Next steps

Antoine Ogier

21 juin 2018

Table des matières

Plan

Choses à faire :

- Simuler des jeux de données pour la prédiction.
- Insérer des données manquantes
- Choisir une méthode d'imputation.
- Appliquer cette méthode en séparant les données en un jeu d'entraînement et un d'apprentissage, de différentes manières
- Evaluer l'imputation, d'une part sur la distance aux données de test et d'autre part sur la performance de prédiction
- Répéter l'opération de nombreuses fois pour avoir une bonne estimation de chaque méthode

1 Simulation

1.1 Simulation des covariables

En premier lieu, on simule un tableau de données X qui servira de variable explicative. On utilise un nombre p à définir de colonnes et n de lignes.

1.1.1 Gaussienne multivariée

Option la plus simple, simuler X par tirage d'une loi normale $\mathcal{N}(\mu, \Sigma)$ pour des paramètres choisis. Sans perte de généralité on prend μ nul. Pour Σ, plusieurs possibilités :

- $\Sigma = I_p$ gaussiennes indépendantes
- $\Sigma = (1-\rho)I_p + \rho\mathbb{1}$ où $\mathbb{1}$ est la matrice remplie de 1 : Corrélation identique entre toutes les variables.

- $\Sigma = \begin{pmatrix} 1 & \rho & \dots & \rho & \rho & & & \\ \rho & 1 & \dots & \rho & \rho & & & \\ \vdots & \vdots & \ddots & \vdots & \vdots & & 0 & \\ \rho & \rho & \dots & 1 & \rho & & & \\ \rho & \rho & \dots & \rho & 1 & & & \\ & & & & & 1 & \dots & \rho \\ & & 0 & & & \vdots & \ddots & \vdots \\ & & & & & \rho & \dots & 1 \end{pmatrix}$: deux groupes de variables corrélées entre elles mais indépendantes d'un groupe à l'autre.

1.1.2 Noised low rank matrix with LRsim

Use *LRsim* from package *denoiseR* : a matrix of size $n \times p$ is drawn from a multivariate standard normal then projected on its first k columns and gaussian noise is added :

- $X_i \sim \mathcal{N}(0, 1)$
- $X_i = UDV^T$
- $X = U_k D_k V_k^T + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma^2)$

1.2 Simulation de la réponse

On choisit y la vaiable de réponse

1.2.1 Choix d'une colonne

Une colonne de X est choisie comme variable y.

1.2.2 Modèle de régression

On choisit β un coefficient de régression et $y = X\beta$

1.2.3 Modèle plus complexe que celui de l'imputation

Choisir un modèle qui ne peut pas complètement être capturé par le modèle d'imputation, comme une régression qui prend en compte le carré d'une des variables. Intéressant pour voir si l'imputation par la moyenne n'est pas meilleure quand les prérequis des modèles d'imputation sont violés.

2 Ajout de données manquantes

2.1 MCAR

Enlever des observations complètement au hasard, sur une variable ou sur un ensemble de variables.

2.2 MAR

2.2.1 Seuillage sur une variable différente

Mécanismes de données manquantes MAR

3 Méthodes d'imputation

3.1 Imputation par la moyenne

C'est la méthode la plus simple pour l'imputation. Elle consiste simplement à remplacer les valeurs manquantes par la moyenne observée de la variable en question.

3.2 Algorithme EM pour l'estimation d'une loi normale (méthode de Schafer)

But : trouver les meilleurs paramètres pour une loi normale multivariée correspondant à X. Pour ce faire, on utilise l'algorithme EM, qui améliore itérativement l'estimation du paramètre $\theta = (\mu, \Sigma)$. :

- E-step : l'estimation actuelle de θ est utilisée pour remplacer les valeurs manquantes par leur espérance (conditionnée par les valeurs observées).
- M-step : l'estimation actuelle des données manquantes de X est utilisée pour calculer une nouvelle estimation de θ sur le tableau complété

Une fois que les paramètres sont calculés on peut les utiliser pour imputer les valeurs manquantes, soit en les remplaçant par leur espérance conditionnelle, soir en les simulant selon leur distribution conditionnelle.

Détails mathématiques sur le conditionnement

3.3 Multiple imputation by chained equations

On fait une première imputation X^{imp} par la moyenne.Puis :
For $t = 1 \dots T$:
For $i = 1 \dots p$

— X_i^{imp} the i^{th} column of X^{imp} is chosen as a regression target for all of the other columns. The rows where observations in the original X_i are not missing are used to learn the regression parameters (this is possible because in X_{imp} there are no missing values).
— The values of X_i^{imp} that are missing in X_i are replaced by their predicted values from the regression model. Now that column i of X_{imp} is updated, the next column is picked as target and the operation is reiterated.

