Polytech Paris-Sud 5e année - Big Data

C. Barras - Y. Matveeva

Projet de reconnaissance de chiffres manuscrits - 2ème partie

Le but de ce projet, à réaliser en binôme, est de mettre en place, d'évaluer et de comparer des systèmes de reconnaissance automatique. Afin d'atteindre ces buts on vous propose de mener plusieurs expérimentations sur un corpus de chiffres manuscrits en Python (le corpus reste le même que pour la première partie du projet).

Travail à réaliser

Deuxième partie (séances 3 et 4)

Description de la tâche

Dans une deuxième phase, on vous demande de tester des classifieurs plus performants, de les comparer en évaluant leur performance sur le corpus de développement aussi bien qu'en analysant (pour certains classifieurs) leur comportement vis à vis de différentes transformations de données d'entrées.

Vous n'avez pas à implémenter les phases d'entraînement et de prédiction de ces classifieurs, vous utiliserez les implémentations déjà présentes dans la librairie **scikit-learn** . Vous trouverez des tutoriels pour vous aider sur le site officielle de cette librairie : `http://scikit-learn.org/stable`. À noter : vous y trouverez des descriptions succinctes des algorithmes aussi.

Voici les tâches à accomplir dans cette partie du projet :

1. Tester des différents hyper-paramètres d'un classifieur de choix ou comparer plusieurs classifieurs (autres que ceux implémentés dans la partie 1 de ce projet). Vous avez quelques propositions de sujets de cette étape décrites en bas. Quoique sera votre choix vous en informerez l'encadrant (par mail) au plus tard **dimanche 27 novembre 2016**.
2. Choisir le meilleur système de prédiction selon le taux d'erreur global.
3. Implémenter une méthode d'évaluation plus fine des résultats de prédiction des classifieurs. En plus du taux d'erreur global, vous calculerez pour le meilleur système obtenu une matrice de confusion (avec en ligne la classe réelle de chaque exemple et en colonne la classe reconnue par le système). [1] À partir de cette matrice vous calculerez également la précision et le rappel pour chaque classe.

Vous pouvez bien sûr (par curiosité et pour ajouter du contenu dans votre rapport) utiliser la méthode d'évaluation avec les matrices de confusion pour faire une comparaison plus profonde des systèmes que vous avez choisi.

Propositions de sujets Un des buts principaux de la plupart de ces sujets est de mieux comprendre les présuppositions des modèles des différents classifieurs et de comprendre les relations de différents classifieurs entre eux.

Supervisé :

1. Classifier Bayésien Naïf (gaussienne ou multinomiale) :
   (a) Choix de variables. [2]
   - Quelles sont les variables qui sont les moins utiles pour tout classifieur (pas seulement Naïf Bayes) ? Retrouvez-vous de telles variables dans vos données ?
   - En se basant sur les présuppositions du modèle de Bayes Naïf, pouvez-vous proposer des transformations de données simples (de type 'réduction de dimensionnalité' autre que ACP) permettant d'améliorer la performance de l'algortihme ?
   - Comparer les résultats sans et avec votre variante de réduction de dimensionnalité.

1. Étant donné en entrée deux vecteurs (un pour les valeurs réelles et un pour les valeurs prédites), il existe une solution élégante (mais pas évidente) de vectorisation du calcul de la matrice de confusion. Cela sera un (relativement) grand bonus pour vous d'implémenter cette version vectorisée. Vous pouvez utiliser la fonction **numpy.bincount** pour cela. Elle ne prend qu'un seul vecteur unidimensionnel de valeurs numériques entières en entrée, donc il faudra réfléchir à une transformation de vos données d'entrée. La différence de temps de calcul est impressionnante : 3.2e-05 sec pour la version vectorisée et 3e-03 sec pour la version qui utilise une boucle simple (100 fois plus). Longueur de vecteurs d'entrée = 5000. D'autres solutions (sans **numpy.bincount**) qui pourraient se rapprocher du premier temps de calcul sont bienvenues.

2. Fonction utile : **numpy.where** .

(b) Comparer les résultats du classifieur sans ACP et avec ACP.
— Pouvez vous expliquer les résultats de comparaison (en se basant sur les présuppositions du modèle du classifieur et les propriétés inhérentes à l'ACP) ?
— Commenter la courbe des taux d'erreurs avec ACP.

