donc une très forte hausse des Fonds Propres moyens alors que la hausse du SCR moyen est moins flagrante, d'où une hausse du ratio de solvabilité moyen. Dans 65,5% des cas, le ratio de solvabilité futur estimé est supérieur à 100% alors qu'il était inférieur à 100% en date initiale. La situation de l'assureur semble s'améliorer sensiblement à horizon un an. Il reste néanmoins environ 35% des cas où l'assureur obtient un ratio de solvabilité inférieur à 100%.

Enfin, on note que dans 95% des cas le ratio de solvabilité de l'assureur à horizon un an est supérieur à 35,6%. Ces informations sont particulièrement intéressantes pour l'assureur. L'amélioration probable de sa situation à horizon un an peut en effet l'inciter à conserver son allocation d'actif actuelle et chercher d'autres solutions pour améliorer sa solvabilité. Il est également conscient de son risque, via la mesure du quantile à 5% de son ratio de solvabilité à horizon un an. Toutefois, cette mesure de risque se fait a posteriori. Il serait souhaitable de pouvoir définir un niveau de risque souhaité a priori (par exemple obtenir un ratio de solvabilité à horizon un an supérieur à 120% dans 95% des cas) qui permettrait de définir une allocation d'actifs pertinente pour cet objectif. C'est ce que nous souhaitons proposer via l'utilisation de notre méthode.

4.3.4 Application de la méthode à la projection du ratio de solvabilité à horizon un an

Après avoir présenté la méthode et quelques résultats préliminaires pour l'assureur, nous allons comparer les résultats obtenus par nos différents modèles aux résultats théoriques obtenus par la méthode des SdSdS.

- **Performance des modèles en terme d'erreur quadratique moyenne :**

Nous avons dressé 1 000 bilans prudentiels en date $t=1$. Pour chaque bilan, nous avons projeté 1 000 environnements économiques en date $t=2$ et re-dressé les 1 000 bilans prudentiels correspondants. Ce faisant, nous avons calculé la MSE entre les BE obtenus par les différents modèles et les BE théoriques obtenus par le SdSdS. Nous obtenons ainsi 1 000 MSE, chaque MSE correspondant à l'erreur quadratique moyenne commise pour 1 000 calculs de BE en date $t=2$. L'erreur quadratique moyenne globale obtenue sur les 1 000 000 de BE en date $t=2$ est alors la moyenne des 1 000 MSE obtenues. Nous allons la calculer. Nous allons également nous intéresser à la MSE maximale obtenue par les modèles pour 1 000 BE en date $t=2$, ainsi qu'au quantile à 95%³ de la MSE dans les 1000 scénarios.

Nous présentons les résultats obtenus sous forme de tableau :

	XGB	SVM	RF
MSE globale	0,90	0,99	1,05
$q_{95\%}(MSE)$	1,44	1,57	1,72
max(MSE)	3,15	2,37	4,63

TABLE 4.9 – MSE des différents modèles à horizon deux ans

On note une légère augmentation de la MSE par rapport à celle obtenue en $t=1$ par XGBoost notamment. Toutefois la MSE des différents modèles reste acceptable. On note également qu'elle est davantage stable pour le SVM ce qui est probablement lié aux résultats sur les queues de distribution obtenus en date $t=1$. Nous poursuivons par la comparaison des SCR et ratio de solvabilité moyens estimés par les différents modèles, ainsi que leur écart type respectif.

- **Performance des modèles en terme de prédiction du SCR :**

	SdSdS	XGB	SVM	RF
SCR moyen	15,87	15,82	15,63	16,35
min(SCR)	8,3	7,49	7,42	8,03
max(SCR)	26,65	27,09	26,26	28,56
écart type du SCR	2,70	2,78	2,81	2,82

TABLE 4.10 – écart type et moyenne du SCR à horizon un an

XGBoost semble confirmer sa meilleure performance via un SCR moyen plus proche du SCR moyen estimé par les SdSdS. Il obtient également un écart type plus proche de l'écart type théorique. On note également que les différents modèles ont un écart type plus élevé que l'écart type théorique. Le SCR minimal estimé est en effet beaucoup plus faible, de même que le SCR moyen. RF semble sur-estimer

3. Nous retenons le quantile à 95% de la MSE, car elle correspond à la MSE minimale atteinte dans les 50 scénarios les plus défavorables. Considérer un quantile plus élevé reviendrait à étudier un nombre de scénarios trop restreint.

le SCR. Ce résultat est d'autant plus flagrant lorsque l'on trace la distribution des SCR estimés par les différents modèles et qu'on la compare à la distribution théorique :