Il est intéressant de voir que quand le modèle utilisé est une régression linéaire bayesienne, cette méthode est équivalente à la précédente.

3.4 Imputation par la PCA

Un algorithme EM est utilisé de manière similaire à la première méthode, pour réaliser une décomposition SVD des données (en utilisant comme variables latentes les axes de projection), et en dériver une imputation.

Plus de contenu math ici, trouver un article de ref

4 Validation/Apprentissage

4.1 Principe de la validation croisée

4.1.1 Le problème de l'ERM

Dans le contexte de la prédiction, notre but est d'apprendre un modèle sous la forme $y = f(x, \psi)$, où ψ est un paramètre à apprendre et f est un modèle choisi (cela peut correspondre au choix d'une fonction dans une clase de fonctions donnée).

On peut définir une fonction de coût $L(y, \hat{y})$ qui évalue la précision d'une prédiction. Le but est alors de trouver le paramètre ψ qui minimise le risque $\mathbb{E}(L(y, f(X)))$. Cependant, nous n'avons pas accès à l'espérance réelle du risque. Il faut donc avoir recours à la Minimisation Empirique du Risque (ERM), c'est à dire trouver le paramètre minimisant le risque empirique :

$$R_{emp} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(X_i, \psi))$$

Malheureusement, on voit que si cette méthode est indispensable pour le choix de ψ, le risque empirique calculé ainsi ne nous donne que peu d'indications sur la performance que le modèle aura sur de nouvelles données (il a été optimisé spécifiquement pour un X et un y donnés). Pour tester la généralité du modèle, il faut avoir recours à des données qui n'ont pas été utilisées pour apprendre le modèle : c'est le principe de la validation croisée.

4.1.2 La division des données

La méthode la plus simple et la plus courante pour procéder à une validation croisée et de diviser les données en deux parties, appelées données d'apprentissage et données de validation.

On choisit $n_A < n$ la taille des données d'apprentissage, puis $I_A = (i_1, \ldots, i_{n_A})$ un ensemble d'indices. Cela nous permet de définir X_A composé des lignes $i_1, \ldots, i_{n_A}$ de X et y_A contenant les observations correspondantes. Cela constitue les données d'apprentissage. Les données restantes sont appelées X_V et y_V et constituent les données de validation, de taille $n_V = n - n_A$.

Une fois cette opération faire, l'ERM est réalisée comme décrit précédemment mais seulement sur les données d'apprentissage, alors que les données de validation restent cachées. Cela donne un paramètre estimé $\hat{\psi}$. La performance du modèle est finalement évaluée par :

$$R_V = \frac{1}{n_V} \sum_{i=1, i \notin I_A}^{n} L(y_i, f(X_i, \hat{\psi}))$$

C'est cette valeur qui sert à la sélection de modèle.

4.2 Le cas de l'imputation

4.2.1 L'imputation comme ERM

Lorsque des données sont manquantes le processus d'apprentissage est légèrement différent. En effet, la plupart des modèles de prédiction ne fonctionnent qu'avec des données complètes. C'est pourquoi on procède à une étape d'imputation des données manquantes.

Autrement dit, un premier modèle est appliqué aux données pour modéliser les valeurs des données non observées. On choisit un modèle $X^{complete} = g(X, \phi)$, où $X^{complete}$ représente les données réelles sant valeurs manquantes (la fonction g peut en fait être un réalisation d'une variable aléatoire).

Il s'agit ici d'un contexte non supervisé (il n'y a pas de variable de réponse, comme précédemment y), on évalue donc ϕ grâce à une fonction de coût $R'_{emp}(X, \phi) = L'(g(X, \phi), \phi)$ qui nous permet de choisir un paramètre estimé en minimisant ce coût. La fonction de coût est généralement choisie en fonction de la vraisemblance des données imputées selon un modèle génératif choisi.

Mais est-ce que c'est complètement vrai ? Dans MICE j'ai l'impression qu'on minimise plutôt un proxy qu'une vraie loss sur les données. cf decision trees où on

4.2.2 Combinaison des deux étapes

On peut donc regrouper les étapes d'imputation des données manquantes et de prédiction de y en une seule :

$$h(X, (\psi, \phi)) = f(X^{imp}, \psi) = f(g(X, \phi), \psi)$$

L'intérêt de cette notation est d'expliciter le fait que l'imputation est partie prenante de la minimisation du risque, et que ses paramètres doivent donc être soumis à la validation croisée comme ceux de la prédiction : les données X_A sont utilisées pour estimer $(\hat{\psi}, \hat{\phi})$. On obtient ainsi une prédiction $\hat{y}_V = h(X_V, (\hat{\psi}, \hat{\phi}))$ qui peut être comparée à y_V.

