Introduction à la science des données
par Loïc Herman, page 1 de 1

1 LVQ

1.1 Fonctionnement

Algorithme 1 : Vecteurs prototypes

Initialiser les vecteurs prototypes `// Il faut au moins un prototype par classe`
Répéter nombre_de_epochs fois :
 Pour toute la base de données d'entraînement :
 – choisir un point aléatoirement
 – trouver le prototype le plus proche (best matching unit ou bmu)
 – sils ont la même classe, rapprocher le prototype de la donnée exemple
 – sinon, éloigner le prototype
 -> diminuer le facteur de rapprochement/éloignement
Prédire la classe des nouvelles observations en trouvant le plus proche prototype

1.2 Hyper-paramètres

Split train-test : part des données à séparer pour l'apprentissage des prototypes
Nombre de prototypes : nombre de prototypes à placer aléatoirement. À ajuster en fonction du nombre de clusters.
Learning rate : $]0,1[$, rate proportionel à la distance de déplacement des prototypes. Conditionne la vitesse d'apprentissage. Trop grand et il y a un risque de diverger, trop petit et cela prendra trop de temps à converger à une valeur satisfaisante.
Nombre d'épochs nombre de fois où nous effectuons le raffinement des prototypes pour toute la base de données d'entraînement :

- le nombre depochs est figé et déterminé de manière expérimentale
- répéter jusquà ce que l'erreur MSE soit plus petite quun seuil déterminé par expérience
- répéter jusquà ce quon ne voit plus d'amélioration conséquente

k-nn : une fois les prototypes détérminés, donne combien de prototypes alentour doivent être considérés.

1.3 Matrice de confusion

		Prédit	
Réel		TP	FN
		FP	TN

accuracy : $\frac{TP+TN}{TP+FN+FP+TN}$
precision : $\frac{TP}{TP+FP}$
recall : $\frac{TP}{TP+FN}$
f-score : $\frac{2TP}{2TP+FP+FN}$

2 Régression linéaire

2.1 Hyper-paramètres

learning rate (lr) : Conditionne la vitesse dapprentissage, S'il est trop grand, il est possible d'osciller autour du minimum, voire de diverger. S'il est trop petit, le temps mis pour arriver au minimum peut grandement augmenter.
Nombre depochs : Nombre ditérations ńădajustementsăž à faire avant de sarrêter. Peut être déterminé de plusieurs manièresă:

- Nombre figé et déterminé de manière expérimentale.
- Jusqu'à ce que l'erreur MSE soit plus petite qu'un seuil déterminé par expérience.
- Jusqu'à ce qu'on ne voie plus d'amélioration conséquente

2.2 Marche à suivre

1. m et b initialisé aléatoirement
2. calcul de l'erreur
$$E = \frac{1}{2N}\sum (y_i - \hat{y}_i)^2$$
$$= \frac{1}{2N}\sum (y_i - (mx_i + b))^2$$
3. m et b mis à jour
$$m = m - lr \cdot \frac{\partial E}{\partial m}$$
$$b = b - lr \cdot \frac{\partial E}{\partial b}$$
4. Répéter autant de fois que n_epochs

$$\frac{\partial E}{\partial m} = -\frac{1}{N}\sum_{i=1\ldots N} [x_i \cdot (y_i - (mx_i + b))]$$
$$\frac{\partial E}{\partial b} = -\frac{1}{N}\sum_{i=1\ldots N} (y_i - (mx_i + b))$$

2.3 Par covariance

$$m = \frac{cov_{x,y}}{var_x}$$
$$b = \overline{y} - m\overline{x}$$

2.4 Descente stochastique

Estimation du gradient sur la base d'une donnée d'entrainement choisie au hasard, au lieu de toutes les données du train set.
Avantage : Permet le traitement de très grand ensemble de données
Avantage : Permet un apprentissage progressif, une adaptation à la volée du modèle sur des nouvelles données récoltées
Désavantage : Ne converge jamais vraiment comme la descente de gradient, mais finit par errer proche du minimum

2.5 Descente par batch

Compromis entre la descente de gradient ń complète ż et la descente stochastique. Gradient calculé à partir d'un échantillon de M observations du train set, composé aléatoirement.
Avantage : Beaucoup plus stable que la descente stochastique.
Désavantage : Même désavantage que la descente stochastique

2.6 Évaluation de la régression linéaire

$\hat{y}_i$ est la prédiction pour y_i.
$\overline{y}$ est la moyenne arithmétique des valeurs y_i.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

Erreur

$$E = \frac{1}{2n}\sum_{i\in n}(y_i - (mx_i + b))^2$$

x_i valeur de x
y_i valeur de y
m valeur actuelle de la pente
b valeur actuelle de l'abscisse
n nombre d'observations