(c) Comparer la version gaussienne avec le classifieur de distance minimum (distance euclidienne pondérée par des inverses de variances $\frac{1}{\sigma_i^2}$ (Mahalanobis diagonale)).

2. Classifieur Bayésien Gaussien Non-naïf (sans présuppositions sur les relations entre variables), i.e. "Linear / Quadratic Discriminant Analysis" :
   (a) Comparer les taux d'erreurs de LDA et de QDA.
   (b) Comparer cette variante générale avec la variante naïve ("Gaussian Naïve Bayes").
   (c) Comparer avec le classifieur de distance minimum (distance de Mahalanobis).
   N.B. : Pensez au choix de variables si vous n'utilisez pas ACP (1a).
3. Classifieur de k plus proches voisins :
   (a) Tester de différentes mesures de distance.
   (b) Tester de différentes façons de pondérer les points du voisinage.
   (c) Tester de différentes valeurs de k (en commençant par k=1 de la première partie).
   (d) Ce classifieur peut être vu comme un classifieur bayésien non-paramétrique. Comparer ce classifieur avec la distance euclidienne avec le classifieur bayésien gaussien.
   (e)* Des variantes optimisées de l'implémentation de l'algorithme.
   — Comparer les temps de calcul pour une (des) variante(s) optimisée(s) de l'implémentation de l'algorithme avec une variante non-optimisée. [3]
   — Comment est-ce que le temps de calcul se compare au temps de calcul pour les autres classifieurs que vous avez implémenté / testé dans ce projet ?
4.* Classifieur SVM (C-SVC) [4] :
   (a) (?) Comparer SVM linéaire avec le perceptron multiclasse.
   (b) Tester des différents noyaux pour SVM et comparer la performance.

Non-supervisé :

1. Modèle de Mélange des Gaussiens (algorithme EM)
   — tester avec plusieurs nombres de classes, choisir le nombre de classes avec le critère BIC ; est-ce que le choix est égale au nombre de classes attendu ?
   — tester plusieurs méthodes d'initialiser l'algorithme : multistart randomisé, k-moyennes
   (a) *non-supervisé vs. supervisé* : comparer avec le classifieur Bayésien Gaussien (non-naïf supervisé) ;
   comparaison détaillée (voir étape 3 de la deuxième partie du projet) avec les valeurs réelles aussi bien que des systèmes d'apprentissage entre eux ;
   (b) *non-supervisé vs. non-supervisé* : comparer avec K-Moyennes (non-supervisé) [5] :
   comparaison détaillée (comme en 1a).

Le rendu

Dans le compte rendu de cette partie vous décrirez la démarche que vous avez poursuivi : quels systèmes d'apprentissage automatique avez vous comparé, comment, quel était le résultat. Quelles conclusions avez-vous tirée de cette comparaison ? Vous pouvez commenter l'efficacité en termes de temps de calcul / de la complexité aussi bien que les taux d'erreurs. Pour chaque algorithme de classification choisi vous fournirez aussi une description brève (pas plus d'un paragraphe) en vos propres mots du fonctionnement de l'algorithme et de son modèle sous-jacent.

Vous pouvez reprendre le rapport de la première partie en l'étandant et éventuellement le modifiant ou fournir un rapport séparé pour la deuxième partie.

Le compte-rendu est à rendre avec les sources commentés dans l'espace 'Travaux' du cours Et5BigData pour le **15 décembre 2016**.

3. Il va falloir comprendre au moins l'idée générale de l'optimisation et la décrire dans le rapport.
4. Vu que SVM n'était discuté que brièvement en cours, il est possible qu'il vous faudra de la lecture supplémentaire pour bien assimiler (et expliquer dans le rapport) le fonctionnement de ce classifieur (sans et avec noyaux). Qu'est-ce que fait le classifieur SVM linéaire ? Qu'est-ce que c'est qu'un noyau ? Pourquoi on l'utilise ?
5. En effet, l'algorithme K-Moyennes est une variante limite de l'algorithme EM de séparation de mélanges de gaussiennes quand les matrices de covariance de gaussiennes considérées sont toutes diagonales, et leurs variances tendent vers 0.

Annexe

0.1 ACP : son cadre et ces objectifs

$X \in \mathbb{R}^{n \times d}$: matrice des observations (une ligne = une observation, une colonne = un trait (un descripteur des observations)).

Supposons que X est centralisé (i.e. moyennes de lignes = 0).