Note : le contexte est néanmoins sensiblement différent du l'ERM classique. En effet, tout le but de l'imputation est de permettre l'utilisation de n'importe quel modèle prédictif une fois que X est imputé. Une manière de voir la situation est la suivante : deux personnes travaillent sur les données de manière successive, l'Imputeur et l'Analyste. L'Imputeur ne sait pas quel modèle l'analyste va choisir d'utiliser, il va donc optimiser l'imputation en utilisant seulement les informations dont il dispose. Une fois les données imputées, elle sont transmise à l'Analyste qui procède à la prédiction.

Une implication essentielle de ce modèle est qu'il est impossible d'optimiser ϕ et ψ de manière jointe : ils sont estimées successivement :

$$\hat{\phi} = \arg\min_{\phi} R'_{emp}(X, \phi)$$
$$\hat{\psi} = \arg\min_{\psi} R_{emp}(y, h(X, (\psi, \hat{\phi})))$$

5 Validation croisée et imputation

5.1 Le problème des implémentations actuelles

Le formalisme développé précédemment permet de comprendre que dans un contexte de prédiction avec données manquantes, il est logique que les deux paramètres $\hat{\psi}_{X_A}$ et $\hat{\phi}_{X_A}$ doivent être estimés **uniquement** en utilisant les données d'apprentissage X_A et y_A. Une fois ces paramètres estimés, la prédiction sur les données de validation est simplement $h(X_V, (\hat{\psi}_{X_A}, \hat{\phi}_{X_A}))$.

Cependant, les implémentations existantes de la plupart méthodes d'imputation présentées secton 3 ne permettent pas de procéder ainsi. En effet, elles fournissent généralement une seule fonction qui prend en entrée des données incomplètes et renvoie les données imputées, sans permettre l'accès au modèle. Avec les notation précédentes, nous avons uniquement accès à $g' : X \mapsto g(X, \hat{\phi}_X)$ où $\hat{\phi}_X$ est le paramètre optimal pour le jeu de données X.

Clarifier tout ça : là je triche un peu parce qu'en fait on optimise séparément les deux paramètres, et que je n'explique pas trop pourquoi en validation on aurait le droit de juste utiliser la loss de prédiction alors qu'on a défini deux modèles successifs avec des loss différentes. C'est gérable mais

5.2 Alternatives

Cela veut dire que, sans changer d'implémentation, deux solutions sont possibles :

— Imputer toutes les données d'un seul coup, avant de les diviser en données d'apprentissages et de validation. C'est à dire, estimer $\hat{\phi}_X$, puis

$$\hat{\psi}_{X_A} = \arg\min_{\psi} R_{emp}(y, h(X, (\psi, \hat{\phi}_X)))$$

et $\hat{y}_V = h(X_V, (\psi_{X_A}, \phi_X))$.
Dans ce cas, le paramètre ϕ est imputé à partir des données de validation. En particulier, les valeurs imputées dans X_A dépendent de X_V ce qui est contraire aux principes de la validation croisée : les données de validation doivent rester cachées pendant toute l'estimation des paramètres. L'ensembe de la sélection de modèle peut se retrouver faussée dans ce cas.

— Diviser les données, puis imputer séparément les deux jeux de données. Le modèle prédictif est ensuite appris comme d'habitude sur les données complétées. Formellement :

$$\begin{aligned}
\hat{\phi}_{X_A} &= \arg\min_{\phi} R'_{emp}(X_A, \phi) \\
\hat{\phi}_{X_V} &= \arg\min_{\phi} R'_{emp}(X_V, \phi) \\
\hat{\psi}_{X_A} &= \arg\min_{\psi} R_{emp}(y, h(X, (\psi, \hat{\phi}_{X_A}))) \\
\hat{y}_V &= h(X_V, (\psi_{X_A}, \phi_{X_V}))
\end{aligned}$$

Le point important des équations précédentes est que nous utilisons un paramètre ϕ_{X_V} pour calculer la prédiction sur X_V, alors que le paramètre ψ_{X_A} a été optimisé pour ϕ_{X_V}. Si ces deux paramètres diffèrent sensiblement, rien ne garantit que la prédiction sera toujours valide : tout le principe de l'ERM est que les données de validation doivent être issues de la même distribution que celles d'apprentissage, car dans le cas contraire le modèle entraîné sur X_A n'a pas de raison d'être valide sur X_V. Si l'imputation est faite avec deux paramètres différents sur les deux jeux données, nous violons cette hypothèse.

