TP 2 - Estimation de densité, K plus proches voisins

1 Données : velib

Télécharger l'archive jointe au sujet et le fichier code. L'archive contient 4 semaines de logs de 1217 stations velib de Paris à partir du 1 octobre 2016 et des informations sur les stations [1]. Le code permet de lire l'archive et d'importer les objets suivants :

- `stations` : un dictionnaire où chaque clé correspond à un identifiant station velib `idvelib` et chaque valeur à un n-uplet : (nom station, adresse station, x,y, nombre d'attaches, longitude, latitude) ; (x,y) correspond à une localisation normalisée entre 0 et 1 ; le nombre d'attaches correspond au nombre maximal
- `histo` : une matrice de taille 1217×43200, chaque ligne correspond à une station, chaque colonne au nombre de vélos disponible par minute pour la station correspondante ;
- `stations_idx` : correspondance entre le `idvelib` et l'indice de la ligne de la matrice correspondante ;
- `idx_stations` : correspondance entre la ligne de la matrice et le `idvelib`

La commande `histo[10,:]` permet de récupérer l'historique de la station 10 sur toute la durée ($60 * 24 * 28$ minutes), `histo[10,1440:2880]` permet de récupérer l'historique pour le 2ème jour. Les informations sur la station 10 sont données par `stations[idx_stations[10]]`. Vous trouverez dans le code les lignes nécessaires à l'affichage de la carte de Paris et des stations.

2 Estimation de densité

L'objectif de cette partie est d'estimer la densité spatiale de la répartition de l'offre des vélibs sur le territoire, notée $p_s(x, y)$ avec $(x, y) \in [0, 1]^2$. Vous allez pour cela étudier deux méthodes, la méthode des histogrammes et la fenêtre de Parzen.

2.1 Histogramme

Soit un échantillon $\mathcal{X} = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n\} \subset \mathbb{R}^d$, l'objectif de la méthode des histogrammes est de calculer une densité constante par morceaux $p(\mathbf{x})$ sur $\mathbb{R}^d$ reflétant la répartition de l'échantillon. Pour cela, une discrétisation de l'espace en hypercube est considérée et la densité est estimée à partir du comptage du nombre d'exemples tombant dans chaque hypercube rapporté au nombre total d'exemples multipié par le volulme de l'hypercube. Il est possible de pondéré les exemples par un poids $w^i \in \mathbb{R}^+$ (quel est l'intérêt ?) Pour une partition $\{C_i\}_{i=1}^N$ de l'espace, la densité estimée pour un point $\mathbf{x}$ de la partition $C_{i_\mathbf{x}}$ est

$$p(\mathbf{x}) = \frac{\sum_{i=1}^n w^i \mathbf{1}_{\mathbf{x}^i \in C_{i_\mathbf{x}}}}{V_{C_{i_\mathbf{x}}} \sum_{i=1}^n w^i}$$

avec $V_{C_{i_\mathbf{x}}}$ le volume de la partition.

Q 2.1 La matrice `geo_data` contient une station par ligne et 3 colonnes : les coordonnées x, y et le nombre d'attaches (nombre de vélos théoriquement disponibles). Afficher les stations et la carte de Paris en arrière plan à l'aide du code fourni.

Q 2.2 Calculer l'histogramme de $p_s(x, y)$ qui correspond à la densité spatiale de l'offre vélib théorique en utilisant une grille de discrétisation de 10×10. Afficher le résultat en utilisant la commande

1. Ces données sont disponibles sur `https://developer.jcdecaux.com/#/home`

`plt.imshow(resultat,extent=[0,1,0,1],interpolation='none',alpha=0.3,origin = "lower",aspect=0.6)` (faire attention de prendre la transposée de la matrice `resultat` selon le sens de calcul de l'histogramme). . Comparer à la répartition des stations vélibs. Réitérer sur plusieurs autres discrétisations spatiales.