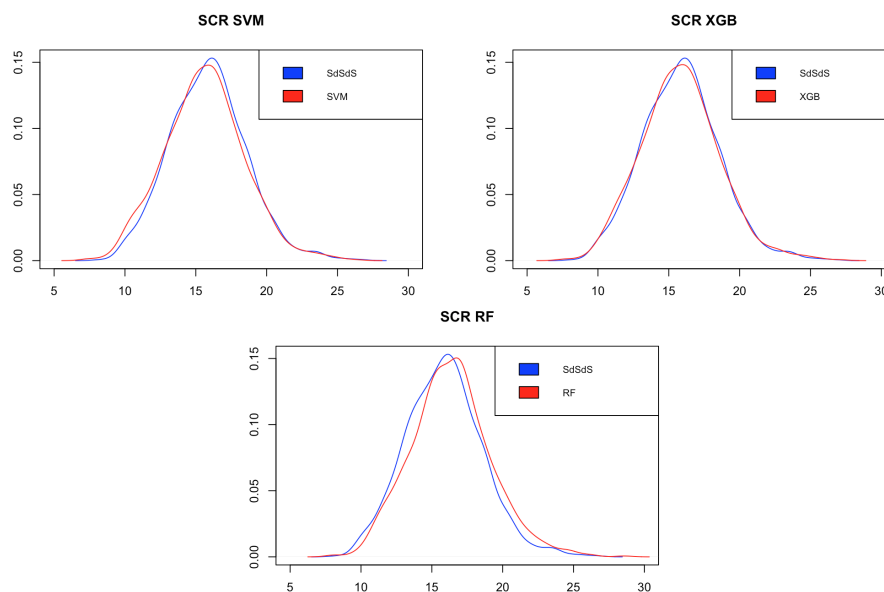


FIGURE 4.19 – Comparaison des distributions des SCR obtenus par les modèles à la distribution théorique

Ces graphiques démontrent une bonne adéquation du modèle XGBoost et une performance relativement correcte du SVM également. Nous poursuivons en étudiant la performance des différents modèles en terme de prédiction du ratio de solvabilité.

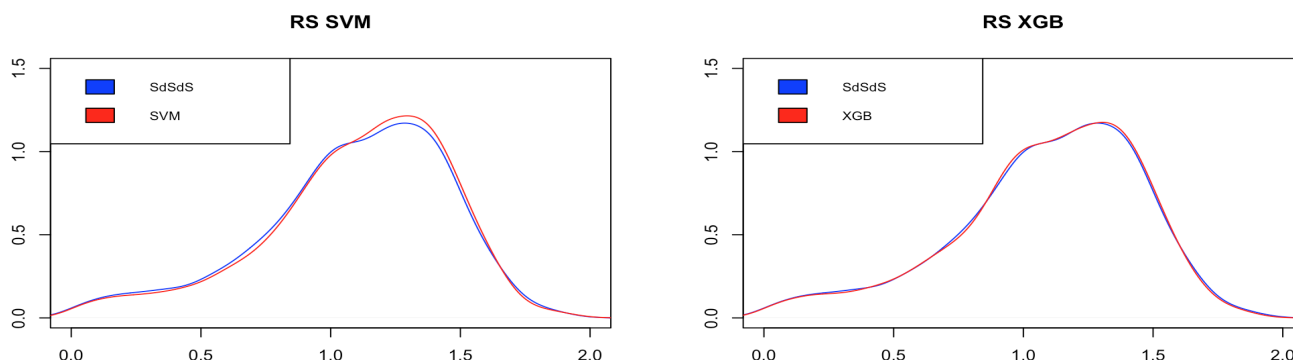
• **Performance des modèles en terme de prédiction du ratio de solvabilité à horizon un an :**

Nous comparons le ratio de solvabilité moyen et l'écart type du ratio de solvabilité à horizon un an estimé par les différents modèles :

	SdSdS	XGB	SVM	RF
RS moyen	108,9%	109,0%	110,2%	105,5%
écart type du RS	0,370	0,367	0,365	0,356

TABLE 4.11 – écart type et moyenne du RS à horizon un an

XGBoost et SVM obtiennent de très bonnes performances en terme de prédiction du ratio de solvabilité à horizon un an. XGBoost est particulièrement proche de la distribution théorique :



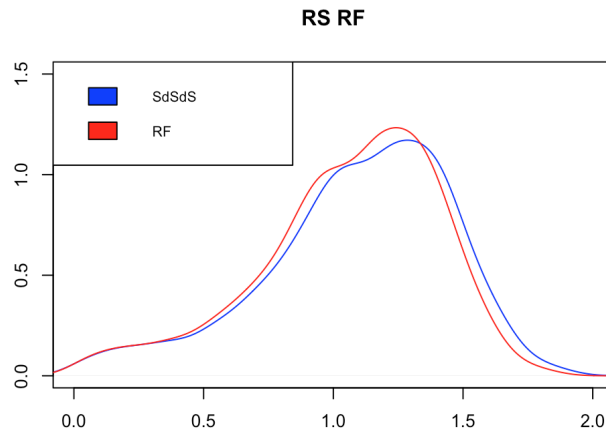


FIGURE 4.20 – Comparaison des distributions des RS obtenus par les modèles à la distribution théorique

Nous avons ainsi proposé une méthode adaptable à différentes problématiques actuarielles. Il semble que XGBoost soit le modèle le plus à même de s'adapter aux différentes situations et reste le plus stable en terme de performances. Nous allons maintenant proposer une application de la méthode à une problématique de choix d'allocations d'actifs.

