Projet 3A
Analyse d'imagerie polarimétrique

Guinaudeau Alexandre Hulot Pierre Dejoie Etienne

19 février 2016

Résumé

Le résumé (abstract en anglais) de mon article.

Introduction

Aujourd'hui, le dépistage du cancer du col de l'utérus est effectué à l'œil nu. Si la patiente est effectivement atteinte d'un cancer, un échantillon (appelé conisation) est prélevé, et un spécialiste découpe l'échantillon pour déterminer les zones atteintes. *ADM polar* cherche à détecter les cellules cancéreuses à partir d'imagerie polarimétrique. L'idée à terme serait de concevoir un outil qui permette de détecter les zones effectivement atteintes *in vivo*, pour ainsi mieux délimiter la zone à prélever et réduire les risques de prélèvement pour les patientes saines.

ADM polar fait analyser ces échantillons par un spécialiste, pour déterminer l'état (sain, malade, bénin...) des cellules le long de coupes. Puis, après avoir pris plusieurs images dans des configurations de polarisation différentes, *ADM polar* reconstitue la matrice de Müller de l'échantillon, un ensemble de 16 images qui représentent l'état de polarisation. Il reste alors à déterminer la corrélation entre les valeurs dans la matrice de Müller d'une cellule et son diagnostique.

Dans le cadre du projet de 3e année, nous avons souhaité appliquer les connaissances théoriques apprises en cours à des données réelles. Nous avons donc rejoint le projet d'*ADM polar*, dans le but de détecter les cellules cancéreuses, à partir des images polarimétriques fournies par *ADM polar*.

La première étape a été la visualisation des données, pour mieux savoir quels modèles d'apprentissage seraient les plus susceptibles de porter leurs fruits. Puis, nous avons cherché à réduire la dimension du problème en sélectionnant ls paramètres les plus discriminants dans la classification des cellules. Enfin, nous avons testé différents modèles de classification et mesuré leur performance en termes de précision et de coût de calcul.

Chapitre 1

Contexte

1.1 ADM Polar, contexte du projet

1.2 Présentation des données

Pour ce projet nous disposions d'un jeu de données de 163000 pixels répartis sur 17 images différentes. Pour chaque pixel nous avions les 16 éléments de la matrice de Müller, leur position, leur diagnostic, et l'image correspondante. Ces Pixels sont répartis en 4/5 de sains et 1/5 de malades. L'affichage des pixels nous montre que dans chaque image, les pixels sont répartis en zones au diagnostic identique. Sur les 17 images, 11 possèdent que des zones saines, une a une zone saine et une zone malade et les 5 dernières ont que des zones malades. Le nombre d'images étant assez limité il n'a été facile de trouver des points communs aux pixels sain et aux pixels malades.

1.2.1 la Matrice de Müller

Les données sont obtenues par polarisation de la lumière. De la lumière polarisée est envoyée sur l'échantillon, réfléchi par celui ci est analysé ensuite. La matrice de Müller est obtenue à partir de l'analyse de la polarisation de la lumière réfléchie. Le postulat du laboratoire sur lequel repose tout le traitement des données est que les cellules saines et malades polarisent différemment la lumière. Cela implique qu'il est possible de reconnaître une cellule malade par la polarisation de la lumière réfléchie. Le PICM se bases pour cela d'observations de relation entre les images polarisées et le diagnostic. La polarisation génère 16 images par échantillon. On associe alors à chaque pixel une matrice $4x4$ appelée matrice de Müller. Dans cette matrice, la première ligne et la première colonne représentent l'intensité du signal, ces éléments influent peu sur le diagnostic mais ont des valeur élevés, ils introduisent beaucoup de biais et de sur-apprentissage. Nous avons donc décidé de ne pas les prendre en compte. Les éléments diagonaux sont normalement censé être les plus significatifs mais les reflets introduisent beaucoup de bruit rendant ces données difficilement utilisables. Il reste 6 éléments dans la matrices sur lesquels nous allons nous concentrer.