Q 2.3 Que remarquez vous par rapport à la stabilité des résultats ? De combien de paramètres est composé le modèle ? Comment évolue ce nombre de paramètres en fonction du nombre de dimension ? du pas de discrétisation ? Conclure sur les avantages et faiblesses de la méthode.

2.2 Fenêtres de Parzen

Les fenêtres de Parzen permettent d'éliminer les effets de bord de la méthode des histogrammes : une fenêtre (un hypercube) centrée autour du point à estimer est utilisée plutôt que des hypercubes statiques. Les dimensions de la fenêtre sont précisés par un paramètre `sigma`. L'estimation est calculée en comptant le nombre de points présents dans la fenêtre, possiblement pondéré en utilisant un noyau en fonction de la distance au centre de la fenêtre - le point à estimer.
La formule générale pour des données en dimension d, un paramètre de bande σ et des poids w^i est $p(\mathbf{x}) = \frac{1}{\sigma^d \sum_{i=1}^n w^i} \sum_{i=1}^n w^i K(\frac{\mathbf{x}-\mathbf{x}^i}{\sigma})$ et $K(\mathbf{x})$ un noyau de convolution [2].
En utilisant un noyau uniforme : $K_u(\mathbf{x}) = \begin{cases} 1 & \text{si } \forall j,\ |x_j| \leq 0.5 \\ 0 & \text{sinon} \end{cases}$, la méthode est équivalente à celles des histogrammes en utilisant un pas σ de discrétisation, mais avec un hypercube centré sur le point à estimer. Le noyau gaussien $K_g(\mathbf{x}) = \frac{e^{-0.5\|\mathbf{x}\|^2}}{(2\pi)^{d/2}}$ est le noyau le plus utilisé en pratique.

Q 2.4 Implémenter les fonctions `uni(x,data,sigma)` et `gaussian(x,data,sigma)` qui prennent en entrée une donnée `x`, une matrice `data` de taille $n \times d$ d'exemples et le paramètre `sigma` et renvoient un vecteur de taille n, la valeur du noyau uniforme/gaussien en $K\left(\frac{\mathbf{x}-\text{data[i]}}{\sigma}\right)$ (sans boucle `for`).

Q 2.5 Implémenter la fonction `parzen(x,data,weight,sigma,kernel)` qui renvoie l'estimation de la densité au point `x` de dimension d pour un échantillon `data` de taille $n \times d$ pondéré par le vecteur `weight` de taille n en utilisant la fonction noyau `kernel`.

Q 2.6 Expérimenter les deux noyaux. Faites varier le paramètre. Que remarquez vous quand celui-ci tend vers 0 ? vers l'infini ?

Q 2.7 Pourquoi les moindres carrés sont difficiles à appliquer pour évaluer l'estimation ? Proposer une mesure d'évaluation basée sur la vraisemblance. Comparer les résultats des différents noyaux et des différents paramétrages.

Estimateur de Nadaraya-Watson et K plus proches voisins

L'estimateur de Nadaraya-Watson permet d'adapter les fenêtres de Parzen à la régression locale. Soit $f : \mathbb{R}^d \to \mathbb{R}$ fonction objectif, l'estimateur $\hat{f}$ sur un échantillon $\{(\mathbf{x}^i, y^i = f(\mathbf{x}^i))\}_{i=1}^n$ est défini par $\hat{f}(\mathbf{x}) = \frac{\sum_{i=1}^n y^i K(\frac{\mathbf{x}-\mathbf{x}^i}{\sigma})}{\sum_{i=1}^n K(\frac{\mathbf{x}-\mathbf{x}^i}{\sigma})}$.
Dans les questions suivantes, on veut modéliser la demande en vélib en fonction du temps, tout d'abord en se focalisant sur une station, puis sur le territoire.

Q 2.8 Construire la matrice `take` contenant le nombre de vélos empruntés aux stations par minute. Tracer pour quelques journées et pour quelques stations le nombre d'emprunts par minute. Voyez-vous une différence entre stations ? entre journées ? entre jour ouvert et week-end ? Quelle(s) hypothèse(s) de régularité peut-on faire pour modéliser la demande ?