4.4 Application de la méthode au choix d'une allocation d'actifs

Nous proposons une application concrète de notre méthode permettant à un assureur de se positionner sur un choix d'allocation d'actifs. Nous proposons d'utiliser notre méthode afin de déterminer les couples (rendement moyen du ratio de solvabilité à horizon un an, volatilité du ratio de solvabilité à horizon un an) pour des allocations d'actifs allant de 20% action à 40% action avec un pas de 5% entre chaque allocation testée.

Nous retenons l'utilisation du modèle XGBoost pour cet exercice, puisqu'il s'agit du modèle le plus stable en terme de performance et également du modèle le plus rapide lors de la phase de prédiction du BE.

Nous proposons le process suivant pour déterminer l'allocation d'actifs :

- choix d'un seuil de risque (par exemple ratio de solvabilité supérieur à 120% dans 95% des cas, ou volatilité du ratio),
- calcul du ratio de solvabilité moyen à horizon un an et de sa volatilité pour différentes allocations d'actifs via la méthode SdSdML,
- une fois les calculs effectués, il est alors possible de se positionner sur un choix d'allocation d'actifs en fonction du critère de risque retenu, du rendement obtenu et éventuellement d'autres critères (avis d'expert, etc). Nous pouvons également affiner le choix sur une zone plus restreinte avec un pas plus faible.

Lors de l'application de notre méthode, nous pouvons également comparer l'efficacité de celle-ci par rapport à la méthode des SdSdS⁴. Nous obtenons les résultats suivants pour l'estimation de la distribution du ratio de solvabilité à horizon un an (pour une allocation d'actifs choisie) :

- temps de calcul SdSdS : 75 heures,
- temps de calcul pour la méthode alternative : 19 heures.

4. Notons toutefois, que ces résultats ont été obtenus sur une machine légèrement plus performante que précédemment, équipée d'un processeur Intel Core i7

Ainsi, en estimant 5 allocations d'actifs, notre méthode a nécessité 95 heures de calculs (4 jours) contre 375 heures (15 jours) si l'on choisissait d'utiliser la méthode des SdSdS.

Nous conservons les informations du ratio de solvabilité initial, du ratio de solvabilité moyen à horizon un an, de la performance du ratio à horizon un an, de sa volatilité et du quantile à 5% du ratio à horizon un an. Nous obtenons les résultats suivants obtenus par notre méthode, regroupés dans un tableau :

part action	RS_0	$moy(RS_1)$	$perf_{RS}$	$Vol(RS)$	$VaR_{5\%}(RS)$
20%	137,03%	142,97%	4,35%	34,7%	79,78%
25%	113,68%	121,12%	6,55%	36,2%	56,14%
30%	98,04%	109,0%	11,18%	36,7%	35,60%
35%	85,87%	96,36%	12,22%	37,1%	23,10%
40%	74,74%	84,00%	12,39%	37,9%	9,50%

TABLE 4.12 – Ratio de solvabilité futur pour différentes allocations d'actifs

Afin de bien rendre compte de ces résultats, nous traçons sur un même graphique la performance du ratio de solvabilité à horizon un an en fonction de la volatilité du ratio de solvabilité à horizon un an :

