IFT3395/6390, Devoir 1

Date affiché: Oct. 1, 2014,
À remettre: Oct. 8, 2014, au *debut* du class

---

1. (1/10) Soit les exemples d'entraînement $x_1, \ldots, x_n$. À partir de ces exemples, montrez que l'estimation du maximum de vraisemblance pour les paramètres d'une distribution Bernoulli est: $\frac{1}{N}\sum_n x_n$, en fixant la dérivée de la log-vraisemblance à zéro.

2. (1/10) En classe, nous avons discuté du modèle génératif gaussien pour la classification, ainsi que du modèles de Bayes naïf discret. Décrivez la façon dont on peut définir un *classifieur gaussien naïf de Bayes*. Quels sont ses paramètres? Quels sont les estimations du maximum de vraisemblance?

3. (3/10) Dans cette question, nous «abuserons» du modèle de régression linéaire multi-dimensionnelle pour résoudre une tâche de classification.

   Téléchargez les fichiers suivants:

   `www.iro.umontreal.ca/~memisevr/teaching/ift3395_2014/devoirs/train_images.txt`

   `www.iro.umontreal.ca/~memisevr/teaching/ift3395_2014/devoirs/test_images.txt`

   `www.iro.umontreal.ca/~memisevr/teaching/ift3395_2014/devoirs/train_labels.txt`

   `www.iro.umontreal.ca/~memisevr/teaching/ift3395_2014/devoirs/test_labels.txt`

   et enregistrez les localement. Jetez un coup d'oeil au contenu des fichiers. Ils contiennent 1000 images binaires de chiffres manuscrits ainsi que leur classe, toutes encodées sous forme de matrices de nombres réels, séparés par des virgules. Plus spécifiquement, chacun des deux premiers fichiers représent 1000 images binaires de taille 28×28 pixels (une image par ligne). Chacun des deux derniers fichiers représentent 1000 cibles sous forme d'encodage «one-hot». Les premier et troisième fichiers représentent les données d'entraînement, tandis que les deuxième et quatrième représentent les données de test.

   Utilisez les *données d'entraînement* pour entraîner un modèle de régression linéaire régularisée (sans terme biais), qui prédit un vecteur de sortie de dimension 10, à partir d'un vecteur d'entrées $\mathbf{x}$ de dimension 784.

   Rappelez vous que le modèle est défini par une matrice $\mathbf{W}$ de taille 784×10 qui minimize le coût $E(\mathbf{W}) = \sum_n \|\mathbf{t}_n - \mathbf{W}^{\mathrm{T}}\mathbf{x}_n\|^2 + \lambda \sum_k \|\mathbf{w}_k\|^2$ où $\mathbf{w}_k$ est une colonne de la matrice $\mathbf{W}$ et λ est une constante que vous devez choisir. Pour l'entraînement, vous pouvez utiliser soit la descente de gradient ou la solution de forme fermée dont nous

avons discuté en classe. Expérimentez avec différentes valeurs pour λ (ou bien, pour λ sous $(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda I)^{-1}$ dans le cas de la solution de forme fermée). Vous pouvez trouver que pour certaines valeurs, l'apprentissage devient instable ou ne fonctionne pas.

Après l'entraînement du modèle de régression, classifiez les exemples d'entraînement et de test de la manière suivante: Pour chaque exemple (d'entraînement ou de test) $\mathbf{x}$, calculez $\mathbf{W}^{\mathrm{T}}\mathbf{x}$. Définissez l'indice de la dimension qui donne la plus grande valeur comme étant la classe pour cet exemple. En d'autres termes, la sortie de votre classifieur pour l'entrée $\mathbf{x}$ est défini comme suit:

$$\operatorname*{argmax}_k \mathbf{w}_k^{\mathrm{T}}\mathbf{x}$$

Indice:

- Si vous utilisez python, vous pouvez utiliser la fonction numpy.linalg.inv() pour inverser une matrice.

À remettre: Quelle valeur du paramètre de régularisation λ, parmis celles que vous avez essayée, fonctionnent le mieux pour les données de test? Pour cette valeur, quel est le taux d'erreur de classification sur les données d'entraînement et quel est le taux d'erreur de classification sur les données de test?

4. (4/10) Estimez un *classifieur Bernoulli naïf de Bayes* en utilisant les données de la question précédente, et appliquez le sur les données d'entraînement et les données de test.

   À remettre:

   - Les 10 probabilités à priori.
   - Dix images de taille 28×28 pixels, où chacun montre la distribution $p(\mathbf{x}|\mathcal{C}_k)$ pour une classe $\mathcal{C}_k$ tel qu'une image noire et blanche en teinte de gris.
   - Deux chiffres: le taux d'erreur de classification sur les données d'entraînement et le taux d'erreur de classification sur les données de test?

   Indices:

   - Pour estimer le classifieur naïf de Bayes, vous aurez besoin d'estimer les probabilités à priori sur les classes, ainsi que les probabilités conditionelles Bernoullis sur les vecteur de taille 784 representant des images.
   - Il peut être utile de représenter toutes les probabilités en utilisant de la logarithme lorsque vous effectuez des calculs avec eux, afin d'éviter débordement.
   - Pour éviter de prendre le log de 0 vous pouvez vous y prendre de la manière suivante: Lors de l'estimation des probabilités de classe conditionnelle, supposez que votre ensemble de données d'entraînement contient *deux* images supplémentaires pour chaque classe: Une image blanche contenant seulement des valeurs 1, et une image noire contenant seulement des valeurs 0. (Dans ce cas, le vrai nombre total d'images dans l'ensemble d'entraînement est de 1020 plutôt que 1000.)

- Si vous utilisez python / matplotlib, vous pouvez utiliser pylab.imshow () pour visualiser les images. Réorganisez les vecteurs de dimension 784 dans des matrices de taille 28×28 avant de les passer à imshow(). Vous pouvez utiliser pylab.gray() pour représenter les images avec la palette de couleurs grise.

5. (1/10) Un réseau de neurones pour la classification transforme K scores y_k, un par classe, en probabilités à l'aide de la fonction softmax:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{\exp(y_k(\mathbf{x}))}{\sum_j \exp(y_j(\mathbf{x}))}$$

Quelle est la dérivée de la log-probabilité d'un exemple d'entraînement selon ce modèle $\log p(t_n|\boldsymbol{x}_n)$ par rapport à une des entrées, y_k?