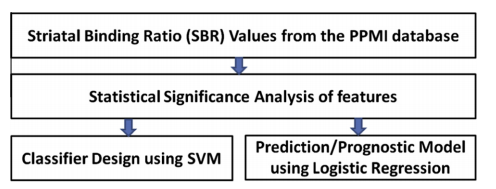
**Diagnostic précoce de la maladie de Parkinson**

Par classification automatique et modèle de prédiction

Les différentes étapes pour le projet :

1. Obtention et explication des données SBR
2. Analyse des caractéristiques
3. Design du classifieur SVM
4. Regression logistique
5. Approche bayésienne



4 classifieurs à faire.

**Obtention et explication des données SBR**

Explication sujet

La difficulté de l’article vient du fait d’avoir une détection précoce de la maladie. Ce qui est intéressant c’est que j’ai plus de données que lors de la rédaction de l’article

Plus le marqueur augmente (ou diminue) plus on a de chance d’avoir la maladie. Différent de plus on est proche de la moyenne et plus on a de chance d’avoir la maladie.

Récupération à partir du labo

Ce qui est intéressant c’est que comparé à l’article, j’ai une base de donnée un peu plus grande : 674 à 736

181 🡺 211 patients normaux

493 🡺 525 patients parkinson

73.15% de PD 🡺 71.33% de PD

Qu’est-ce que SBR ?

**Analyse des caractéristiques**

Calcul de la divergence entre chaque (pour chaque SBR) à l’aide de Fisher

SBR right caudate 0.8157

SBR left caudate 0.8447

SBR Right putamen 1.7557

SBR left putamen 1.8807

Calcul du taux de recouvrement pour chaque classe.

