

L'objectif de ce TP est de mettre au point un processus d'apprentissage complet sur des jeux de données différents.

Jeu de données CALTECH:

Le premier jeu de données que nous allons voir est CALTECH. Celui-ci contient des images réelles. Ces images ont été décrites par le biais de divers descripteurs. L'objectif sera donc de tester différentes méthodes de classification sur ces divers descripteurs.

Descripteur du patch:

Classification par les arbres

Dans un premier temps nous nous intéresserons au descripteur patches qui identifie les images par un sous échantillonnage à la taille 32 x 32 en niveau de gris soit 1024 pixels.

Les premières données que nous analyserons sont celles issues du fichier caltech_patch. Celles-ci seront utilisées comme ensemble d'apprentissage et ensemble de validation lors de notre apprentissage complet.

Nombre de données		Attributs	Classes
100		1024 valeurs de niveau de gris	faces
100		1024 valeurs de niveau de gris	léopards
100		1024 valeurs de niveau de gris	plane
100		1024 valeurs de niveau de gris	watch
100		1024 valeurs de niveau de gris	moto
Total	500		

Nous nous servirons d'un deuxième fichier de données afin de composer notre ensemble de tests. Nous disposons de 250 données dont 50 dans chacune des classes.

Pour notre apprentissage, comme nous l'avons vu précédemment, nous disposons que d'un seul fichier de données pour l'ensemble de notre apprentissage et l'ensemble de notre validation. C'est pour cela que nous allons séparé notre ensemble de manière aléatoire de la manière suivante : 65 % des données seront utilisé pour l'apprentissage tandis que les 35 autres % seront utilisés pour notre validation. (Cette séparation a été testée par rapport à d'autres et représente la meilleur parmi les séparations testées.) Le deuxième fichier quant à lui constitue l'ensemble de tests.

Ainsi l'ensemble sur lequel s'effectue l'apprentissage contient 70 faces, 61 léopards 66 motos, 65 plane et enfin 63 watch. Par conséquence pour l'ensemble de validation nous avons respectivement 30,39,34,35 et 37 données dans chaque catégorie. Nous construisons l'arbre par la suite mais par soucis d'affichage nous ne le montrerons pas.

Afin de trouver un bon compromis entre performance et complexité, nous allons élagué notre arbre (ceci évitera notamment le surapprentissage) avec notre ensemble de validation.

Nous devrons donc testé tous les élagages possibles et ne garder que le meilleur.

L'arbre élagué obtenu nous permet d'obtenir le résultat suivant :

Modèle	Taux de précision
Arbre de décision	68.4%

Le résultat obtenu n'est pas très séduisant puisque beaucoup d'erreurs sont faites malgré la quantité importante de données. Cela est dû au fait que l'arbre de décision effectue ses choix par rapport à 1 pixel dans une image. Il est difficile de déterminer par un ensemble de quelques pixels si une image sous-échantillonnée est une moto ou un léopard. Malgré la difficulté de l'apprentissage sur cet ensemble de données, le résultat est tout de même respectable.

Classification par le modèle bayésien

Dans cette partie nous allons appliquer la méthode bayésienne. Nous allons utiliser un ensemble de données d'apprentissage de 500 données. Cette fois-ci nous n'avons pas besoin de séparer notre base de données en 2 ensembles car il n'y a pas d'hyper paramètres.

Nous obtenons donc les résultats suivants :

Modèles	Erreurs sur test
Bayésien	69.6%

Le résultat obtenu est similaire à celui acquis par les arbres. On remarque que les erreurs sont issues d'un peu toutes les classes excepté les léopards qui ont très peu d'erreurs. Cela est normal car si l'on regarde les images des léopards, on arrive parfaitement à distinguer une identité unique sur cette image (L'animal a la même couleur sur toutes les photos et est dans un environnement similaire dans chaque image.)

Classification par SVM

_____L'apprentissage ici est très important puisqu'il faut fixer plusieurs hyper paramètres. En effet nous disposons de plusieurs types de kernel pour la SVM : (radial, linéaire, polynomiale) et pour chacun d'entre eux des hyperparamètres leurs sont associés.

Pour le linéaire nous avons besoin d'un coût C afin d'avoir une marge stricte ou une marge souple dans le but d'accepter plus ou moins de valeurs dans notre classification. Le SVM avec un kernel radial lui aura un paramètre de gaussienne, appelé gamma, en plus. Le polynomial aura quant à lui en plus du coût, un degré polynomial et un coefficient pour le degré 0.