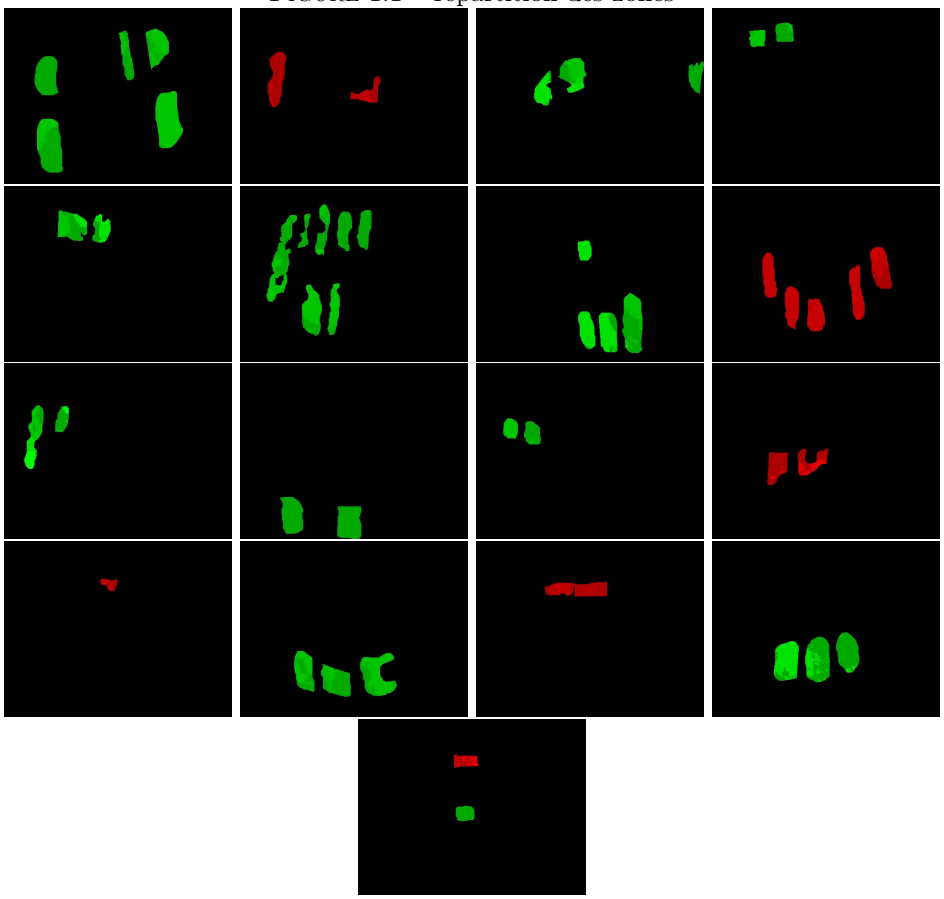
1.2.2 Le Clustering

La disposition des données n'étant pas facilement exploitable directement il fallait réduire la complexité du problème, par exemple en limitant le nombre de données. Nous avons ainsi décidé de clustériser les données de chaque image afin d'en avoir beaucoup moins, mais plus significatives. La position du pixel dans l'image n'a pas été prise en compte, même si les zones résultantes sont proches. La clustérisation à été faite par un algorithme de type KNN (k plus proches voisins) sur les 6 éléments significatifs de la matrice de Müller. Les diagnostics étant propres à chaque image (une seule image avec un diagnostic non uniforme), chaque cluster est associé à un diagnostic.

La méthode de clustering utilisé consiste en :

— choisir k, le nombre de clusters
— regrouper les pixels de l'image selon ces k clusters (par KNN)
— calculer l'erreur totale commise (somme des distances au centre du cluster)

FIGURE 1.1 – répartition des zones



— recommencer avec un k plus grand si l'erreur est trop grande

Cette méthode nous a permis de passer de 163000 points à 52 plus significatifs. Elle nous a aussi permis de mieux comprendre la répartition des points. Le clustering a surtout un apport au niveau de la compréhension des données. Il pourrait avoir un impact (en tant que tel) sur les résultats, mais son but a été surtout de mieux comprendre la disposition des données pour pouvoir apprendre dessus.