SBR right caudate 27.5815%

SBR left caudate 28.6685%

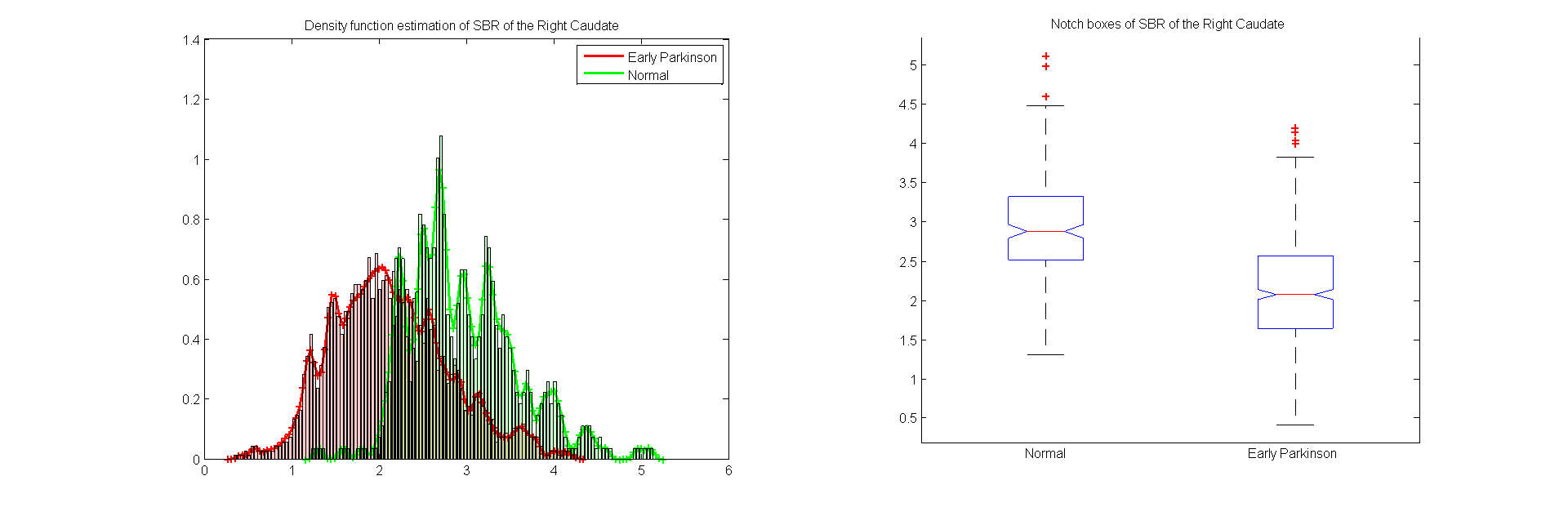
SBR Right putamen 20.7880%

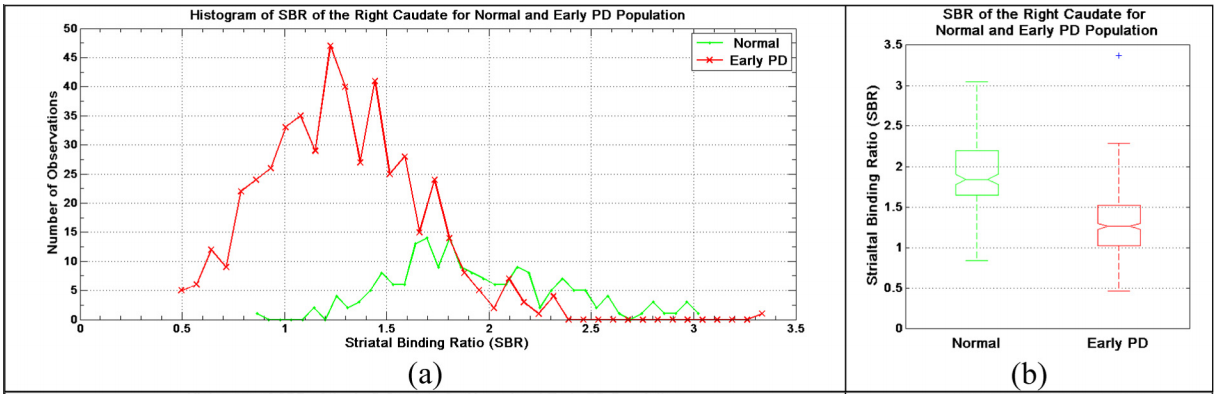
SBR left putamen 21.4674%

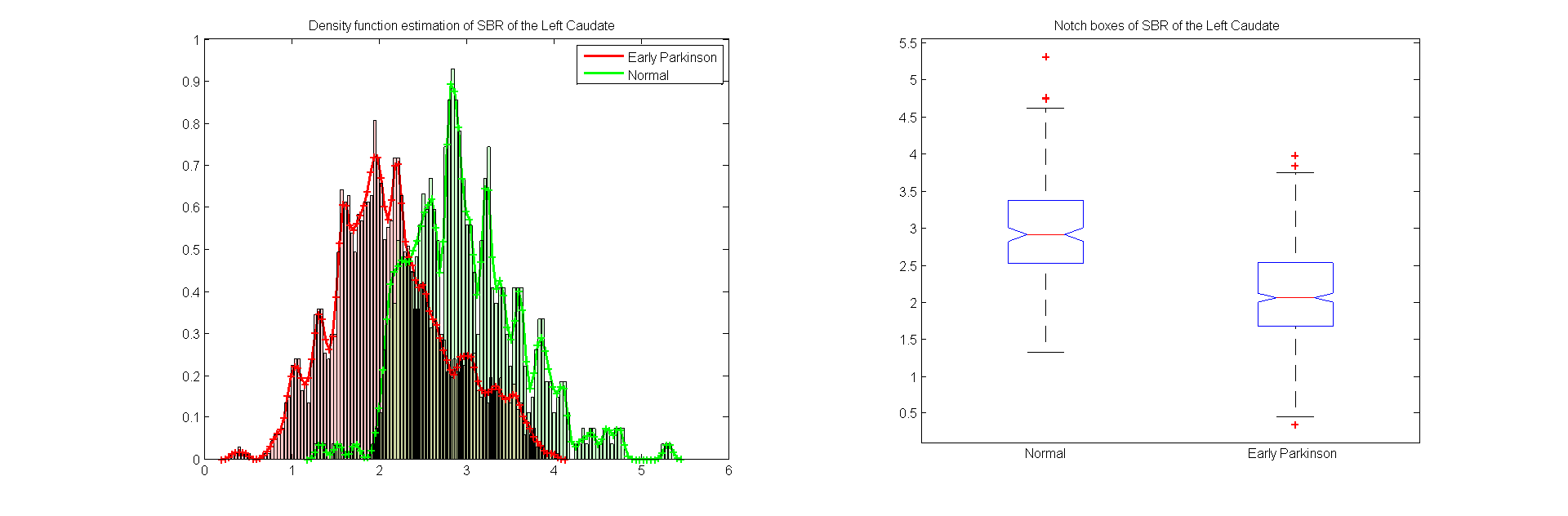
D = 0.8157 ; O = 27.5815% D = 0.8447 ; O = 28.6685%

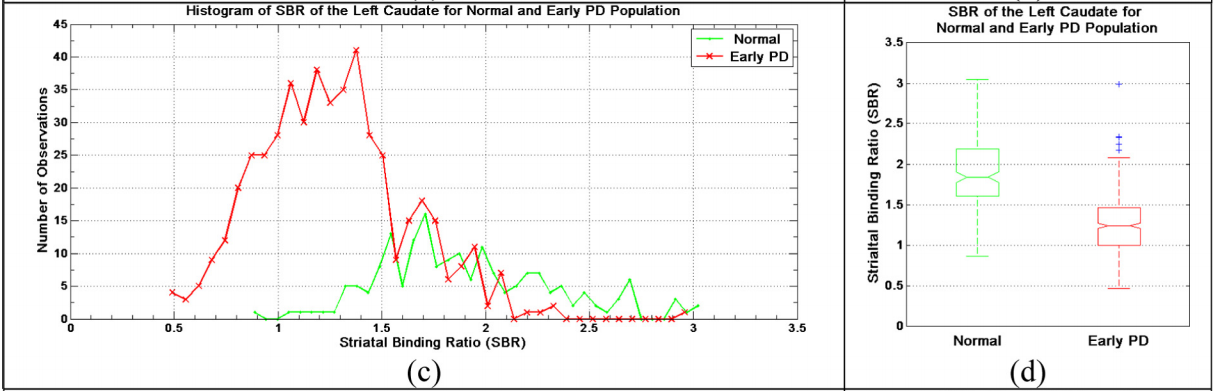
D = 1.7557 ; O = 20.7880% D = 1.8807 ; O = 21.4674%