Modèles	Taux de précision
---------	-------------------

- Commenté [1]: J'ai 80.4% avec un randomForest
- Commenté [2]: Wow c'est énorme comme résultat
- Commenté [3]: Tu étais pas a 71 -72 hier ?

- Commenté [4]: Pas fini les tests encore sur le nbayes. Je mettrai ça dans l'aprem.

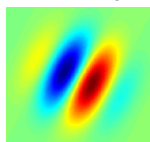
	Meilleur résultat obtenu avec kernel :polynomial, C = 1, degré = 2, coeff0 = 1.1
SVM	82%

Commenté [5]: 83.2% avec un kernel linéaire(!) C=1

Le meilleur résultat obtenu est obtenu avec une SVM polynomial avec un coût C de 1, un degré de 2 et un coeff0 de 1.1. Le SVM étant une méthode permettant de réduire considérablement l'erreur empirique il est tout à fait normal d'obtenir des résultats de cette envergure. Encore une fois les erreurs se font un peu sur toutes les classes excepté le léopard (très peu d'erreur).

Descripteur du gist:

Dans un second temps nous nous intéresserons au descripteur gist qui identifie les images par concaténation des filtres de Gabor sur des grilles de taille 4x4 de l'image. Pour rappel le filtre Gabor est un filtre linéaire utilisé pour les analyses de textures. Il analyse les fréquences d'une image par le biais d'une fonction gaussienne. L'image en sortie devient une image couleur de ce type là.



Filtre de Gabor en 2 dimensions

Les données que nous analyserons sont celles issues du fichier caltech_gist. Celles-ci seront utilisées comme ensemble d'apprentissage et ensemble de validation lors de notre apprentissage complet. Ici nous avons un filtre de Gabor avec 60 dimensions (4 directions et 15 échelles) sur des grilles 4x4 nous avons donc 16x60 valeurs par image. Ainsi nous avons 960 attributs pour chaque image. Ces 960 attributs représentent les valeurs du filtre gabor entre 0 et 1. Mis à part ça le nombre d'éléments par classe est le même qu'auparavant, nous avons 500 images et 250 pour le test.

Pour notre apprentissage, nous ne séparerons pas notre bases de données car nous n'avons pas d'hyper paramètres à définir. Par conséquence nous n'avons pas besoin d'un ensemble de validation.

Classification par les arbres

Modèle	Taux de précision
Arbre de décision	71.2%

Commenté [6]: sdsdf

Le résultat est encore une fois assez faible même si celui-ci est légèrement supérieur qu'avec le descripteur du patch. A noter que pour cette classification, le meilleur arbre était l'arbre non élagué. Beaucoup d'erreurs ont été effectuées sur les motos, les montres ainsi que les avions.

Commenté [7]: J'ai du 84.8% d'accuracy avec un treebag

Classification par le modèle bayésien

Modèles	Taux de précision
Bayésien	80.8%

Le modèle bayésien est nettement meilleur pour le descripteur du gist que son homologue précédent. Toutefois les erreurs s'effectuent de nouveau sur les mêmes classes.

Classification par SVM

Modèles	Taux de précision
	Meilleur résultat obtenu avec kernel :polynomial, C = 0.01, degré = 5, coeff0 = 2.1
SVM	82%

Le meilleur résultat est obtenu avec une SVM polynomial avec un coût C de 0.01, un degré de 5 et un coeff0 de 2,1. Les erreurs se font principalement sur les motos et les montres. Il est en effet difficile de les distinguer étant donné que les motos et les montres ont des couleurs totalement différentes d'une image à l'autre. Qui plus est le fond sur lequel elles sont prises est différent. Il est donc difficile d'établir une règle précise permettant de déterminer à coup sûr si une image est une montre ou une moto.

Descripteur des histogrammes:

Histogramme de luminance

Pour ce descripteur nous disposons d'un ensemble de 500 données comme dans le cas précédent, seul les attributs changent. Comme son nom l'indique l'image est décrite par rapport à son histogramme de luminance, c'est la raison pour laquelle nous avons 256 valeurs de nombres entiers.