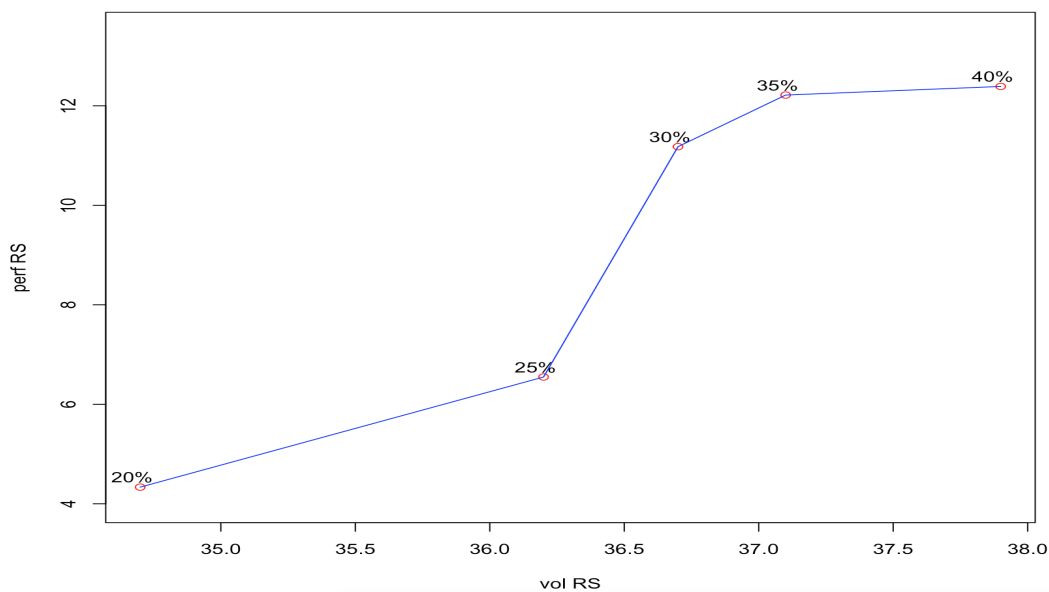


FIGURE 4.21 – Rendement du ratio de solvabilité à horizon un an en fonction de sa volatilité

Ces résultats indiquent comme attendu que la performance moyenne du ratio de solvabilité à horizon un an augmente avec la part action en portefeuille, le risque également, qu'il soit modélisé par la volatilité du ratio futur, ou par la VaR à 5% du ratio dans un an.

Toutefois, ces résultats montrent également qu'il semble inutile d'augmenter de manière trop importante la part action en portefeuille, puisque le gain en terme de rendement est relativement minime pour un risque largement accru. Il semble davantage pertinent de choisir une allocation d'actifs comprenant entre 25% et 30% d'actions. Il pourrait alors être judicieux d'affiner les calculs dans cette zone.

Notons également que le choix de l'allocation d'actifs impacte fortement le ratio de solvabilité initial. Nous nous sommes placés dans un cadre théorique, de façon à nous intéresser à des allocations d'actifs utilisées par des assureurs retraite. Cependant, un assureur est tenu d'avoir des Fonds Propres suffisants pour couvrir son SCR. Nous conseillerions alors à l'assureur que nous avons étudié d'envisager d'augmenter ce ratio de solvabilité initial. Au vu des performances moyennes attendues à horizon un an

et du faible impact sur la volatilité attendue du ratio de solvabilité lorsque la part action diminue légèrement, nous conseillons de privilégier une hausse de la solvabilité initiale par d'autres moyens qu'une forte modification de l'allocation d'actifs en portefeuille via la mise en place de réassurance par exemple.

Tout au long de ce chapitre, nous avons étudié la mise en place de notre méthode en pratique. Nous avons étudié sa capacité à se substituer aux SdS et à prédire un SCR vision modèle interne, puis à projeter le ratio de solvabilité à horizon un an. Nous avons montré que notre méthode avec le modèle XGBoost obtenait des résultats robustes et stables pour différentes allocations d'actifs et présentait un gain de temps conséquent par rapport à l'utilisation des SdS. Nous avons également proposé une utilisation concrète de la méthode pour un assureur retraite. Nous allons à présent conclure cette étude, en exposer ses limites et proposer des pistes d'améliorations.

Conclusion

Dans ce mémoire, nous nous sommes intéressés au ratio de solvabilité et à sa distribution à horizon un an. Nous avons vu que le ratio de solvabilité permet de prendre en compte les interactions actif-passif, qui sont fortement présentes en assurance vie, mais également de tenir compte de l'impact d'une allocation d'actifs sur le SCR. Sa distribution dans le futur est également demandée lors d'un exercice ORSA et apporte des informations quant au niveau de risque de l'assureur. Nous avons alors proposé d'utiliser la distribution du ratio de solvabilité futur pour aiguiller un choix d'allocation d'actifs.

Nous avons présenté la méthode des Simulations dans les Simulations, souvent utilisée pour estimer la distribution du ratio de solvabilité futur. Nous nous sommes attardés sur la problématique du temps de calcul de cette méthode et en avons présenté quelques alternatives. Nous avons remarqué que ces alternatives étaient généralement restreintes à un cadre spécifique d'utilisation : calcul d'un SCR vision modèle interne ou projection du ratio de solvabilité. Nous avons souhaité nous affranchir de ces contraintes et proposer une méthode générale, pouvant se substituer aux SdS pour ces différentes problématiques.

Nous avons alors proposé de mettre en place des modèles de *machine learning* visant à apprendre la provision *Best Estimate* future d'un assureur retraite en fonction de son allocation d'actifs et de l'environnement économique. Nous avons étudié trois modèles d'apprentissage supervisé : SVM, XGBoost et RF.