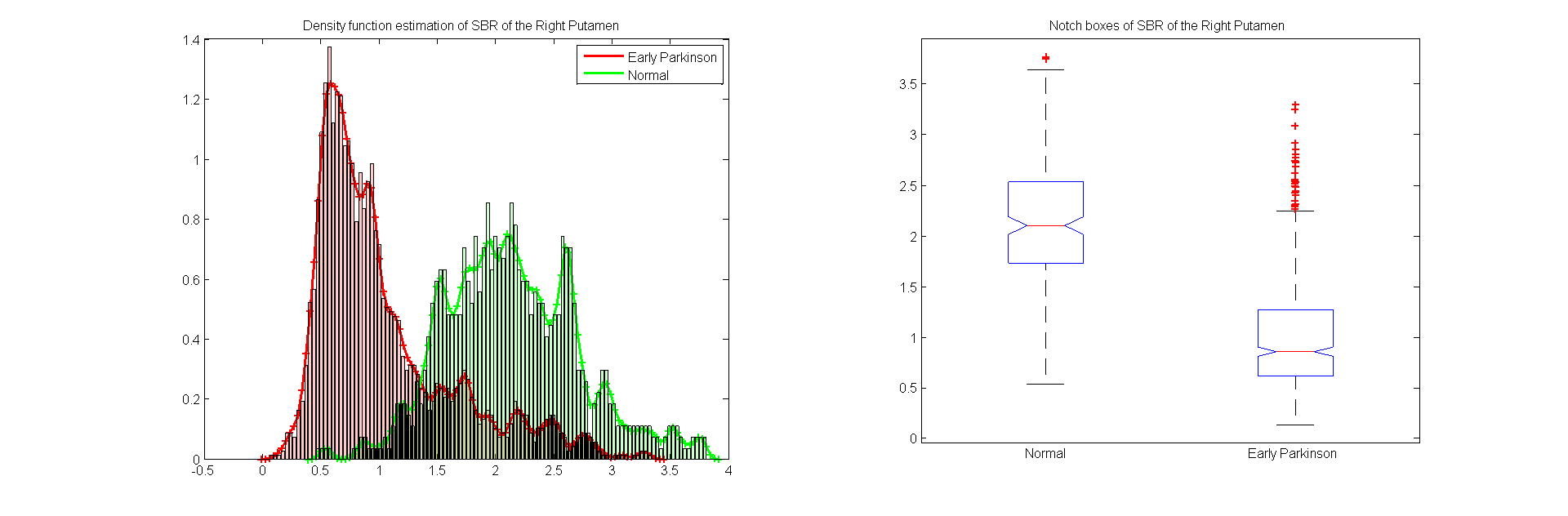
Affichage des données dans boite à moustache et modélisation de la fonction de densité par fenêtres de parzen (noyeau carré et gaussien )