3 Données et caractéristiques

Variable continue : Peut supposer un nombre infini de valeurs réelle. Valeurs mesurées et non comptées.
Variable discrète : Ne peut revêtir qu'un nombre fini de valeurs réelles. Valeur issue d'un comptage.
Variable numérique (variable quantitative) : Peut supposer un nombre infini de nombres. Ex : nombre de pas / jour
Variable catégorielle (ou variable qualitative) : Chaque valeur peut être classée dans une catégorie particulière. Catégories devant être mutuellement exclusives et exhaustives. Ex : terrien, extraterrestre OU animal, végétal, minéral
Donnée nominale : Exprime des identifications (objet) ou des appartenances à des catégories (classe d'objets). Ex : objet (l'Amazone), classe (zone urbaine, agricole, forêt)
Donnée ordinale : Exprime un rang dans une classe d'objets, une qualité dans un énoncé hiérarchique tel que petit, moyen, grand. L'expression numérique de ces propriétés qualitatives est relative à l'intérieur d'un intervalle de valeurs défini arbitrairement. p.ex. petit=0, moyen=1, grand=2.
Donnée cardinale (ou d'intervalle-rapport) : Ajoute une notion de distance entre les propriétés ou valeurs de l'attribut exprimées de manière quantitative. C'est la catégorie la plus courante de la mesure d'un paramètre physique tel que, notamment, la longueur, la tension électrique, l'intensité lumineuse, la température.
Données séquentielles (ou séries temporelles) : Suite de valeurs numériques représentant l'évolution d'une quantité spécifique au cours du temps.
Scalaire : valeur qui n'est spécifiée que par sa grandeur. Il s'agit d'une mesure physique ou de la valeur d'une variable. Ex : x = 42
Array : Structure de données qui peut contenir un nombre fixe d'éléments du même type. Ex : Vecteur en physique (V = [0.25, -0.72, 9.8])
Matrice : Structure de données bidimensionnelle (2D) dans laquelle les nombres sont organisés en lignes et en colonnes. Ex : M = [[1,2], [3,4]]
Tenseur : Objet mathématique très général. En fait, il peut-être un scalaire (d'ordre 0), ou un vecteur (ordre 1), une matrice (ordre 2), mais en général, il peut être d'ordre > 2. Pour nous, ça sera une structure de données de 3 dimensions ou plus.
Données structurées : Données hautement organisées et soigneusement formatées. C'est le type de données que l'on peut mettre dans des tableaux, des feuilles de calcul, des bases de données.
Données non structurées : Données représentées ou stockées sans format prédéfini. La collecte, le traitement et l'analyse de données non structurées nécessitent davantage de travail. Ex : des informations sur des pages web (nom, occupation, e-mail, etc.)
Données semi-structurées : Données contenant des éléments qui facilitent la séparation des champs et des enregistrements. Ex : JSON
Donnée manquante : Pas de valeur. Ex : {human 1 : 1.30, human 2 : 1.75, human 3 : 1.68, alien :...
Donnée inexacte : (vous vous débrouillez pour avoir une valeur). Ex :. {human 1 : 1.30, human 2 : 1.75, human 3 : 1.68, alien : 1.70.}
Donnée tronquée : Valeur ignorée. Ex : {human 1 :1.30, human 2 :1.75, human 3 :1.68}
Donnée censurée : (au lieu de la valeur exacte, on trouve une plage de valeurs). Ex : (human 1 : 1.30, human 2 : 1.75, human 3 : 1.68, alien : >160.)
Il existe des nombreuses bases de données pour développer et évaluer les algorithmes d'analyse et modélisation des données. MAIS, même si les données abondent aujourd'hui, il existe toujours un problème pour obtenir des "données étiquetées" et des données de qualité. On peut avoir le meilleur des algorithmes, si les données sont mauvaises le résultat le sera aussi (garbage in, garbage out).

4 k-means

4.1 Marche à suivre

4.1.1 k-means

1. Soient k groupes de données (clusters) et les prototypes μ_i des clusters à partir de k observations aléatoires
2. Classifier chaque donnée en entrée de la base de données (x_j) en cherchant le prototype le plus proche
3. Actualiser les centres des clusters
$$\mu_i = \overline{x_j}\,, \quad x_j \in N_i$$
4. Répétez les étapes 2 et 3 jusqu'à ce que les étiquettes associées à toutes les données d'entrée de la base de données ne changent plus.

4.1.2 k-means++

1. Soit un centre placé au hasard
2. $\forall x_i \in N_j\,;\, D(x_i) = \left\|\overrightarrow{d(x_i,\mu_j)}\right\|$
3. Choisir un nouveau μ_j aléatoirement pondéré pour minimiser la distance
4. Répétez les étapes 2 et 3 jusqu'à ce que k centres aient été choisis
5. Une fois prototypes choisis, passer au groupement par k-means.

4.2 Within-cluster Sum of Squared Errors (WSS)

$$WSS_j = \sum_{i\in N_j}\left\|\overrightarrow{d(x_i,\mu_j)}\right\|^2$$

x_i valeur d'un point du cluster j
μ_j centre du cluster j
N_j nombre d'observations du cluster j

5 Calculs statistiques

5.1 Variance

$$S^2 = \frac{1}{n-1}\sum_{i\in n}(x_i - \overline{x})^2$$

x_i valeur d'une observation de x
$\overline{x}$ moyenne de toutes les observations de x
n nombre d'observations

5.2 Covariance

$$cov_{x,y} = \frac{1}{N-1}\sum_{i\in N}(x_i - \overline{x})(y_i - \overline{y})$$

x_i valeur de x
y_i valeur de y
$\overline{x}$ moyenne de x
$\overline{y}$ moyenne de y
N nombre de valeurs

5.3 Distance Euclidienne

$$\left\|\overrightarrow{d(x,y)}\right\| = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}$$