But : obtenir une factorization $U = XP$, $P \in \mathbb{R}^{d \times p}, p \leq d$.

Les propriétés à assurer :

1. Obtenir de nouvelles variables ($\{U_{\cdot j}\}$) et réduire la dimensionnalité ($p \leq d$).
2. Obtenir ces variables comme des transformations **linéaires** des variables d'origine.
3. [IMPORTANT]
   Faire en sorte que les nouvelles variables soient **orthogonales** (matrice de covariance — diagonale).
4. Minimiser la "perte" des "informations" en réduisant la dimensionnalité.
   On maximise la somme de variances de variables $\sum_{j=1}^{p} \|U_{\cdot j}\|^2$.
5. Minimiser la défiguration de distances euclidiennes entre les observations (i.e. les lignes)

$$\sum_i \sum_j \left(D_{euc}(X_{i\cdot}, X_{j\cdot}) - D_{euc}(U_{i\cdot}, U_{j\cdot})\right) \to \min$$

.

Solution : ACP.

$\{P_{\cdot j_s}\}_{s=1}^{p^*}$: une base orthonormale de l'espace vectoriel $\mathbb{X}$, l'espace $\mathbb{X}$ étant constitué de toutes les combinaisons linéaires de vecteurs $\{X_{\cdot j}\}_{j=1}^{d}$ (donc de traits).

On passe d'une base classique (constituée de vecteurs-indicateurs avec toutes les valeurs étant nulles sauf une position) à la base $\{P_{\cdot j_s}\}_{s=1}^{p^*}$. On obtient ainsi de nouvelles variables $\{U_j\}$.

Solution algorithmique : calcul de vecteurs propres.

$\{P_{\cdot j}\}_{j=1}^{d}$ = les vecteurs propres de la matrice $C \in \mathbb{R}^{d \times d}$ de covariance (empirique) de X. Chaque vecteur correspond à une valeur propre $\lambda_j \geq 0 : CP_{\cdot j} = \lambda_j P_{\cdot j}$.

$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d \geq 0$

$p^* = rank(X), p^* \leq d : p^*$ = le nombre de vecteurs propres $\{P_{\cdot j_s}, s \in 1 : p^*\}$ ($\{j_s\}_{s=1}^{p^*} \subseteq \{1, 2, \ldots d\}$) correspondant à des valeurs propres $\{\lambda_{\cdot j_s}\}_{s=1}^{p^*}$ différentes non-nulles.

$\{P_{\cdot j_s}\}_{s=1}^{p^*}$ est un sous-ensemble de $\{P_{\cdot j}\}_{j=1}^{d}$.

$\{P_{\cdot j_s}\}_{s=1}^{p^*}$ est orthonormale : $\|P_{\cdot j_s}\| = 1, \forall s \neq t \, \langle P_{\cdot j_s}, P_{\cdot j_t} \rangle = 0$.

Les vecteurs $\{U_{\cdot j_s}\}_{s=1}^{p^*}$ correspondants sont orthogonales aussi.

Important : On ne peut pas obtenir plus de p^* nouvelles variables orthogonales.

Leurs variances : $\|U_{\cdot j_s}\|^2 = \lambda_{j_s}$.

$\sum_j \lambda_j = \sum_j \|X_{\cdot j}\|^2$.

On peut prendre moins de variables $p < p^*$. Si on prend les variables correspondants aux valeurs propres les plus grandes, on assure la propriété 4.

0.2 Orthogonalité versus indépendance de variables aléatoires

$x_1, \ldots, x_d \in \mathbb{R}^d$

Orthogonalité = indépendance **linéaire** :

$$\forall \alpha_1, \ldots, \alpha_d \in \mathbb{R} \; \sum_j \alpha_j x_j \neq 0.$$

Indépendance (mutuelle) de variables aléatoires = possibilité de factoriser la fonction de distribution jointe :

$$p(x_1, \ldots, x_d) = p(x_1)p(x_2)\ldots p(x_d) \tag{1}$$

Indépendance plus forte que orthogonalité : dépendances non-linéaires $\Rightarrow$ possibilité d'orthogonalité en absence d'indépendance au sens 1.

Pour la distribution normale (gaussienne) : Orthogonalité $\Rightarrow$ Indépendance.