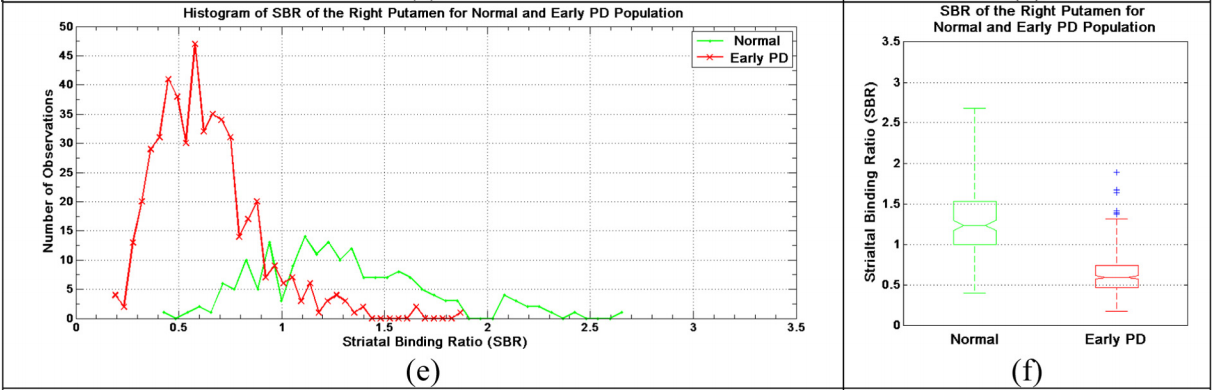


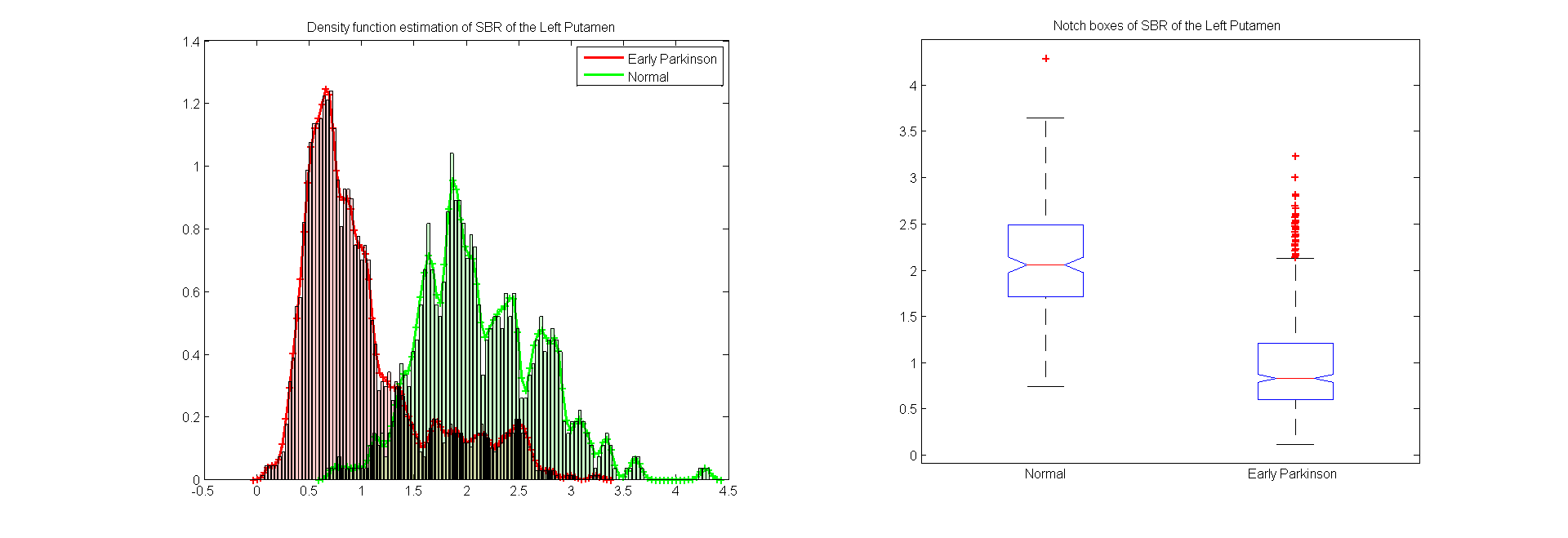


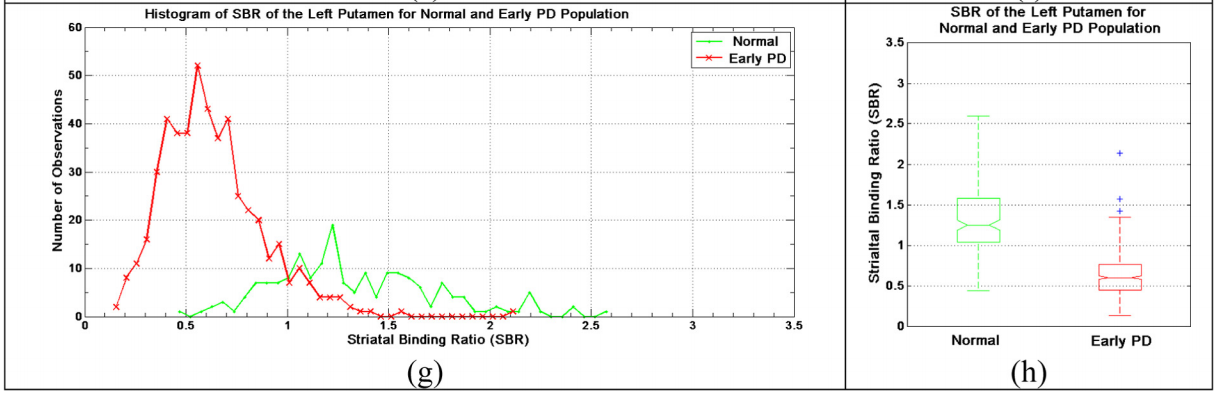












Choix de la base d’apprentissage/ de test. Cross validation pour SVM et 2/3 ; 1/3 pour le reste.

**Erreurs commises par l’article** :

La base de donnée contient en fait 811 patients au lieu de 736. Seulement il y a certains patients qui ont étés analysés plusieurs fois (à des dates différentes) et donc ces variables ne sont plus indépendantes ! J’ai donc bien sélectionné des patients différents 🡺 736 patients

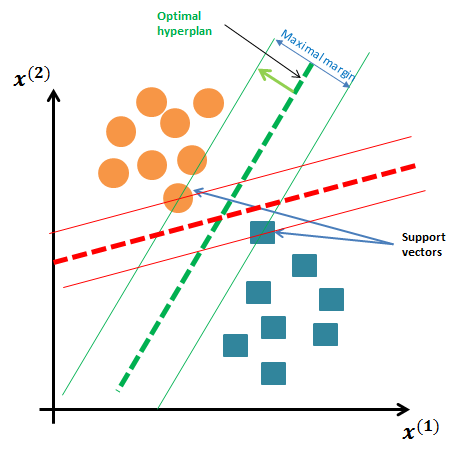
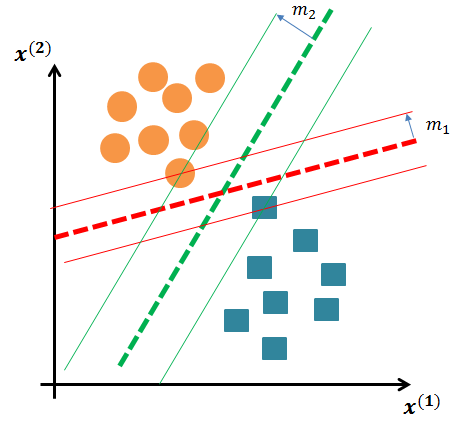
Pour la validation du logit, il ne découpe pas ces bases en training/test. Il a donc forcément de meilleurs résultats !