Chapitre 2

Le traitement des données

2.1 Prétraitement des données

Le Prétraitement de données est l'étape essentielle qui précède l'apprentissage. Elle est la clé de celui ci, un bon prétraitement permet d'éliminer le bruit et présente les données sous un angle facilement exploitable. C'est là que se situe tout l'enjeu d'un classifieur Big Data ainsi que le travail d'un data analyst, les méthodes d'apprentissage étant préexistantes.

Les différents types de prétraitements que l'on peut faire avant de traiter les données

2.2 Les différentes approches de traitement des données

2.2.1 La visualisation des données

La première étape du traitement des données est sa visualisation, pour déterminer les modèles de classification les plus prometteurs.

L'affichage des 16 images de la matrice de Müller des échantillons nous montre que les éléments diagonaux se distinguent des autres : ils correspondent à la conservation d'une composante de polarisation (rectiligne, circulaire...), et il y a donc une variance des valeurs bien plus importantes que sur les autre composantes. Par ailleurs, les images de la première ligne et de la première colonne paraissent très bruitées.

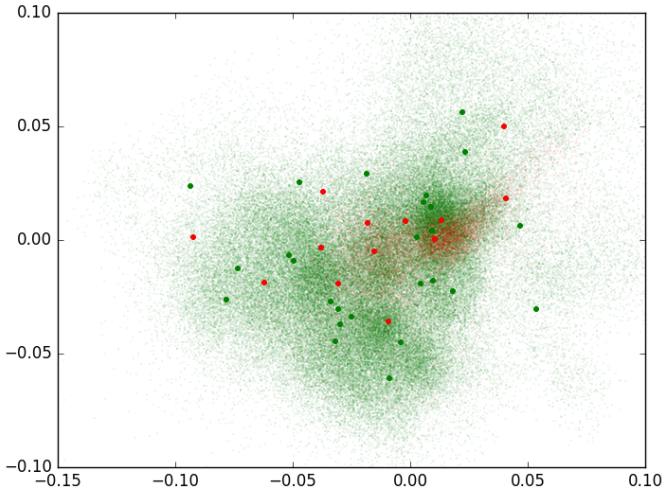
A partir de ces images et de l'interprétation des chercheurs, nous avons déduit que la première colonne et la première ligne ne permettraient pas de discriminer les cellules. De plus, les données sont normalisées par rapport à l'élément M11 de la matrice de Müller. Or, cet élément correspond à l'intensité lumineuse de l'échantillon (autrement dit, c'est une image en noir et blanc de l'échantillon). Cet élément dépend donc fortement des conditions de l'expérience, et en particulier certaines zones peuvent être très lumineuses si elles sont éclairées, ce qui est évidemment indépendant du diagnostique des cellules. Nous avons donc décidé de multiplier les autres images par la composant M11, afin d'obtenir des valeurs comparables d'une image à l'autre. Nous avons donc décidé de réduire la dimension de 16 à 9 voire 6, en ne sélectionnant que les éléments Mij avec i et j différents de 1, voire uniquement les éléments extradiagonaux, pour que les résultats ne soient pas faussés par la forte différence de variance entre les éléments diagonaux et extradiagonaux de la matrice.

Enfin, pour mesurer la performance d'une méthode de classification, nous avons décidé de tester sur des images indépendantes des images d'apprentissage. Tester les pixels d'une même image risque trop fortement de créer un phénomène de surapprentissage (overfitting), en classant les pixels comme leurs voisins, ce qui n'a pas d'intérêt.