Nous avons tout d'abord confronté notre méthode à un calcul de SCR vision modèle interne. Nous avons montré que XGBoost permettait une estimation robuste du SCR pour différentes allocations d'actifs. Les autres modèles obtiennent une prédiction du SCR moins précise pour certaines allocations d'actifs. Nous avons vus que le SVM semble davantage performant lorsque la part action en portefeuille augmente et que le RF peut être intéressant sur des tranches actions plus faibles. Il serait alors intéressant de proposer un modèle agrégatif afin d'augmenter la précision de la méthode lors de la prédiction du SCR. Nous pouvons également envisager de modifier la base d'apprentissage de façon à ce qu'elle contienne davantage de scénarios « extrêmes », afin d'obtenir une distribution des quantiles des Fonds Propres futurs plus robuste et donc une meilleure estimation du SCR.

Nous avons ensuite comparé la distribution du ratio de solvabilité à horizon un an obtenue par notre méthode à la distribution de référence obtenue via des SdSdS. XGBoost s'avère également le plus adapté à cet exercice et obtient une estimation de la distribution du ratio de solvabilité très proche de la distribution de référence. Nous avons donc retenu ce modèle pour estimer la distribution du ratio de solvabilité futur pour différentes allocations d'actifs.

Nous pouvons alors aiguiller le choix d'une allocation d'actifs en fonction d'un grand nombre de critères : ratio de solvabilité initial, performance moyenne du ratio de solvabilité à horizon un an, volatilité du ratio de solvabilité futur, VaR à 5% du ratio de solvabilité dans un an, etc. Notre méthode permet ainsi de définir a priori un seuil de risque à ne pas dépasser (par exemple : obtenir un ratio de solvabilité futur supérieur à 120%, dans 95% des cas), et d'en déduire une allocation d'actifs « optimale » en fonction de cette contrainte.

Dans le cadre de notre étude, nous remarquons qu'il semble pertinent d'utiliser une allocation d'actifs avec une part action relativement élevée, de l'ordre de 30%, mais qu'il est contre productif d'augmenter davantage la part action en portefeuille. Il peut alors être judicieux de mettre en place une politique de réassurance afin de s'approcher de cette allocation d'actifs, sans impacter trop fortement la solvabilité. Enfin, nous devons garder à l'esprit que les résultats obtenus sont dépendants d'une mesure de risque choisie, qu'il serait judicieux de compléter avec d'autres critères ou avis d'experts pour avoir une vision plus complète.

Liste des abréviations, des sigles et des symboles

BE	Meilleur estimateur (<i>Best Estimate</i> en anglais)
Càdlàg	Continue à droite limitée à gauche
EIOPA	Autorité Européenne des Assurances et des Pensions Professionnelles (<i>European Insurance and Occupational Pensions Authority</i> en anglais)
EQM	Erreur Quadratique Moyenne
FP	Fonds Propres
FV	Juste valeur (<i>Fair Value</i> en anglais)
GBR	<i>Gradian Boosting Regressor</i>
GSE	Générateur de Scénarios Économiques
IFRS	Normes internationales d'information financière (<i>International Financial Reporting Standards</i> en anglais)
LKOCV	<i>Leave k Out Cross Validation</i>
LOOCV	<i>Leave One Out Cross-Validation</i>
MR	Monde Réel
MSE	<i>Mean Squared Error</i>
NS	<i>Nested Simulations</i>
OAT	Obligation Assimilable du Trésor
ORSA	<i>Own Risk and Solvency Assessment</i>
PB	Participation aux bénéfices
PT	Provisions Techniques
RF	Forêt aléatoire (<i>Random Forest</i> en anglais)
RM	Marge de Risque (<i>Risk Margin</i> en anglais)
RMSE	<i>Root Mean Squared Error</i>
RN	Risque Neutre
RS	Ratio de Solvabilité
SCR	Capital de solvabilité requis (<i>Solvency Capital Requirement</i> en anglais)
SdS	Simulations dans les Simulations
SdSdML	Simulations dans les Simulations suivie de l'utilisation d'un modèle de machine learning pour estimer le BE
SdSdS	Simulations dans les Simulations dans les Simulations

SII Solvabilité II

SiSiML *Simulations in Simulations in Machine Learning*

SiSiS *Simulations in Simulations in Simulations*

SR *Solvency Ratio*

SVM *Support Vector Machine*

TMG Taux Minimum Garanti

UFR Taux ultime (*Ultimate Forward Rate* en anglais)