2. $K(x) \geq 0,\ \int K(x) = 1$ en particulier.

Q 2.9 Implémenter l'estimateur de Nadaraya-Watson. Expérimenter sur la demande pour une station sur une journée, puis en agrégeant plusieurs journées de la même station. Observer l'évolution en fonction de l'ajout de données. Interpréter le rôle de σ et le choix du noyau.

Q 2.10 Comme souvent en régression, les moindres carrés sont utilisés pour l'évaluation des modèles. Que doit-on considérer comme découpage des données en ensemble apprentissage et test afin de pouvoir faire de la sélection de modèles ? Expérimenter afin de déterminer le meilleur paramètre. Est-il constant sur chaque station ? Pour chaque jour ?

Q 2.11 Est-il possible d'utiliser le même algorithme pour prédire $u(x, y, t)$ la demande en un point (x, y) à un instant t ? Quel problème peut-il se poser dans les régions désertiques en stations ?
Une approche très similaire sont les k plus proches voisins. Au lieu de pénaliser les exemples par une distance au point $\mathbf{x}$ à estimer, cette approche consiste à moyenner les valeurs des k échantillons les plus proches de $\mathbf{x}$: $\hat{f}(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{N} y^i \mathbf{1}_{x^i \in N_k(\mathbf{x})}$ avec $N_k(\mathbf{x})$ les k plus proches voisins de $\mathbf{x}$. Implémenter et comparer les résultats pour l'estimation de $u(x, y, t)$.

3 Bonus : pour aller plus loin . . . K-means

L'algorithme K-moyenne est un des algorithme d'apprentissage non supervisé des plus utilisés. Le principe est de trouver une partition de l'espace d'entrée (des clusters) en considérant la densité d'exemples pour caractériser ces partitions. Un cluster C_i correspond à la donnée d'un prototype $\mu_i \in \mathbb{R}^d$ dans l'espace d'entrée. Chaque échantillon $\mathbf{x}$ est affecté au cluster le plus proche selon une distance (euclidienne ou autre) entre l'exemple et le prototype du cluster. Soit $s_C : \mathbb{R} \to \mathbb{N}$ la fonction d'affectation associée au clustering $C = \{C_1, C_2, \ldots, C_k\}$: $s_C(\mathbf{x}) = argmin_i \|\mu_i - \mathbf{x}\|^2$. La fonction de coût sur un ensemble de données $\{x_1, \ldots, x_n\}$ considérée dans ce cadre est la moyenne des distances intra-clusters : $\frac{1}{n}\sum_{i=1}^{n}\sum_{j|s_C(\mathbf{x}^j)=i} \|\mu_i - \mathbf{x}^j\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\mu_{s_C(\mathbf{x}^i)} - \mathbf{x}^i\|^2$. C'est également ce qu'on appelle le coût de reconstruction : effectivement, dans le cadre de cette approche, chaque donnée d'entrée peut être "représentée" par le prototype associé : on réalise ainsi une compression de l'information. L'algorithme fonctionne en deux étapes, (la généralisation de cet algorithme est appelée algorithme E-M, Expectation-Maximization) :

- à partir d'un clustering C^t, les prototypes $\mu_i^t = \frac{1}{|C_i|}\sum_{x^j \in C_i^t} x^j$, barycentres des exemples affectés à ce cluster ;
- à partir de ces nouveaux barycentres, calculer la nouvelle affectation (le prototype le plus proche).

Ces deux étapes sont alternées jusqu'à stabilisation des centres des clusters.

Q 3.1 On souhaite opérer un clustering des stations en considérant comme description d'une station la demande par unité de temps en vélib. Quelle distance considérée ? Faut-il normaliser ? Proposer et implémenter un protocole expérimental. Faites varier le nombre de clusters et comparer les résultats. Comment évaluer le résultat de votre clustering ?