Nous avons ensuite décidé de visualiser les données dans un espace de plus petite dimension. En le projetant dans un espace de dimension 2 optimal (voir la section PCA), on obtient l'image suivante :

Chaque pixel est représenté par un petit point, vert s'il est sain et rouge s'il est malade. Les gros points sont les centres des clusters, qui représentent le milieu d'un ensemble de points proches

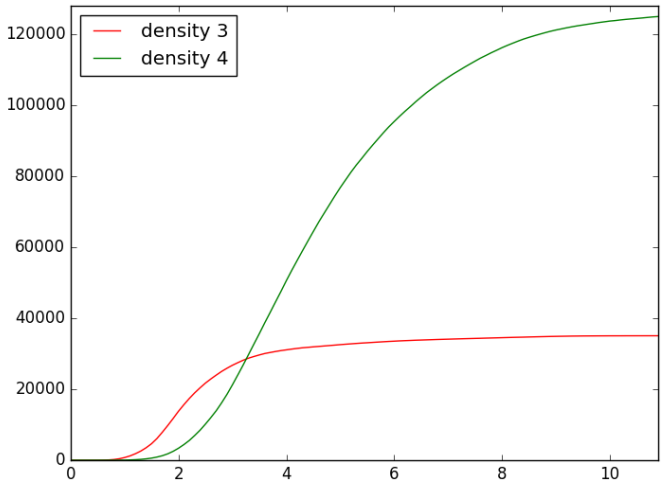
FIGURE 2.1 – Affichage des points et des clusters en dimension 2



d'une même image. On a retiré ici les valeurs abhérentes, certains points isolés étant très éloignés de tous les autres.

Cette visualisation des données nous apprend que les points malades sont moins dispersés que les points sains, et que les moyennes des points sains et malades sont décalées. Plus précisément, on peut alors afficher les densités de points situés à une certaine distance (dans l'espace de dimension 6) de la moyenne de tous les points.

FIGURE 2.2 – Densité de points en fonction de leur distance à la moyenne des points



On remarque qu'à proximité de ce centre, il y a plus de points malades que sains, bien qu'il y ait quatre fois plus de points sains au total. Les points sains sont pour la plupart bien plus éloignés. Cette distance semble donc être un facteur intéressant à prendre en compte pour la suite.

2.2.2 Réduction de dimension

PCA

rappel de la méthode L'Analyse en Composantes Principales (ou PCA) consiste à essayer de représenter les données dans un espace de plus petites dimensions. Les vecteurs directeurs du nouvel espace maximise la variance entre les données. Nous présentons ici les résultats pour la dimension 2.

prétraitement utilisé Nous effectuons cette PCA sur les centres des clusters préalablement présentés (cf 1.1.1). Les centres des clusters représentent de manière fidèle l'ensemble des points qu'il rassemble. Chaque cluster est représenté par un vecteur d'éléments de la matrice de Müller. Tous les éléments de Müller sont gardés à l'exception de la première ligne et première colonne qui ne sont pas a priori pertinentes (d'après les informations des physiciens)

