Statistics 161/261. Spring 2010. Prof. A.L. Yuille. Homework 3.

Due: Wednesday 9/June. 2010.

Chp 11. Alpaydin.

Question 3.5.

Chp 13. Alpaydin.

Question 1,2.

Additional Questions

Question 1. Primal Dual Quadratic Optimization.

The primal problem is formulated as follows:

$$L_p(\vec{a}, b, \{z_i\}; \{\alpha_i, \mu_i\}) = (1/2)|\vec{a}|^2 + \gamma \sum_{i=1}^{m} z_i - \sum_{i=1}^{m} \alpha_i \{\omega_i(\vec{a}\cdot\vec{x}_i + b) - (1 - z_i)\} - \sum_{i=1}^{m} \mu_i z_i. \quad (1)$$

Explain the meaning of all the terms and variables in this equation. What constraints do the variables satisfy? Calculate the form of the solution $\vec{a}$ by minimizing L_p with respect to $\vec{a}, b, \{z_i\}$. What are the support vectors?

What is the dual formulation? Describe a strategy for solving the primal problem.

How is this approach extended to multi-class classification and to latent support vector machines?

Question 2. Support Vector Machine.

Consider a binary classification problem where the data is one-dimensional. There are two negative ($\omega = -1$) examples at $x = \pm 1$ and one positive ($\omega = 1$) example at $x = 0$. Show that you cannot classify this data perfectly by linear separation (i.e. by a decision rule $sign(ax + b)$ for some a, b).

Now formulate this problem with slack variables $\{z_i\}$. The classifier with the largest margin is obtained by solving the primal problem: $L_P = (1/2)a^2 + \gamma \sum_{i=1}^{3} z_i - \sum_{i=1}^{3} \alpha_i \{\omega_i(ax_i + b) - (1 - z_i)\} - \sum_{i=1}^{3} \mu_i z_i$.
where γ is a constant, the $\{\alpha_i, \mu_i\}$ are Lagrange multipliers (constrained to be non-negative) and the $\{z_i\}$ are non-negative.

Minimize L_p by searching for the minimum of $(1/2)a^2 + \gamma(z_1 + z_2 + z_3)$ subject to the constraints $\omega_i(ax_i + b) - (1 - z_i) \geq 0,\ \forall\ i \in \{1, 2, 3\}$ with $z_i \geq 0,\ \forall\ i \in \{1, 2, 3\}$. (Hint: exploit structure of the problem to guess where the decision boundary should be). What are the support vectors?

Question 3. Features and Kernels.

Show that you can solve the classification problem of Question 2 using kernel methods. (Hint: consider a special choice of kernel).

Consider a binary classification problem. The data lies in the two-dimensional plane $(x_1, x_2) \in R^2$. The only positive example ($\omega = +1$) is at the origin $(0, 0)$. The are 8 negative examples ($\omega = -1$) equally spaced on a circle centered on the origin with radius R.

Choose a three-dimensional feature vector which enables the positive and negative examples to be separated by a linear hyperplane (in feature space).

Use these feature vectors to calculate a kernel $K(x_1, x_2; x_1', x_2')$ for this problem.

Show that the classifier in feature space can be expresses using kernels as $sign(g(x_1, x_2))$ where $g(x_1, x_2) = \sum_{\mu=1}^{8} \alpha_\mu \omega_\mu K(x_1, x_2 : x_1^\mu, x_2^\mu) + \alpha_0 \omega_0 K(x_1, x_2 : 0, 0) + b$, where $\{(x_1^\mu, x_2^\mu) : \mu = 1, ..., 8\}$ are the data points on the circle.

What is the minimal number of α's that need to be non-zero to enable classification? (The answer will depend on your choice of feature vectors).

Question 4. AdaBoost.

The AdaBoost learning algorithm takes an input dataset $\{(x_i, y_i) : i = 1, ..., m\}$. It updates the weights of the data by $D_{t+1}(i) = D_t(i) e^{-y_i \alpha_t h_t(x_i)} / Z_t$. It selects the weak classifier $h_t(.)$ which minimizes $Z_t = \sum_{i=1}^{m} D_t(i) e^{-y_i \alpha_t h_t(x_i)}$. What strong classifier does AdaBoost learn? How does $Z_1 Z_2 ... Z_N$ relate to the error rate of the strong classifier?

Apply AdaBoost to the one-dimensional problem where the data lies on the x-axis. There is one positive example at $x = 0$ and two negative examples at $x = \pm 1$. There are three weak classifiers are $h_1(x) = 1, \ x > 1/2, \& \ h_1(x) = -1, \ x < 1/2, \ h_2(x) = 1, \ x > -1/2, \& \ h_2(x) = -1, \ x < 1/2$, and $h_3(x) = 1, \ \forall x$. Show that this data can be classified correctly by a strong classifier which uses only three weak classifiers. Calculate the first two iterations of AdaBoost for this problem. Are they sufficient to classify the data correctly?