Département d'informatique et de recherche opérationnelle Automne 2008

IFT 6266: Algorithmes d'apprentissage

Yoshua Bengio et James Bergstra, pift6266@iro.umontreal.ca

Devoir #2, Donné le 22 septembre 2008, Dû le 6 octobre 2008

Ces exercices visent à vous familiariser avec la notion de vraisemblance, le maximum de vraisemblance (si ces notions ne vous sont pas déjà familières), la notion de critère d'entraînement et d'optimisation numérique du critère (en particulier par descente de gradient) afin de trouver la solution à un problème d'apprentissage.

L'exercice pratique, comme ceux du premier devoir, visent à vous faire explorer les notions d'erreur d'apprentissage et de test quand on varie certains hyper-paramètres. Imaginez-vous comme des scientifiques expérimentaux qui cherchent à découvrir des régularités (des lois scientifiques) en faisant des expériences. Montrez-nous ce que vous aurez découvert.

1. (2 points) On lance un dé pipé n fois en l'air et on obtient les mesures suivantes: $d_1, d_2, \ldots, d_n$, avec $d_i \in \{1, 2, 3, 4, 5, 6\}$ indiquant la face sur laquelle le dé est tombé. On considère les d_i des réalisations i.i.d. de la variable aléatoire D. Montrez que l'estimateur au maximum de vraisemblance pour $P(D = k)$ est la fréquence relative de l'évènement $D = k$ dans les données.

2. (2 points) On nous donne une série de n mesures de taille (en centimètres) et d'âge (en années) pour des adolescents entre 12 et 18 ans. On veut construire un modèle gaussien de la distribution conditionnelle de la taille étant donné l'âge, de la forme suivante:

$$p(\text{taille}|\text{age}) = N(\text{taille}; \mu(\text{age}), \sigma^2)$$

où $N(y; m, \sigma^2) = \exp(-0.5(y - m)^2/\sigma^2)/\sqrt{2\pi\sigma}$ est la densité normale d'espérance m et de variance σ^2, et

$$\mu(x) = wx + b$$

est un prédicteur linéaire de la taille y étant donné l'âge x. Les paramètres inconnus sont donc les scalaires w, b et σ. Soit $\{(x_i, y_i)\}_{i=1}^n$ un ensemble donné de n paires (age,taille). Montrez comment choisir (w, b, σ) par le principe du maximum de vraisemblance, c'est à dire pour maximiser

$$L = \log p(y_1, y_2, ...y_n|x_1, x_2, ...x_n)$$

où comme on nous dit que les n mesures sont i.i.d., on peut tirer avantage du fait que

$$p(y_1, y_2, ...y_n|x_1, x_2, ...x_n) = p(y_1|x_1)p(y_2|x_2)...p(y_n|x_n).$$

Grâce à ce modèle, on pourra répondre à la question suivante, étant donné l'âge x d'un nouvel adolescent: quelle est la distribution prévue des tailles y, pour cet adolescent? comme on a choisi une loi conditionnelle gaussienne, on pourra donner la taille attendue (en moyenne), et l'incertitude autour de cette espérance (l'écart-type), pour cette nouvelle valeur d'âge x, et on pourrait répondre à des questions du genre "quel est la probabilité estimée que cet adolescent mesure entre 180 et 185 cm?". Expliquez comment on pourrait numériquement répondre à cette dernière question.

3. (6 points) Vous allez implanter un algorithme de descente de gradient stochastique pour la régression logistique, qui estime $P(Y = 1|x)$ (pour Y binaire) par $f(x) = sigmoid(w'x+b)$, avec $sigmoid(a) = 1/(1+e^{-a})$. Soit $\theta = (b, w)$ le vecteur des paramètres. Vous allez utiliser une régularisation L1 sur w (on pénalise les valeurs absolues des poids). Vous allez considérer une perte par exemple (incluant la régularisation) qui est

$$L(x, y, \theta) = -y \log f(x) - (1 - y) \log(1 - f(x)) + \lambda \sum_i |w_i|.$$

La descente de gradient stochastique itère sur les exemples d'apprentissage (x, y) et pour chacun effectue la mise à jour des paramètres

$$\theta \leftarrow \theta - \epsilon \frac{\partial L(x, y, \theta)}{\partial \theta}.$$

La première étape est donc d'écrire la formule pour $\frac{\partial L(x,y,\theta)}{\partial \theta}$. La deuxième est d'implanter un algorithme qui fait l'apprentissage itératif. Vous pouvez initialiser tous les paramètres à 0. Après une mise à jour des paramètres pour chaque exemple d'apprentissage, vous pouvez mesurer l'erreur de classification et la moins-log-vraisemblance (la moyenne des $-\log P(y|x)$) du prédicteur sur les données d'apprentissage et les données de test. L'erreur d'apprentissage devrait généralement diminuer d'une fois à l'autre. Si ϵ est trop grand l'erreur va osciller ou même augmenter (possiblement avec une valeur infinie de $-\log P(y|x)$ quand le predicteur donne une probabilité numériquement 0 à la valeur observée de y). Il va vous falloir expérimenter avec ϵ et λ pour trouver les valeurs qui fonctionnent le mieux pour un jeu de données particulier. Pour ces deux hyper-paramètres, on peut s'attendre à trouver une courbe en U (si vous ne la trouvez pas, il y a sans doute une erreur dans votre implantation ou bien vous n'avez pas étendu suffisament votre recherche de valeurs). Pour ϵ des valeurs raisonnables à explorer pourraient être 0.1, 0.01, 0.001, 1e-4, 1e-5. Il y a un troisième hyper-paramètre qui est le nombre d'itérations de l'optimisation, que vous pouvez garder fixe ici (une valeur raisonnable comme point de départ: 100,000 mises à jour des paramètres). Remarquez cependant comment l'erreur d'apprentissage et l'erreur de test varient en fonction du nombre d'itérations (en recalculant ces deux erreurs après chaque passe à travers l'ensemble d'apprentissage). Faites vos expérimentations sur les données d'images de rectangles et triangles du devoir 1. Utilisez la même division des exemples en ensemble d'apprentissage D_{train} et ensemble de test D_{test} (de même taille).