VaR *Value at Risk*

VCAV Indice de Volatilité du CAC40

XGB, XGBoost *eXtreme Gradient Boosting*

Annexes

Annexe A : Les formules analytiques de Vasicek à deux facteurs

La formule analytique permettant de trouver le prix des zéros coupons de maturité T à la date t est :

$$P(t, T) = \exp(A(T - t) - B_1(T - t) \times r_t - B_2(T - t) \times m_t)$$

avec :

$$\begin{aligned} - B_1(T - t) &= \frac{1 - e^{-\alpha_1(T-t)}}{\alpha_1} \\ - B_2(T - t) &= \frac{\alpha_1}{\alpha_1 - \alpha_2} \left(\frac{1 - e^{-\alpha_2(T-t)}}{\alpha_2} - \frac{1 - e^{-\alpha_1(T-t)}}{\alpha_1} \right) \\ - a &= \mu - \frac{\sigma_1^2}{2\alpha_2^2} \\ - b &= \frac{\sigma_1^2 B_1(T-t)^2}{4\alpha_1} \\ - c &= \frac{T-t}{\alpha_2^2} \\ - d &= 2 \frac{B_1(T-t) + B_2(T-t)}{\alpha_2^2} \\ - e &= \frac{1 - e^{-2\alpha_1(T-t)}}{2\alpha_1(\alpha_1 - \alpha_2)^2} \\ - f &= \frac{2\alpha_1(1 - e^{-(\alpha_1 + \alpha_2)(T-t)})}{\alpha_2(\alpha_1 - \alpha_2)^2(\alpha_1 + \alpha_2)} \\ - g &= \frac{\alpha_1^2(1 - e^{-2\alpha_1(T-t)})}{2\alpha_2^3(\alpha_1 - \alpha_2)^2} \\ - A(T - t) &= a(B_1(T - t) - (T - t)) + \mu B_2(T - t) - b + \frac{\sigma_2^2}{2}(c - d + e - f + g) \quad (\text{E}) \end{aligned}$$

La courbe des taux pour chaque scénario est déterminée par (E).

Annexe B : Le qq-plot

Nous proposons un exemple numérique simple afin de bien visualiser l'impact d'une distribution sur un qq-plot. Nous construisons un vecteur X de 10 000 données suivant une loi normale centrée réduite. Nous construisons à partir de X les vecteurs Y , Z et L de la manière suivante :

- $Y=X$. On remplace les 100 premières données de Y par le minimum de X auquel on retranche une loi normale de moyenne 1 et écart type 2. Y a donc une queue de répartition gauche plus lourde que X .
- $L=X$. On remplace les 100 premières données de L par le maximum de X auquel on ajoute une loi normale de moyenne 1 et écart type 2. L a donc une queue de répartition droite plus lourde que X .
- $Z=X$. On remplace les 100 valeurs les plus élevées et les 100 valeurs les plus faibles de Z par 0. Z a donc des queues de répartition plus légères que X à droite comme à gauche.

Nous traçons les qq-plot de (X,X) , (X,Y) , (X,Z) et (X,L) et obtenons le graphique suivant :

