Fiche ML

Charles Vin

S2-2023

1 Généralité

— Fonction de perte : quantifie l'erreur associé à une décision. Erreur simple : A chaque fois qu'on se trompe, on compte 1 : 0-1 loss
— Risque : Proba de se tromper, $R(y_i|x) = \sum_j l(y_i, y_j)P(y_j|x)$

2 Arbre de décision

Algo général :

1. Déterminer la meilleure caractéristique dans l'ensemble de données d'entraînement.
2. Diviser les données d'entraînement en sous-ensembles contenant les valeurs possibles de la meilleure caractéristique.
3. Générez de manière récursive de nouveaux arbres de décision en utilisant les sous-ensembles de données créés.
4. Lorsqu'on ne peut plus classifier les données, on s'arrête.

Méthode de division des données : On vas utiliser l'entropie

Définition 2.1 (Entropie)**.** Origine de la formule de l'entropie Soit X une variable aléatoire pouvant prendre n valeurs x_i

$$H(X) = -\sum_{i=1}^{n} P(X = x_i)\log(P(X = x_i)).$$

Mesure l'homogénéité d'un dataset. C'est également la moyenne de la suprise (voir la vidéo)

Définition 2.2 (Gain d'information)**.** Mesure la réduction expects de l'entropie causé par le partitioning des exemples.

En faisant un test T sur un des attributs, on obtient deux partitions d'exemples de X : X_1 qui vérifie le test et X_2 qui ne vérifie pas le test (resp. Y_1 et Y_2).

$$H(Y|T) = \frac{|X_1|}{|X|}H(Y_1) + \frac{|X_2|}{|X|}H(Y_2).$$

Gain d'information :

$$I(T, Y) = H(Y) - H(Y|T).$$

On veut maximiser le gain d'information par le split $\Leftrightarrow$ minimiser $H(Y|T)$

3 Classfieur bayesien

On a :
— $P(y)$ fréquence des classe dans le dataset
— $P(x|y)$ les points de notre jeux de donnée. Graphiquement : les points coloriés
On cherche :

$$\arg\max_y P(y|x) = \arg\max_y \frac{P(x|y)P(y)}{P(x)}.$$

$P(x)$ difficile à calculer = répartition des points dans l'espace, dans le graph 2d non colorié. En général très petit, uniquement utile pour générer des données, pas pour faire l'argmax (aka classifier). Classifier bayésien = le classifier qui minimise le risque = le meilleurs classifieur possible

4 Estimation de densité

4.1 Par histogramme

Définition 4.1 (Estimation par histogramme)**.**
— Cas discret : Comptage dans chaque classe puis normalisation par le nombre d'exemple N
— Cas continue : Discrétisation des valeurs puis comptage et normalisation

Importance de la discrétisation :
— Petit $\rightarrow$ sur-apprentissage,
— Trop grand $\rightarrow$ sous-apprentissage
Limite :
— Grande dimention $\rightarrow$ Perte de sens
— Effet de bord : petit changement dans les bins, gros changement d'estimation.
$\rightarrow$ Solution : Estimation par noyaux

4.2 Estimation de densité par noyaux

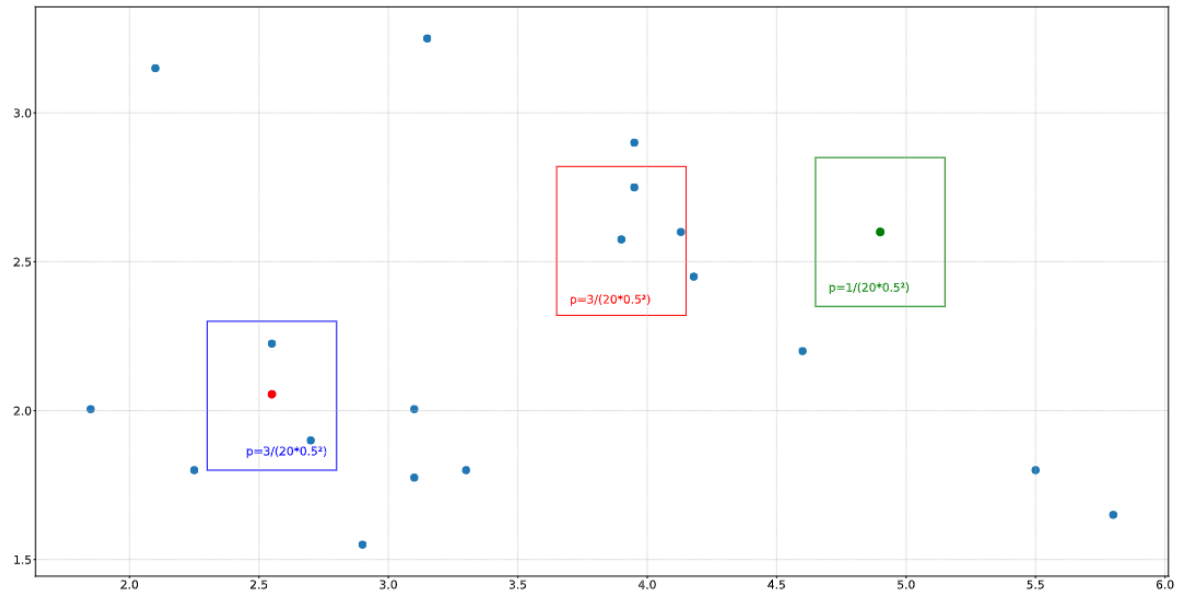

Figure 1 – Intuition de l'estimation par noyaux

Intuition figure 1 : Plutôt que de décider d'une discrétisation a priori, l'estimation est faîte en centrant une fenêtre autour du point d'intérêt x_0 (dans un espace de dimension d). $\rightarrow$ Problème : pas continue (si on bouge la boite et qu'un point rentre dedans, ça fait faire un saut à la fonction)

4.2.1 Fenêtre de Parzen

On combine la solution précédente avec une densité/noyaux. Classiquement Gaussien. pour obtenir un truc lisse et continue

Définition 4.2 (Fenêtre de Parzen)**.** Soit $(x_1, \ldots, x_N) \sim f$ iid

$$\hat{f}_h(x) = \frac{1}{N * h} \sum_{i=1}^{N} K(\frac{x - x_i}{h}).$$

Avec K le noyaux **centrée et réduit sur** x , souvent une fonction gaussienne. Si c'est une fonction rectangle ça fonctionne aussi. Puis y'a plein d'autre noyaux possible.