FIGURE 2.3 – centre des clusters avant transformation

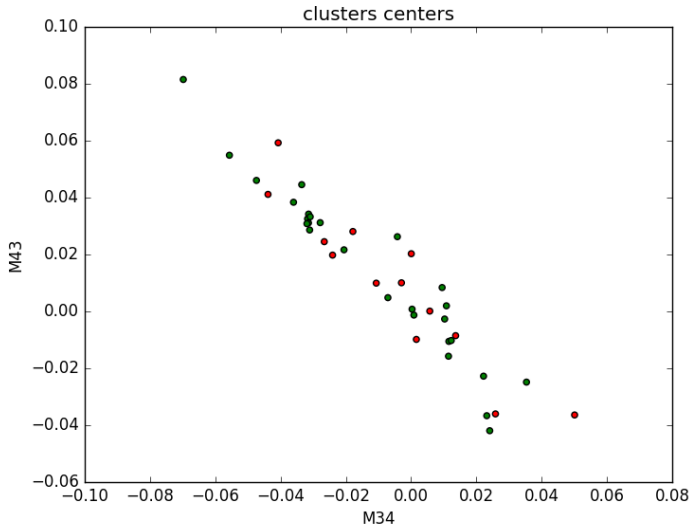


FIGURE 2.4 – centre des clusters après transformation

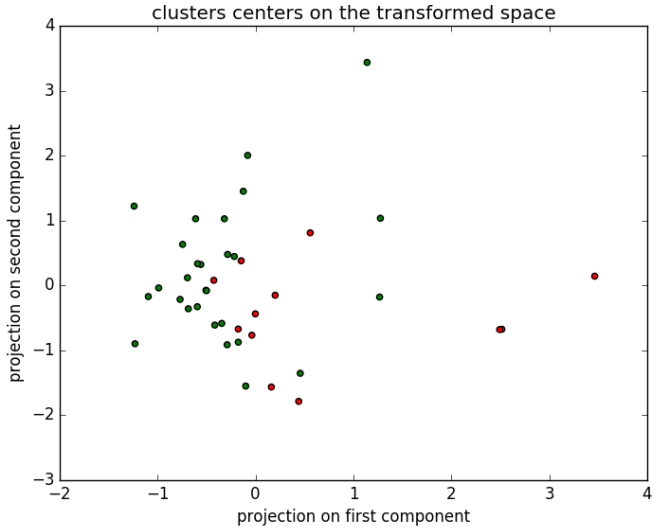


FIGURE 2.5 – Part de variance expliquée

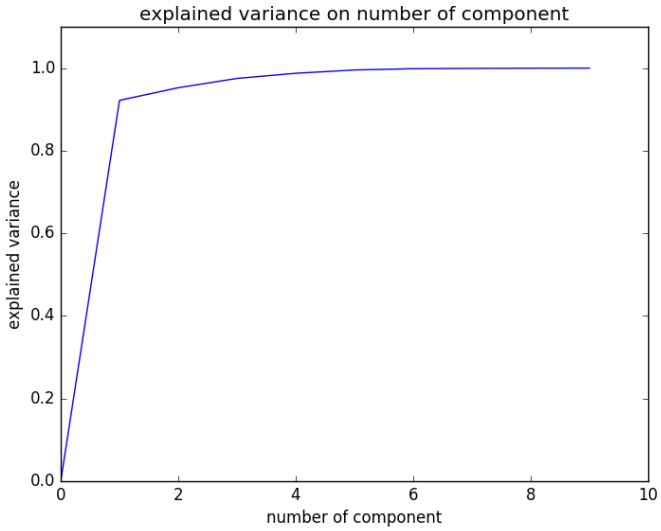
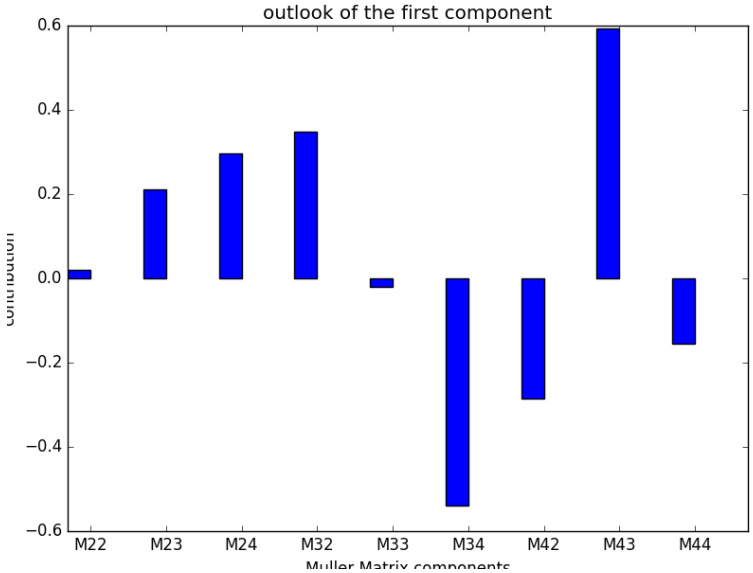


FIGURE 2.6 – analyse de la composante principale



résultats La réduction de dimension par PCA semble efficace. La composante principale explique 90% de la variance (fig 2.3). De l'analyse de la première composante (fig 2.4) ressort deux effets principaux : - La petite contribution des éléments diagonaux de la matrice de Müller - Le rôle prépondérant de M34 et M43

On remarque une certaine anti-corrélation des éléments de la matrice de Müller. Le poids de M43 est proche de l'opposé de celui de M34. Le poids de M42 est également proche de l'opposé de celui de M24. Cette observation n'est par contre pas vérifiée pour M23 et M32 qui semblent corrélés.

conclusion La PCA effectuée sur les centres des clusters valident certaines supposition comme le rôle faible des éléments diagonaux ou le rôle important des éléments M34 et M43.