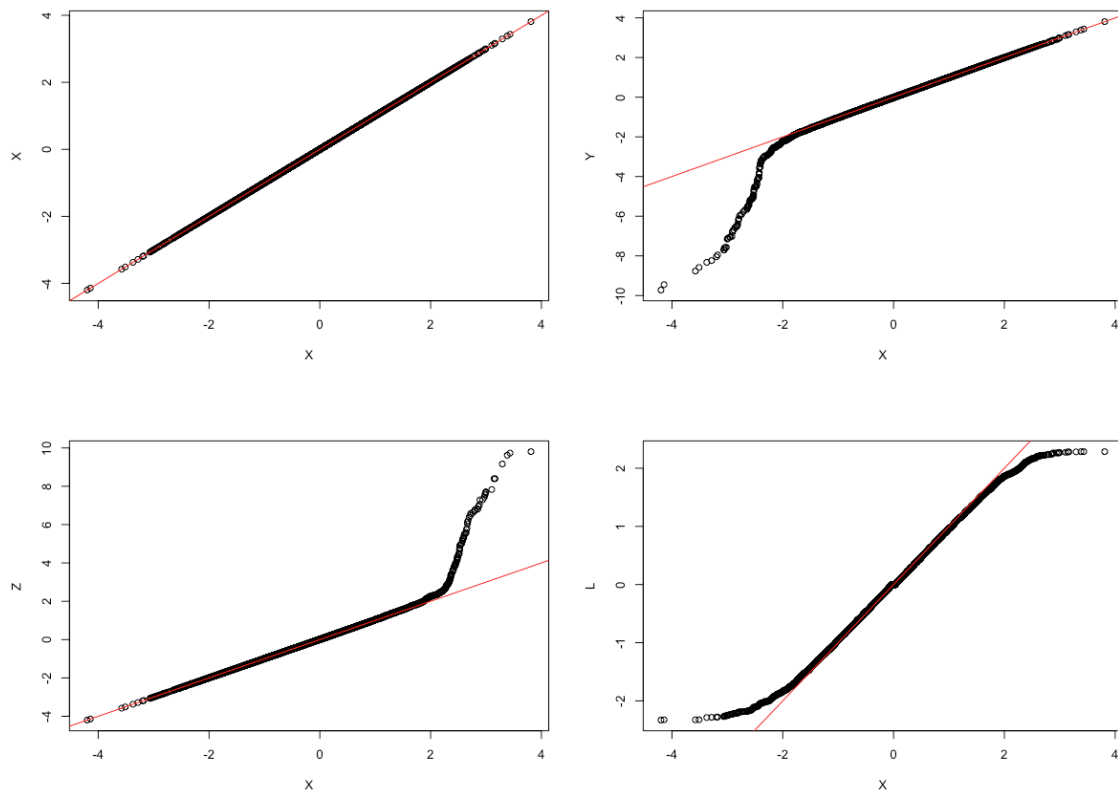


FIGURE 4.22 – Exemple de qq-plot

On remarque que le qq-plot de (X,X) est parfaitement aligné. En haut à droite, il est clair que la queue de distribution gauche de Y est plus lourde que celle de X . De même pour la queue droite de la distribution de Z (en bas à gauche). Enfin, en bas à droite, on observe une des queues de distributions plus légères à droite et à gauche pour L .

Annexe C : qq-plot entre la distribution théorique des Fonds Propres à horizon un an et la distribution obtenue par les modèles de machine learning :