**Design du classifieur SVM**

Compréhension SVM :

SVM est une technique d’apprentissage supervisé qui offre une méthode pour définir l’hyper plan de décision. Ceci sans supposition de distribution à priori de nos classes. L’idée est de trouver l’hyperplan qui maximise la marge, c’est-à-dire l’hyperplan qui « fait s‘éloigner » les points les plus critiques (appelés vecteurs de support). En effet, ce sont ces vecteurs de support qui pose le plus de problème.

Avantages / Desavantages <http://www.diw-berlin.de/documents/publikationen/73/88369/dp811.pdf>



Est l’équation de l’hyper plan de décision.

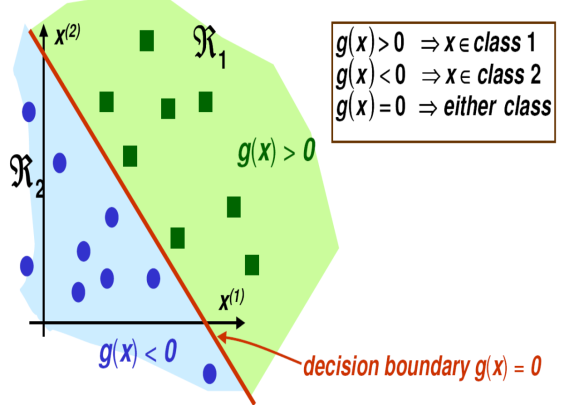
La distance d’un point à l’hyper-plan optimal est de

En prenant les vecteurs de support, on peut retrouver la taille de la marge

Le but est de maximiser cette marge ou minimiser , SVM propose ainsi de résoudre le problème de minimisation suivant :

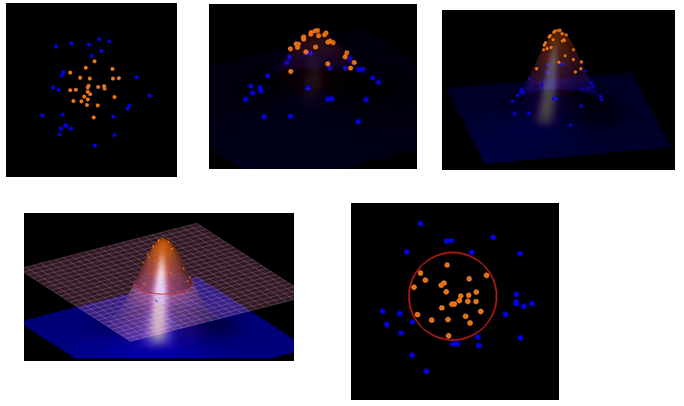
Avec la contrainte (dû au recouvrement) :

La décision {-1,1} sur la classe du point est donné par la fonction :



Si on veut projeter dans un espace de caractéristique plus élevé, on peut utiliser la fonction noyau , la décision devient alors :

permet ainsi d’obtenir des frontières de décision non linéaire, comme par exemple avec un noyau gaussien.



J’ai utilisé SVM – light

<http://download.joachims.org/svm_light/current/svm_light.tar.gz>

La première étape est de designer le classifieur, pour cela il faut choisir les bons paramètres. J’ai décidé d’effectuer ce choix des paramètres sur tout ma base de donnée, ainsi je prends de cette base qui servira de training pour le SVM et servira pour la validation.

Je ressors une erreur pour chaque paramètre, et c’est la plus petite erreur va me permettre de déterminer le meilleur paramètre. On retrouve en général 2 paramètres :

* Paramètre
* Paramètres liés au noyau

apparaît lorsque l’on a des problèmes de recouvrement. C’est lui qui va définir le compromis entre la marge maximale et les erreurs de classification appelées la pénalité de l’erreur.

J’ai choisi de faire varier de à ce qui permet d’avoir une idée assez globale de la meilleure valeur. [http://mi.eng.cam.ac.uk/~kkc21/thesis\_main/node29.html]

Pour définir les meilleurs paramètres pour chaque noyau, je fais à chaque fois varier tous les paramètres un à un, et je stocke les erreurs de classification ressorties par le SVM dans une hypermatrice d’erreur de dimension avec le nombre de variations sur les paramètres et le nombre de paramètres.

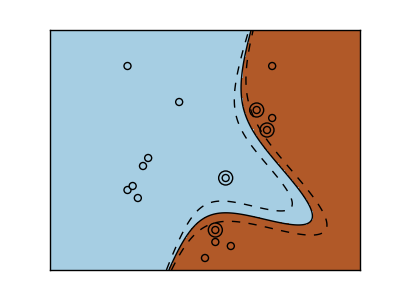
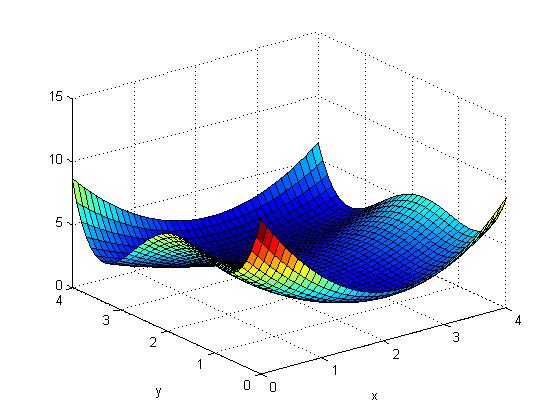
J’ai opté pour 4 noyaux différents pour la suite.