Par contre, la projection de la PCA en 2 dimensions ne nous permet pas de séparer les données de manière suffisantes pour être capable de distinguer des zones clairement différentes entre les clusters sains et les clusters malades. (fig 2.2)

2.2.3 Méthode de classification

Arbre décisionnel et Random Forest

rappel de la méthode

Arbre décisionnel L'arbre décisionnel est une méthode de classification très classique, qui donne souvent de bon résultats. L'idée est trouver un hyperplan qui sépare au mieux les données (parfois très simple, sur une variable uniquement) et de recommencer cela sur les deux sous ensembles de données ainsi crées. On isole ainsi les zones où les points sont similaires, et l'on peut ainsi prédire le diagnostique d'un pixel, les hyperplans permettant de construire un arbre de décision.

Random Forest La Random Forest ou forêt d'arbre, est juste un ensemble d'arbres de décision. Les données sont divisées aléatoirement, puis l'on associe à chaque sous ensemble un arbre de décision. Le résultat de la prédiction se fait par vote, chaque arbre donne sa prédiction, la prédiction majoritaire l'emporte.

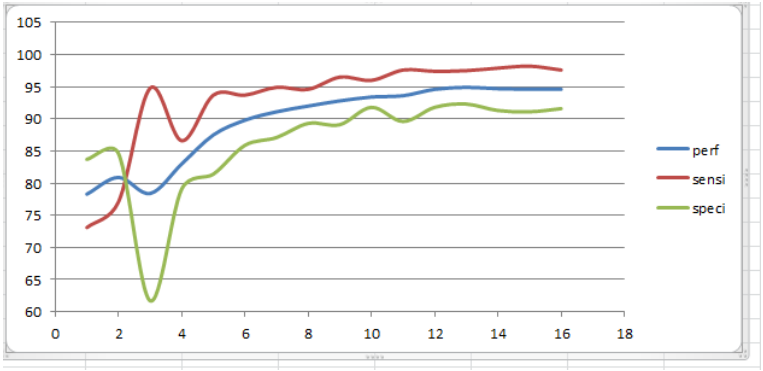
prétraitement utilisé Nous avons commencé à apprendre sur toutes les données. Au début nous avions uniquement les pixels, sans les images associés, puis pour préciser les résultats et limiter le sur-apprentissage nous les avons classés par image. Nous avons appris au début sur tous les paramètres de la matrice de Müller, puis uniquement sur les 6 éléments significatifs.

résultats (notamment graphique) L'apprentissage sur les 16 éléments de la matrice de Müller a résulté dans fort sur-apprentissage : l'arbre de décision réussissait à retrouver l'image de provenance, la forte corrélation entre image et diagnostic ne nous permet donc pas d'aller plus loin. L'apprentissage sur 16 images et le test sur la 17éme a permis de dévoiler ce sur-apprentissage, avec des performances renversées et aléatoires. Sur la figure qui suit on peut voir des résultats très élevés (de l'ordre de 95%) qui témoignent plus d'un sur-apprentissage que d'un réel résultat. En effet la séparation de par image entre ensemble de test et d'apprentissage donne des résultats beaucoup plus mitigés (autour de 40%). La restriction à 6 éléments de la matrice n'améliore pas la performance.

explication Le faible nombre d'images à diagnostic non uniforme conduit l'algorithme à choisir l'image pour déterminer le diagnostic, les différences entre les images étant plus importantes que les différences de diagnostic. L'apprentissage est donc nécessairement biaisé.

piste d'amélioration Pour résoudre ce problème il faudrait avoir accès à des images au diagnostic varié, images sur lesquelles un réel apprentissage est possible. La difficulté à obtenir ces images nous a poussé a changer de méthode d'apprentissage et à pousser plus loin la compréhension que l'on avait des données.