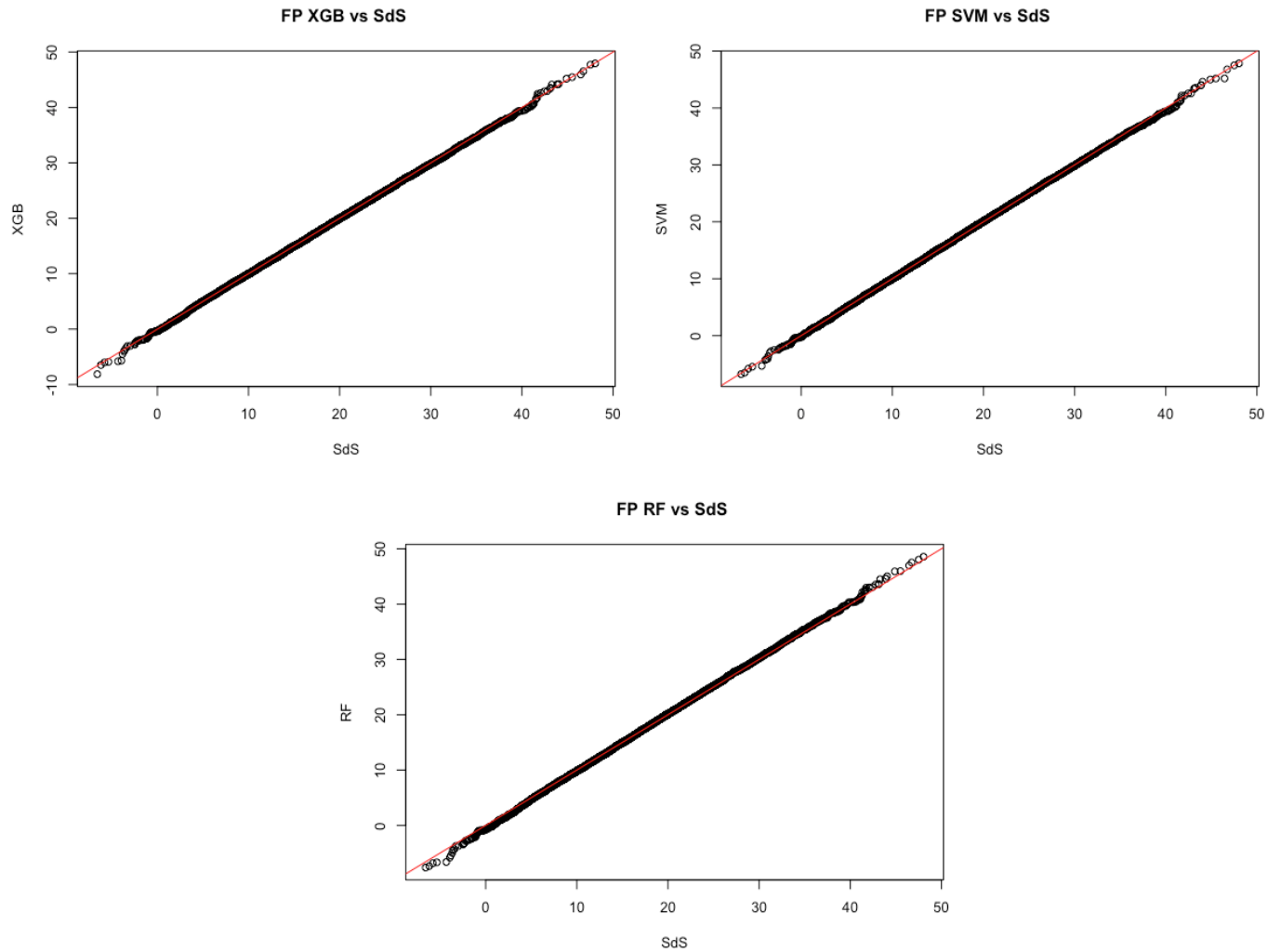


FIGURE 4.23 – qq-plot Fonds Propres modèle vs SdS

Bibliographie

- [1] Q. Bazoge. *Effet du choix des modèles et de leur calibrage sur le bilan prudentiel*. 2016. Mémoire, Institut des Actuaire.
- [2] E. Biernat and M. Lutz. *Data Science : Fondamentaux et études de cas*. Eyrolles, 2015.
- [3] A. Blumer, A. Ehrenfeucht, et al. *Learnability and the Vapnik-Chervonenkis dimension. (Capacité d'apprentissage et dimension de Vapnik-Chervonenkis)*. 1989.
- [4] F. Bonnin, F. Planchet, and M. Juillard. *Best estimate calculations of savings contracts by closed formulas Application to the ORSA*. 2013.
- [5] L. Breiman, J. Friedman, et al. *Classification and Regression Trees*. 1984.
- [6] L. Breiman, J. Friedman, R. Olshen, et al. *Classification and regression trees*. 1984.
- [7] D. Brigo and F. Mercurio. *Interest Rate Models – Theory and Practice*. Springer, 2006.
- [8] A. Buzzi. *Approximation du bilan économique sous Solvabilité II via des méthodes d'apprentissage automatique et application à l'ORSA*. 2017. Mémoire, Institut des Actuaire.
- [9] T. Chen. *Introduction to Boosted Trees*. 2014.
- [10] T. Chen and C. Guestrin. *XGBoost : A scalable Tree Boosting System*. 2016.
- [11] P. Connault. *Présentation du domaine de recherche ; Algorithme CART*. 2008.
- [12] L. Devineau and A. Boumezoued. *Cours d'aggrégation des risques, M2 EURIA*. Milliman, 2016-2017.
- [13] H. Drucker, J-C. Burges, L. Kaufman, et al. *Support Vector Regression Machines*. 1996.
- [14] T. Durand. *Évaluation et optimisation de la rentabilité d'un portefeuille automobile*. 2016. Mémoire, Institut des Actuaire.
- [15] S. Fortmann-Roe. *Understanding the Bias-Variance Tradeoff*. 2012.
- [16] J. Friedman. *Greedy Function Approximation : A Gradient Boosting Machine*. 1999.
- [17] Hull J. *Options, futures et autres actifs dérivés*. Pearson, 2011.
- [18] M. Juillard. *Cours de mesure de risque*. EURIA, 2017.
- [19] A. Koursaris. *A Least Squares Monte Carlo Approach to Liability Proxy Modelling and Capital Calculation*. 2011.
- [20] D. Lamberton and B. Lapeyre. *Introduction au calcul stochastique appliqué à la finance*. Ellipses Marketing, 1992.
- [21] A. Le Tesson. *Evolution de la solvabilité d'un assureur retraite dans le cas d'une forte hausse des taux*. 2014. Mémoire, Institut des Actuaire.
- [22] H. Markowitz. *Portfolio Selection*. 1952.
- [23] A-C. Martial and J-B. Garnier. *Optimisation du calcul du capital économique d'une compagnie d'assurance via la méthode des Simulations dans les Simulations*. 2013. Mémoire, Institut des Actuaire.
- [24] J. Mercer. *Functions of positive and negative type, and their connection with the theory of integral equations*. 1909.
- [25] S. Morrison. *Nested simulation for economic capital*. 2009.

-
- [26] M. Niedzwiedz. *Utilisation des supports vecteurs machines pour l'accélération du calcul du capital économique*. 2013. Mémoire, Institut des Actuares.
- [27] F. Rouchati. *Projection du ratio de solvabilité d'un assureur retraite : les méthodes paramétriques (Curve Fitting et Least Squares Monte Carlo) peuvent-elles se substituer à la méthode des Simulations dans les Simulations ?* 2016. Mémoire, Institut des Actuares.
- [28] A. Roué. Les réseaux bayésiens comme technique d'optimisation numérique dans le cadre de l'orsa, 2015. Mémoire, Institut des Actuares.
- [29] D. Schrage. *Replicating portfolios for insurance liabilities*. 2008.
- [30] E. Tassin, J. Contant, and M. Roujas. Les 3 piliers de solvabilité ii : retour d'expérience. Fixage, 2017.
- [31] V. Vapnik and S. Kotz. 1982.
- [32] F. Vermet. Apprentissage statistique : une approche connexionniste. EURIA, 2015-2016 cours de M1.
-