Noyau linéaire :

C’est le SVM de base où aucune transformation sur notre espace de caractéristique n’est effectuée. Il n’y a pas de paramètre spécifique lié au noyau, le seul paramètre est .

Noyau polynomial :

C’est un noyau non-stationnaire qui est plutôt utilisé dans des problématiques de régression où les données sont normalisées.

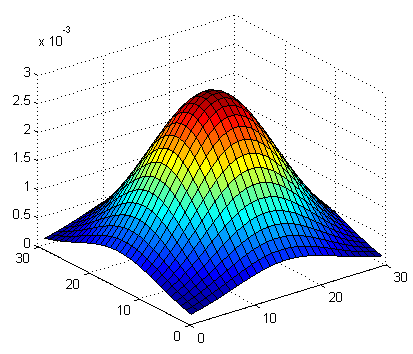
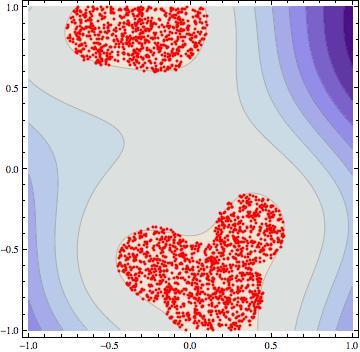
 

On retrouve 3 paramètres ajustables :

* Le degré du polynôme de à
* L’offset de à
* Le facteur de à

Noyau gaussien :

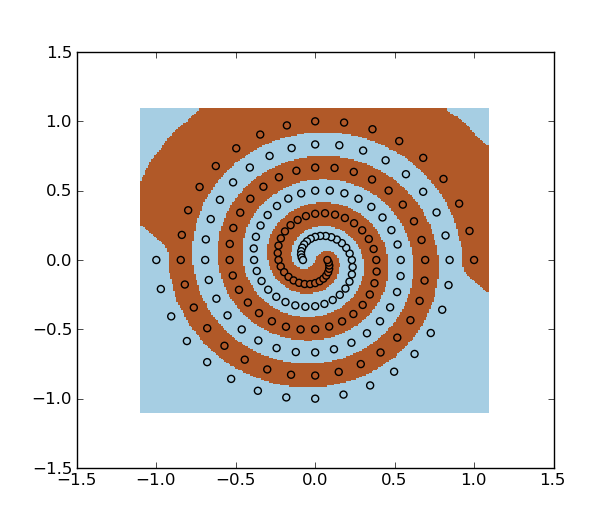
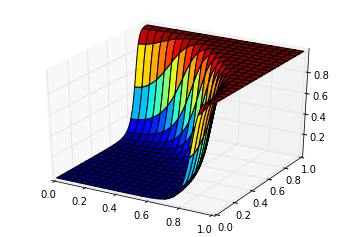
C’est un des noyaux les plus populaires car il permet de classifier sous des formes d’ellipses.



On ne retrouve qu’un seul paramètre ajustable ce qui en fait un noyau relativement simple à analyser. Il s’agit de la constante que j’ai choisi de faire varier de à .

Noyau sigmoid :

Il est vient des réseaux de neuronnes pour permettre leur activation et l’utilisation de ce noyau revient au perceptron d’un réseau de neuronne à 2 couches. Il est indiqué qu’il offre de bons résultats en pratique [http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf]



On retrouve 2 paramètres ajustables :

* L’offset de à
* Le facteur de à

Comme notre base contient plus de Parkinson que normal, les frontière à tendance à être du côté des parkinsons pour « équilibrer » les zones.

On peut aussi jouer sur ce que l’on préfère classer. Vaut-il mieux classer à tord des normaux en tant que malade (dans ce cas on jour sur les paramétres pour que la frontière soit dans les parkinsons). Ou alors on accepte de classer des malades en tant que normaux (on décale la frontière dans la zone des normaux).  
  
J’ai choisi la première option, en effet classer un malade en tant que normal est beaucoup plus dangereux que l’inverse.

Linéaire

|  |  |  |
| --- | --- | --- |
| Label \ prediction | Normal | Parkinson |
| Normal | 9 | 2 |
| Parkinson | 4 | 33 |

Polynomiale

|  |  |  |
| --- | --- | --- |
| Label \ prediction | Normal | Parkinson |
| Normal | 10 | 2 |
| Parkinson | 3 | 36 |

Sigmoïde

|  |  |  |
| --- | --- | --- |
| Label \ prediction | Normal | Parkinson |
| Normal | 10 | 3 |
| Parkinson | 2 | 35 |

Gaussien

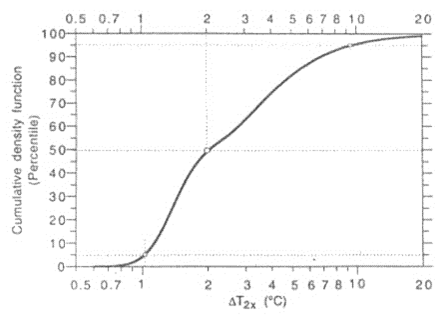
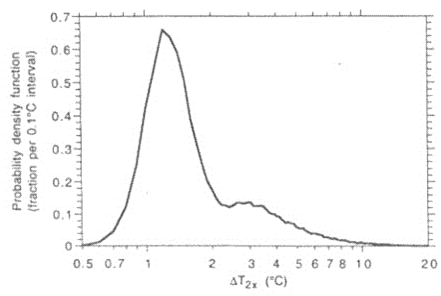
|  |  |  |
| --- | --- | --- |
| Label \ prediction | Normal | Parkinson |
| Normal | 12 | 1 |
| Parkinson | 2 | 35 |