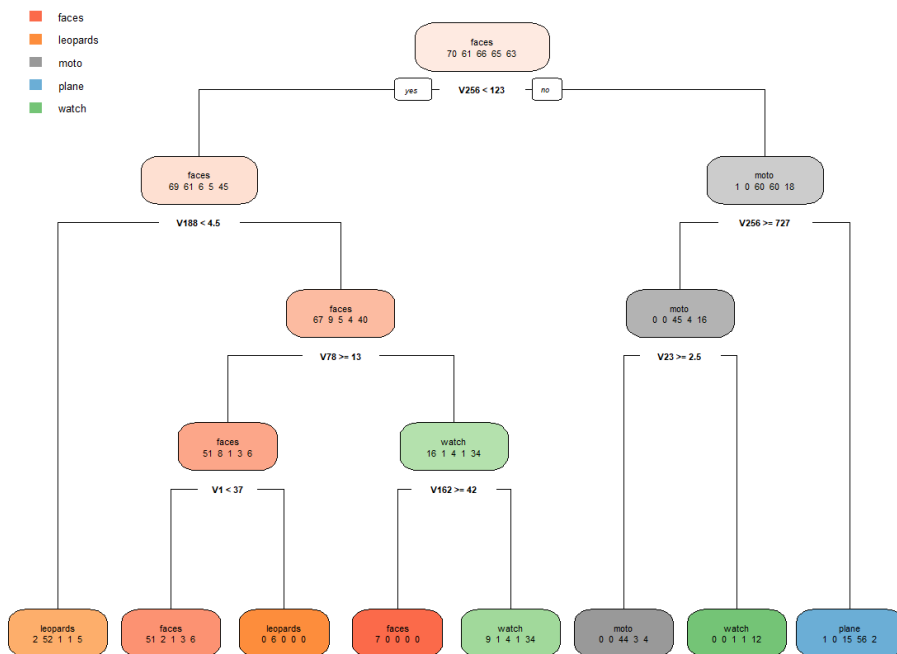
Pour notre apprentissage, nous travaillons une nouvelle fois avec 65 % des données pour l'apprentissage et 35% pour notre validation sauf pour le bayésien.

Classification par les arbres

Modèle	Taux de précision
Arbre de décision	58.8%

Le résultat obtenu est mauvais ici. Nous disposons d'un nombre conséquent d'erreurs sur toutes les classes excepté les léopards encore une fois. Cela est totalement compréhensible étant donné que les images d'une même classe comme les motos ou les montres ou même encore les avions ont des couleurs différentes, des positions différentes dans l'image, des formes différentes. A cela s'ajoute l'arrière plan qui est différent d'une image à l'autre. De même pour les visages, la personne a posée à différents endroits dans l'image avec une luminosité différente il est donc difficile d'obtenir parfaitement une règle juste en regardant un histogramme de luminance avec un arbre de décision.

Commenté [8]: J'ai du 69.2% avec un treebag ou un RandomForest



Arbre élagué

Classification par le modèle bayésien

Nous obtenons les résultats suivants :

Modèles	Taux de précision
Bayésien	46.8%

Même réflexion ici que le classifieur précédent. Le résultat est médiocre ce qui est en partie la faute du descripteur.

Classification par SVM

Modèles	Taux de précision
	Meilleur résultat obtenu avec kernel : linéaire, C = 1e-04
SVM	54.8%

Le meilleur résultat est obtenu ici avec un kernel linéaire et un coût égal à $1e-04$. Même le SVM qui a montré des résultats corrects dans les 2 cas précédents se voit vaincu ici par l'inefficacité du descripteur.

Concaténation des histogrammes des composantes RGB

Pour ce descripteur nous disposons d'un ensemble de 500 données comme dans le cas précédent, seul les attributs changent. Comme son nom l'indique l'image est décrite par rapport aux 3 histogrammes R,G,B c'est la raison pour laquelle nous avons 256 valeurs x 3, soit 768 de nombres entiers.

Pour notre apprentissage, nous travaillons une nouvelle fois avec 65 % des données pour l'apprentissage et 35% pour notre validation sauf pour le bayésien. (Le même que les 2 cas précédent)

Classification par les arbres

Modèle	Taux de précision
Arbres	61.6%

L'idée est exactement la même ici, un histogramme de couleur est trop vague pour parfaitement décrire une image. Comme nous l'avons dit, excepté pour le léopard, les objets changent constamment de couleurs et d'environnements. Il est donc difficile de déterminer la classe d'appartenance.

Classification par le modèle bayésien

Modèles	Taux de précision
Bayésien	48%

Nous pouvons néanmoins noter que les résultats des 2 classifieurs est légèrement supérieur à l'autre descripteur.