FIGURE 2.7 – résultats de la Random forest (pixels mélangés)



K plus proche voisin

rappel de la méthode La méthode des k plus proche voisin consiste à essayer de prédire l'état d'un nouveaux point en se basant sur l'état de ses voisins les plus proches. Cette méthode a l'avantage de pouvoir classer les données selon des schémas non linéaires. Par contre, c'est une méthode sensible à la dimension.

prétraitement utilisé Nous avons testé la méthodes des KNN sur les centres des clusters pour plusieurs raisons : - A chaque classification, l'algorithme doit recalculer l'ensemble des distances avec tout l'échantillon d'apprentissage. Il y a donc une nécessité de réduire le nombre de donnée en entrée sur lesquels on calcule les distances. - De plus, chaque cluster regroupe un ensemble de point très proches les uns des autres. Ainsi, en prenant tous les éléments, les plus proches voisins d'un certain point auraient souvent tous appartenu au même cluster ce qui rend l'information extraite redondante. Cela aurait revenu à utiliser l'algorithme avec 1 seul voisin.

résultats Les résultats présentés ci dessous corresponde au taux de bonne prédiction en prenant une image de test et en cherchant les k plus proches voisins sur les 16 autres. Chaque cluster est représenté par un vecteur comprenant les éléments suivants de la matrice de Müller : ['M23', 'M24', 'M32', 'M34', 'M42', 'M43'].

échantillon de test	taux de bon résultat
1	100%
2	0%
3	100%
4	100%
5	0%
6	100%
7	33%
8	50%
9	66%
10	100%
11	100%
12	100%
13	100%
14	75%
15	66%
16	75%
17	100%
moyenne	69%

En utilisant les 9 éléments de la matrice de Müller Mij avec i et j différents de 1, on obtient un taux moyen de 65%.

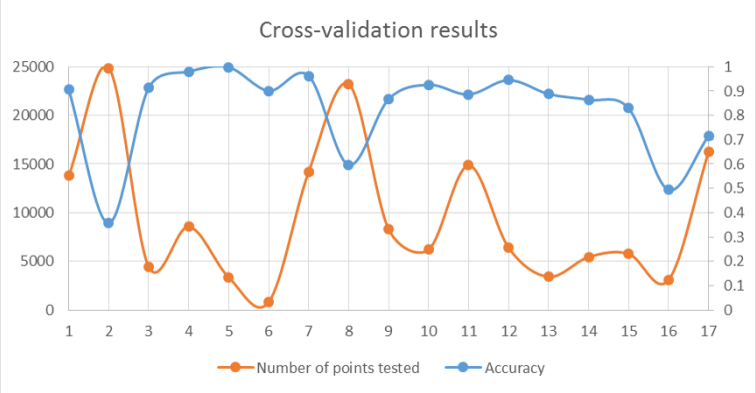
conclusion Les résultats donnés par les KNN ne sont pas très bon. Piste d'explication ?

SVM

Les Machines à Vecteurs Support ou SVM (Support Vector Machine) sont des classifieurs qui cherchent à séparer linéairement deux ensemble de points dans un certain espace en maximisant la marge entre ces deux sets de points. Par défaut (avec un noyeau linéaire), la séparation est donc nécessairement linéaire. Cependant, en transformant nos données pour les plongées dans un autre espace, souvent de dimension supérieur, on peut générer un modèle de capacité supérieur et fortement non linéaire. C'est ce qu'on appelle le Kernel trick

prétraitement utilisé Les SVM sont efficaces lorsque le nombre de feature est inférieur (voire très inférieur) au nombre de données. On applique donc cette méthode aux points directement, et non pas aux centres des clusters.

FIGURE 2.8 – Résultats de la méthode SVM : proportion de bonnes prédictions et nombre de pixels dans l'ensemble de test, pour chaque image.