**Design du de la régression logistique**

Compréhension : <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1065119/>

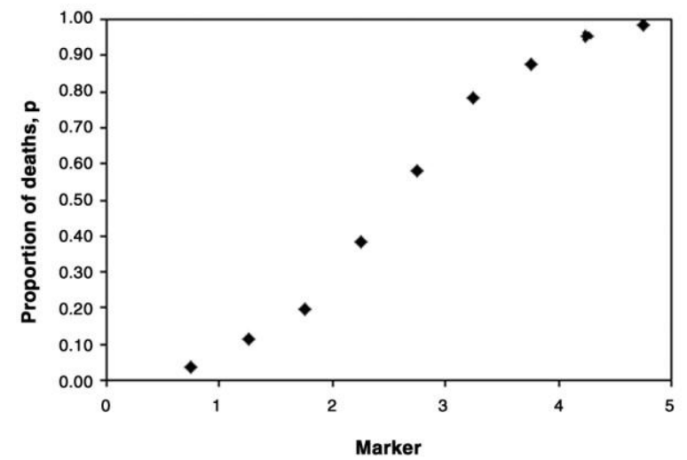
La régression logistique est un modèle de régression binomiale (réponse 1 ou 0). On veut prédire la probabilité d’un évènement binaire (parkinson ou non) en connaissant les variables aléatoire (caractéristiques SBR du patient). Très répandu en médecine pour trouver des facteurs qui caractérisent un groupe de malade par exemple.

Le but est d’étudier non plus la fonction de densité mais son intégrale, c’est-à-dire la fonction de répartition (la probabilité) :



La probabilité que le patient définit par les caractéristiques soit atteint de la maladie de parkinson est :

On veut linéariser la fonction de répartition pour récupérer facilement les probabilités, pour cela on passe par le modèle dit « logit ».

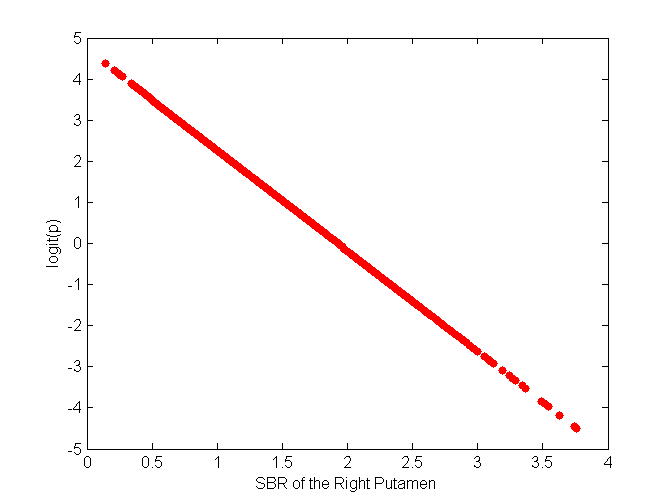
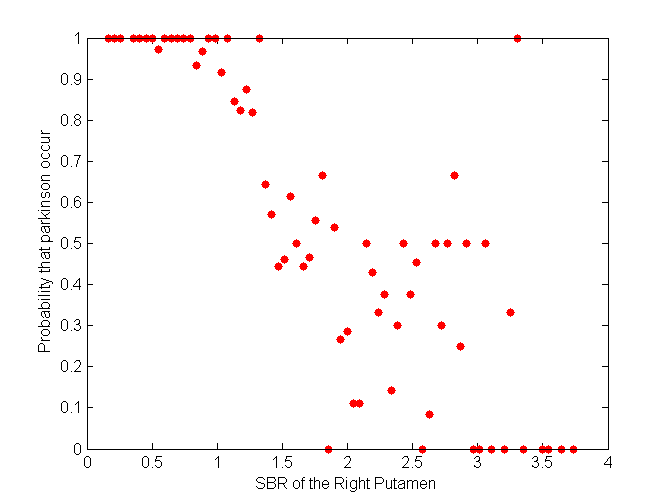
🡺 An external file that holds a picture, illustration, etc.
Object name is cc3045-2.jpg

Comme on est dans un cas multivarié, l’équation devient pour un patient  :

et sont calculés en utilisant la méthode du maximum de vraisemblance de Likelihood. On fait varier les paramètres et et on cherche la probabilité maximale par méthode itérative de recherche de maximum. Une fois les paramètres trouvés, la probabilité est de :

<http://www.mathworks.com/help/stats/mnrfit.html#outputarg_stats>

On analyse ensuite les résultats à l’aide d’informations statistiques.



La variance des points est lié au taux de recouvrement des 2 classes.