Classification par SVM

Modèles	Taux de précision
	Meilleur résultat obtenu avec kernel : radial , C = 100, gamma=0.001
SVM	53.2%

Le meilleur résultat a été obtenu avec un SVM radial avec comme paramètres gamma égal à 0.001 et un coût de 100.

En résumé pour Caltech nous obtenons les résultats ci-dessous selon les différents classifieurs et les différents descripteurs :

Modèles	Taux de précision			
	Descripteur du patch	Descripteur du gist	Descripteur greyHisto	Descripteur ColorHisto
Bayésien	69.6%	80.8%	46.8%	48%
Les arbres	68.4%	71.2%	58.8%	61.6%
SVM	82%	82%	54.8%	53.2%

Comme on peut le voir, une majorité des résultats obtenus par le biais de ces différents classifieurs et descripteurs ne sont pas corrects. Cela est dû par le fait que les descripteurs analysent une image pixel par pixel ou encore par un outil peu représentatif des images. Si certaines corrélations constructives peuvent être établies entre les pixels afin d'établir des règles d'apprentissage, le résultat ne s'en verrait qu'amélioré.

Néanmoins avec les résultats suivants nous pouvons en conclure que le choix du classifieur est important dans le type d'apprentissage que nous souhaitons faire puisque ce dernier dépend grandement de la base de données que l'on lui attribue.

Jeu de données Skin Segmentation:

Le deuxième jeu de données que nous allons voir est SKIN SEGMENTATION. Celui-ci contient une base données conséquente de valeurs BGR classifiées comme étant de la peau ou non. Nous allons donc appliquer les mêmes méthodes que le jeu de données précédent et nous allons comparer les résultats.

Pour les tests nous avons séparé notre ensemble. Dans le cas du bayésien nous avons 65 % représentant l'ensemble d'apprentissage et 35 % représentant l'ensemble de test. En ce qui concerne l'arbre de décision nous avons séparé l'ensemble en 3. 55% pour l'apprentissage, 25% pour la validation et 20% pour le test.

Nous obtenons le résultat suivant :

Modèles	Taux de précision
Bayésien	92.5%
Arbre de décision	98.4%
SVM	92.9%

Le résultat est pleinement satisfaisant. Cela s'explique par le descripteur utilisé mais aussi par la quantité de donnée dont nous disposons. Ainsi le modèle décrit parfaitement ce que nous recherchons et il est aisé pour nos modèles de déterminer (ici par des probabilités) la classe d'un pixel donné.

Un SVM linéaire avec un coût de 0.01 a pas été appliqué ici car son temps d'exécution était bien trop long malgré une réduction de l'ensemble d'apprentissage.

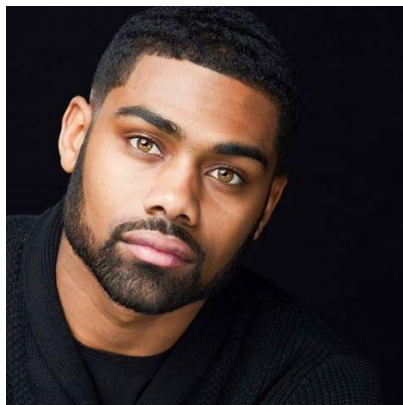


Bayésien



Arbre de décision

Ici l'arbre de décision est totalement induit en erreur par la couleur de cheveux de cette personne, il en est de même pour les sourcil.





Bayésien



Arbre de décision



SVM

On remarque ici quelques erreurs dû au reflet de la lumière sur le visage de cette personne. Malgré le taux de réussite supérieur pour la classification de l'arbre de décision, le rendu est meilleur pour le bayésien car le résultat est très uniforme même si l'arbre de décision semble détecter plus de peau que son homologue. Le SVM quant à lui dispose d'excellents résultats.





Bayésien



Arbre de décision

Le résultat reste assez surprenant, sur le papier l'arbre obtient de meilleurs résultats mais lorsqu'on l'applique à une image on se rend compte que le rendu est bien meilleur avec Bayes.

Conclusion:

La méthode de classification est étroitement liée au descripteur utilisé pour construire la base de donnée. Ainsi il est important de choisir correctement un descripteur suffisamment représentatifs de nos échantillons et de lui associer la bonne méthode de classification. Il est aussi important de ne pas se fier qu'au critère de précision, il est parfois préférable d'analyser des exemples concrets comme le montre la Skin segmentation. Au cours de ce TP nous n'avons pas implémenté les classifieurs les plus optimales comme le randomForest mais les résultats restent dans l'ensemble satisfaisant.