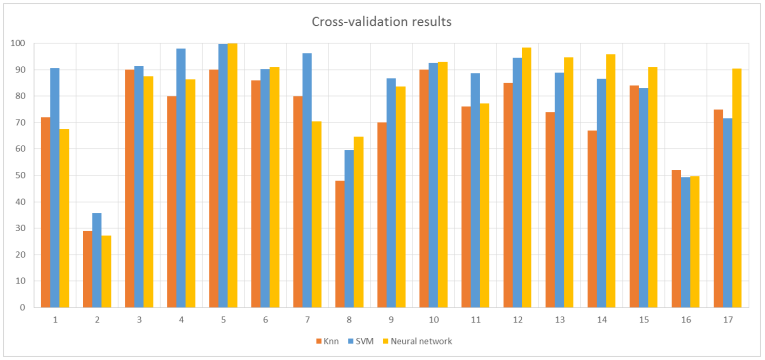


résultats La moyenne de ces prédictions est de 83%. Pourtant, en pondérant ces résultats par le nombre de points dans l'échantillon de test (pour que les images ayant beaucoup de pixels à classifier aient plus d'importance), on obtient un taux moyen de seulement 76%. En effet, les images ayant beaucoup de pixels ont des moins bons taux de prédiction, ce qui pourrait indiquer que nous n'avons pas atteint le seuil de données critiques nécessaires à des résultats optimaux. Avec quelques images supplémentaires, on pourrait espérer garantir un taux de précision d'au moins 80% avec cette méthode. Dans les circonstances actuelles, certaines images ont des résultats très inférieurs à cette valeur moyenne, voire en-deçà de 50% : cette méthode n'est donc pas assez fiable pour permettre de bien classifier les données.

explication et piste d'amélioration Le problème du Kernel trick est qu'il necessite d'avoir une idée a priori du type de fonction que l'on cherche à estimer pour pouvoir utiliser un noyeau adequat. Ce n'est ici pas le cas, on ne sait pas a priori qu'elle pourrait être le type de fonction permettant de distinguer linéairement un pixel sain d'un pixel malade.

Pour contrer ce problème, il faudrait "apprendre la fonction noyeau". C'est en quelque sorte le rôle des neural networks.. On peut voir les n-1 couches du réseau de neuronnes comme étant la fonction du noyeau et la dernière comme une séparation linéaire sur les données plongées dans ce nouvel espace.

FIGURE 2.9 – Comparaison des taux de bonne prédiction des différentes méthodes de classification testées.



Résultats

Moyenne des taux de précision

Méthode	Moyenne simple	Moyenne pondérée
KNN	73.41	65.75
Neural Network	80.49	71.70
SVM	82.56	75.65
Maximum	85.78	79.17

Ce tableau présente les taux de précision moyens des méthodes précédentes, soit en moyennant sur chaque image, soit en pondérant les résultats par le nombre de pixels de l'image (ainsi, chaque pixel intervient 1 fois dans le calcul final, selon sa classification à partir de 16 autres images). Enfin, on calcule le taux de précision maximal pour les méthodes de classification précédentes, image par image. Si l'on combine les méthodes précédentes, on ne peut pas espérer dépasser cette valeur.

La méthode Knn semble avoir de moins bons résultats que les méthodes SVM et Neural network.

Par ailleurs, on remarque que les images ayant beaucoup de pixels (les images 2 et 8 représentent 30% des pixels) ont des moins bons taux de prédiction, ce qui pourrait indiquer que nous n'avons pas atteint le seuil de données critiques nécessaires à des résultats optimaux. Avec quelques images supplémentaires, on pourrait espérer garantir un taux de précision d'au moins 80% avec cette méthode. Notons enfin que l'image 16 a également un taux de précision très faible, en dépit de son petit combre de points. C'est la seule image ayant à la fois des zones saines et malades, il semble donc que les méthodes aient tendance à prédire le même diagnostic pour les pixels d'une même image. Dans les circonstances actuelles, certaines images ont des résultats très inférieurs à cette valeur moyenne, voire en-deçà de 50% : ces méthodes ne sont donc pas assez fiable pour permettre de bien classifier les données.

Table des matières

Table des figures

Liste des tableaux

Index