0.3 Classifieur bayésien

Approche bayésienne : estimer les paramètres en maximisant la probabilité a postériori :

$$y^* = \arg\max_y p_\theta(y \mid x_1, \ldots, x_d) = \arg\max \frac{P(y)p(x_1, \ldots, x_d \mid y)}{p(x_1, \ldots, x_d)} \tag{2}$$

Observons que

$$\max_y (\alpha f(y) + \beta) = \max_y f(y), \tag{3}$$
$$\alpha \text{ et } \beta \text{ ne dépendent pas de y.}$$

Donc

$$y^* = \arg\max_y P(y)p(x_1, \ldots, x_d \mid y). \tag{4}$$

$$\arg\max_y \log(f(y)) = \arg\max_y f(y), \tag{5}$$
$$\text{car } \log(\cdot) \text{ est monotone.}$$

Donc

$$y^* = \arg\max_y (\log P(y) + \log p(x_1, \ldots, x_d \mid y)). \tag{6}$$

Classifieur bayésien naïf : présupposition d'indépendance conditionnelle.

$$p(x_1, \ldots, x_d \mid y) = \prod_{i=1}^{n} p(x_i \mid y) \tag{7}$$

$$y^* = \arg\max_y \left(\log P(y) + \sum_{i=1}^{d} \log p(x_i \mid y) \right). \tag{8}$$

0.3.1 Classifieurs bayésiens gaussiens

Présupposition du modèle : $p(x_1, \ldots, x_d \mid y = y_k) \sim N(\mu_k, C_k)$.
Distribution gausienne (normale) multivariée :

$$p(x \mid y = y_k) = \frac{1}{\sqrt{(2\pi)^d \mid C_k \mid}} e^{-\frac{1}{2}(x-\mu)^T C_k^{-1}(x-\mu)}.$$

Fonction objective pour classer l'observation x :

$$y^*(x) = \arg\max_y g_y(x) \tag{9}$$

Cas général : **QDA** :

$$g_{y_k}(x) = -\frac{1}{2}(x-\mu_k)^T C_k^{-1}(x-\mu_k) - \frac{1}{2} \mid C_k \mid + \log P(y_k). \tag{10}$$

Supposons $\forall k\ C_k = C$: **LDA** :

$$\arg\max_{y_k} [g_{y_k}(x)] = \arg\max_{y_k} \left[-\frac{1}{2}(x-\mu_k)^T C^{-1}(x-\mu_k) - \frac{1}{2} \mid C \mid + \log P(y_k) \right] = \tag{11}$$

$$= \arg\max_{y_k} \left[-\frac{1}{2} x^T C^{-1} x - 2x^T C^{-1}\mu_k + \mu_k^T C^{-1}\mu_k + \log P(y_k) \right] = \tag{12}$$

$$\arg\max_{y_k} \left[-2x^T C^{-1}\mu_k + \mu_k^T C^{-1}\mu_k + \log P(y_k) \right] \tag{13}$$

Supposons $\forall k\ C_k = C,\ C = diag(\sigma_1, \dots \sigma_d)$ (diagonale). On obtient une pondération des distances :

$$\arg\max_{y_k} [g_{y_k}(x)] = \tag{14}$$

$$\arg\max_{y_k} \left[-\frac{1}{2} \sum_{i=1}^{d} \frac{(x_i - \mu_{ik}) * (x_i - \mu_{ik})}{\sigma_i^2} - \frac{1}{2} \mid \sigma^2 I \mid + \log P(y_k) \right] = \tag{15}$$

$$\arg\max_{y_k} \left[-\frac{1}{2} \sum_{i=1}^{d} \frac{(x_i - \mu_{ik}) * (x_i - \mu_{ik})}{\sigma_i^2} + \log P(y_k) \right] \tag{16}$$

Supposons $\forall k\ C_k = \sigma^2 I$ (diagonale et une seule variance pour toute variable).

$$\arg\max_{y_k} [g_{y_k}(x)] = \tag{17}$$

$$\arg\max_{y_k} \left[-\frac{1}{2\sigma^2} (x - \mu_k)^T (x - \mu_k) - \frac{1}{2} \mid \sigma^2 I \mid + \log P(y_k) \right] = \tag{18}$$

$$\arg\max_{y_k} \left[-(x - \mu_k)^T (x - \mu_k) + \log P(y_k) \right] \tag{19}$$

Question : Comment est-ce que l'on obtient des différents classifieurs de distance minimum à partir de là ?