5.3 Exemple et analyse

Prenons l'exemple simple d'une imputation par la moyenne. Comparons les trois approches décrites précédemment :

1. Procéder à l'estimation correcte des paramètres : on calcule la moyenne μ_A des variables de X_A. Ensuite, les valeurs manquantes de X_A **et** de X_V sont

remplacées par μ_A. Ensuite, un modèle prédictif est entraîné sur les données X_A complétées et utilisé pour prédire $\hat{y}_V$ à partir de X_V complété.

2. Procéder à l'imputation avant la séparation : on calcule μ la moyenne des variables du tableau entier X. Les valeurs manquantes de X_A et de X_V sont remplacées par μ, et on procède à la prédiction comme ci-dessus. Dans ce cas, il est clair que la moyenne utilisée pour remplir les données dépend de X_V.
3. Procéder à des imputations séparées après la séparation : on calcule μ_A et μ_V les moyennes respectivement des variables de X_A et X_V. Les valeurs manquantes de X_A sont remplacées par μ_A et celles de X_V par μ_V. Ensuite, la prédiction est réalisée comme précédemment.

   Meme si les distributions sous-jacentes de X_A et X_V sont identiques (puisque la division a été faite au hasard), la moyenne calculée sur les deux jeux de données peut différer, particulièrement pour des données de petite taille (n_A ou n_V petit). Si par exemple, un modèle de régression linéraire est estimé sur X_A complété avec μ_A, notons β_A^i le paramètre de régression pour la variable i. Prenons une observation x de X_V pour la quelle la variable i est manquante. Alors la prédiction $x^T\beta_A$ différera d'une distance $\beta_A^i(\mu_V^i - \mu_A^i)$ entre l'imputation faite correctement et celle faite de manière séparée.

Evidemment, dans le cas de la moyenne il est facile de procéder à l'imputation correcte. Mais losque seule une fonction d'imputation 'boîte noire' est disponible, cela est impossible. Reste alors la méthode d'imputation groupée, qui est totalement inacceptable du point de vue de la sélection de modèle, et celle d'imputation séparée qui est acceptable mais ne garantit pas une bonne prédiction.

En particulier, la deuxième méthode devient complètement inutilisable dans un cas particulier qui est pertinent à notre analyse : si nous devons effectuer une prédiction sur une nouvelles observations avec données manquantes, il est impossible de réaliser une imputation séparée sur cette seule ligne (puisque pour les variables ou l'information est manquante, nous ne disposons d'aucune autre observation permettant d'estimer une valeur crédible), et donc d'appliquer le modèle prédictif par la suite. C'est un point important car dans le cadre de la prédiction du choc hémorragique, nous serons amenés à formuler une prédiction pour un patient fraîchement arrivé à l'hôpital. Cette prédiction pour une seule ligne est donc un point essentiel, qui rend l'imputation séparée également invalide.

Enfin, une dernière méthode permet de trouver un compromis possiblement acceptable. Une par une, les lignes de X_V sont accolées aux données de X_A pour créer un jeu de données de taille $n_A + 1$, qui est imputé en utilisant la boîte noire fournie par la méthode d'imputation. Dans ce cas, c'est bien le paramètre d'imputation $\hat{\phi_{X_A}}$ qui est utilisé (quoi que très légèrement perturbé par l'unique ligne ajoutée aux données), cela permet d'obtenir une imputation correcte de X_V

en répétant la procédure ligne par ligne. Si cela n'est pas acceptable de manière théorique (une ligne très différente des données de X_A pourrait en théorie perturber fortement l'estimation de ϕ), on peut considérer que si n_A est suffisamment grand alors l'imputation obtenue sera presque identique à l'imputation correcte. Un grand désavantage de cette méthode est qu'elle est très coûteuse en terme de calcul puisque l'imputation de X_A est répétée n_V fois.

Malgré leurs problèmes théoriques, il serait intéressant d'illustrer les défauts de ces méthodes en pratique. Il semble en effet que dans de nombreux cas l'effet négatif d'une imputation réalisée de manière incorrecte soit assez peu sensible (quand n, n_a et n_v sont grands, tous les paramètres ϕ_X, ϕ_{X_A}, ϕ_{X_V} seront en fait très proches).

6 Imputation multiple

Même principe, mais on agrège les predictions à la fin, règles de Rubin

7 Résultats