J’ai tout d’abord calculer le logit avec ma base de donnée, un paramètre important est la p-valeur qui donne une information sur le degré de confiance du résultat. En général on veut que cette valeur soit inférieur à .

|  |  |  |
| --- | --- | --- |
| **Prédicteur** |  | **-valeur** |
| Constante |  |  |
| Right Caudate |  |  |
| Left Caudate |  |  |
| Right Putamen |  |  |
| Left Putamen |  |  |

On voit que pour les SBR du Caudate, on a des -valeurs extrêmement élevés. C’est pourquoi on peut multiplier ces valeurs entre elles et espérer obtenir une meilleure -valeur ( ce qui est déjà plus acceptable).

Finalement, la base de donnée ne contient plus que 3 caractéristiques  , et .

Au niveau des résultats, cross validation avec le choix du nombre de box à l’utilisateur.

Plus le nombre est petit, plus les boxes sont grosses. Cross validation permet donc de vérifier l’impact du learning en quelque sort. On voit qu’il est très important car dans le cas de la logit, plus la taille des boxes est grande, meilleurs sont les performances.

J’ai choisi d’utiliser 50 « boites » de taille 14, le 14 ème point sert de test alors que les 13 autres servent de training.

Le fait de multiplier ensemble les 2 SBR du Caudate améliore les performances de quelques pourcents globalement.

Matrice de confusion globale :

|  |  |  |
| --- | --- | --- |
|  | Normal | Parkinson |
| Normal |  |  |
| Parkinson |  |  |

d’erreurs sur les normaux

d’erreurs sur parkinson

Total de d’erreur

Analyse statistique sur **toute la base de données** :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prédicteur** |  | **-valeur** | **Erreur standard** | **-Test de Student** |
| Constante |  |  |  |  |
| Right Caudate Left Caudate |  |  |  |  |
| Right Putamen |  |  |  |  |
| Left Putamen |  |  |  |  |

On remarque que cette fois-ci, les -valeurs sont heureusement plus faibles que précédemment, on reste en dessous du seuil fatidique de . Les erreurs ne sont pas si élevées, les marges de valeur sur les prédicteurs ne sont ainsi pas beaucoup impactées.

On peut aussi calculer le degré de liberté, qui nous donne une information sur le nombre d’équations en plus que l’on a. Il est de ce qui est assez élevé mais logique au vu de notre base de donnée ( patients) pour prédicteurs. Le fait d’avoir remplacé 2 caractéristiques par 1 seule augmente le degré de liberté ce qui peut s’avérer néfaste au niveau algorithmique.

Le -Test ou la statistique de donne une information sur l’importance du prédicteur, plus cette valeur est élevée, plus le prédicteur est important pour la prédiction finale. En utilisant la table de la loi de avec degrés de liberté, on remarque que les prédicteurs ont tendance à ne pas être si performant que ça.

La matrice de corrélation permet de nous dire si nos prédicteurs sont bien indépendants entre eux :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Correlation Matrix |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Elle donne des résultats satisfaisant (surtout pour la constante) mais on remarque que le prédicteur du Left Putamen à tendance à dépendre au Right Putamen (ce qui est physiquement logique si les marqueurs sont situés dans la zone entre les 2).

Enfin, le coefficient de détermination (idéalement) indique que nos données ont bien tendance à être incluses par la régression.

**Approche bayésienne**

En supposant que l’on connaisse l’information à priori qui va nous aider à la décision (dans la chaîne, les cerises apparaissent ¼ du temps), et les densités conditionnelles des classes (la loi de notre modèle). On suppose aussi que l’on arrive à mesure notre vecteur de caractéristique .

ATTENTION, on ne prend que 2/3 de la base de donnée pour la modélisation, ATTENTION, dans mon projet les proba à priori ne sont pas les mêmes

Règle de Bayes :

Avec

Pour modéliser notre densité de probabilité , 2 approches :

* Soit approche paramétrique en déterminant les paramètres de notre loi (gaussienne) à l’aide du max de vraisemblance. Une fois trouvé, on applique la formule suivante pour classifier notre point :
* Soit approche non paramétrique, on caclule notre densité de probabilité à l’aide des fenêtres de parzen. On applique ensuite la régle de décision bayésienne pour classifier notre point.

Pour calculer , le est le point entrant. Il suffit de regarder la fonction de densité (que l’on a calculer) pour le point le plus proche de

Résultats Paramétrique

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Right Caudate | Left Caudate | Right Putamen | Left Putamen |
| Normal |  |  |  |  |
| Parkinson |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Normal | Right Caudate | Left Caudate | Right Putamen | Left Putamen |
| Right Caudate |  |  |  |  |
| Left Caudate |  |  |  |  |
| Right Putamen |  |  |  |  |
| Left Putamen |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parkinson | Right Caudate | Left Caudate | Right Putamen | Left Putamen |
| Right Caudate |  |  |  |  |
| Left Caudate |  |  |  |  |
| Right Putamen |  |  |  |  |
| Left Putamen |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | Normal | Parkinson |
| Normal |  |  |
| Parkinson |  |  |

Résultat non-paramétrique

|  |  |  |
| --- | --- | --- |
|  | Normal | Parkinson |
| Normal |  |  |
| Parkinson |  |  |

**Comparaison et analyse des performances**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SVM | MLR | Bayes param | Bayes non param |
| Error rate |  |  |  |  |
| Time |  |  |  |  |
| Complexity |  |  |  |  |
| Memory consumption | (in MATLAB) |  |  |  |