CSE 575: Statistical Machine Learning Assignment #2

Instructor: Prof. Hanghang Tong
Out: Feb. 19th, 2016; Due: Mar. 18th, 2016

Submit electronically, using the submission link on Blackboard for Assignment #1, a file named `yourFirstName-yourLastName.pdf` *containing your solution to this assignment (a .doc or .docx file is also acceptable, but .pdf is preferred).*

1 Logistic Regression [20 points]

Suppose we have two positive examples $x_1 = (1, 1)$ and $x_2 = (1, -1)$; and two negative examples $x_3 = (-1, 1)$ and $x_4 = (-1, -1)$. We use the standard gradient ascent method (without any additional regularization terms) to train a logistic regression classifier. What is the final weight vector w? Justifiy your answer. You can assume that the weight vector starts at the origin, i.e., $w_0 = (0, 0, 0)'$. How would you explain the final weight vector w you get?

Solutions: The final weight vector $w = (0, \infty, 0)'$. We can prove this by induction.

Suppose at t^{th} iteration, the current weight vector $w_t = (0, c, 0)$, where $c \geq 0$, we will show that after the update, $w_{t+1} = (0, c + d, 0)$, where $d > 0$.

We can verify that $P(y_1 = 1|x_1) = P(y_2 = 1|x_2) = \frac{e^c}{1+e^c} = a$, and $P(y_3 = 1|x_3) = P(y_4 = 1|x_4) = \frac{1}{1+e^c} = b$; and $a + b = 1$.

Thus, by gradient ascent, we have $w_{t+1} = w_t + \eta \sum_{i=1}^{4} x_i(y_i - P(y_i = 1|x_i, w_t)) = w_t + \eta(2(1 - a - b), 2(1 + b - a), 0)' = (0, c + 2\eta(1 + b - a), 0$, where $\eta > 0$ is the learning rate; and $0 < a, b < 1$. Therefore $d = 2\eta(1 + b - a) > 0$.

The intuition is that this final weight vector maximizes the likelihood of the training set.

2 2-Class Logistic Regression [20 points]

Prove that 2-class logistic regression is always a linear classifier.

Solution: see page-47 in lecture slides

3 SVM [20 points]

1 (**Kernel, 10 pts**) Given the following dataset in 1-d space (Figure **??**), which consists of 3 positive data points $\{-1, 0, 1\}$ and 3 negative data points $\{-3, -2, 2\}$.

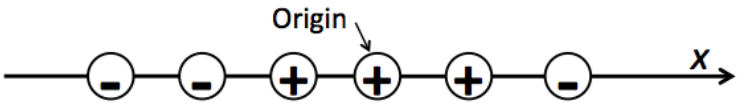

Figure 1: Training Data Set for SVM Classifiers

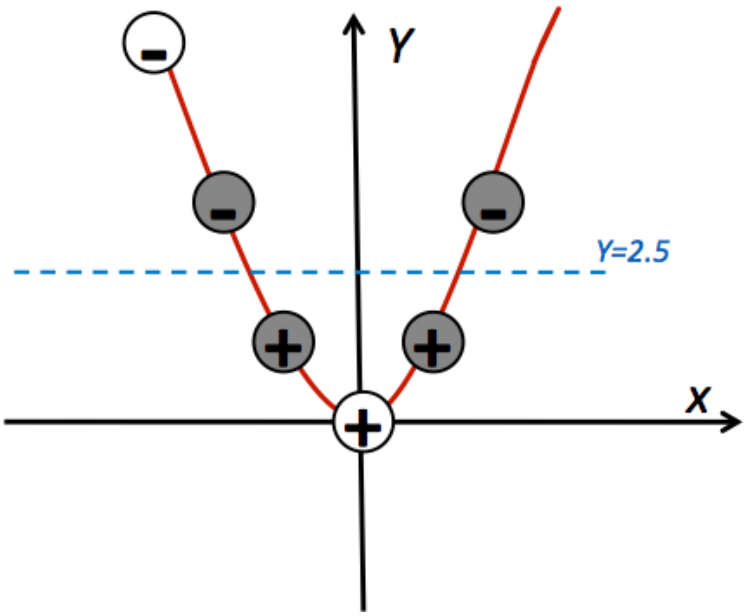

Figure 2: Feature Map

(1) Find a feature map($\{\mathbf{R}^1 \rightarrow \mathbf{R}^2\}$), which will map the original 1-d data points to 2-d space so that the positive set and negative set are linearly separable with each other. Plot the dataset after mapping in 2-d space.

Solutions: $x \rightarrow \{x, x^2\}$. See figure 2.

(2) In your plot, draw the decision boundary given by hard-margin linear SVM. Mark the corresponding support vector(s).

Solutions: See figure 2 (supported vectors are shadowed ones).

(3) For the feature map you choose, what is the corresponding kernel $K(x_1, x_2)$?

Solutions: $K(x_1, x_2) = x_1 x_2 + (x_1 x_2)^2$.

2 (**Hinge Loss, 5 pts**) Given m training data points $\{x_i, y_i\}_{i=1}^m$, remember that the soft-margin linear SVM can be formalized as the following constrained quadratic optimization problem:

$$\begin{aligned} &\text{argmin}_{\{w,b\}} \ \frac{1}{2} w^t w + C \sum_1^m \epsilon_i \\ &\text{Subject to } : y_i(w^t x_i + b) \geq 1 - \epsilon_i \\ &\epsilon_i \geq 0 \ \forall i \end{aligned} \tag{1}$$

(1) Prove that the above formulation is equivalent to the following unconstrained quadratic optimization problem:

$$\text{argmin}_{\{w,b\}} w^t w + \lambda \sum_1^m \max(1 - y_i(w^t x_i + b), 0) \tag{2}$$

Solutions: By eq. (1), we have $\epsilon_i \geq \max(1 - y_i(w^t x_i + b), 0),\ \forall i$. Since we want to minimize ϵ_i, we must have $\epsilon_i = \max(1 - y_i(w^t x_i + b), 0),\ \forall i$. Plus this into eq. 1, we have eq. (2).

(2) What is your intuition for this new optimization formulation (1 or 2 sentences)?

Solutions: Soft-margin SVM tries to balance the simplicity of classifier (the first term) vs. the good prediction on training dataset (the second term).

(3) What is the value for λ (as a function of C)?

Solutions: $\lambda = 2C$.

3 (**SMO, 5 pts**) Suppose we are given 4 data points in 2-d space: $x_1 = (0, 1),\ y_1 = -1$; $x_2 = (2, 0),\ y_2 = +1$; $x_3 = (1, 0),\ y_3 = +1$; and $x_4 = (0, 2),\ y_4 = -1$. We will use these 4 data points to train a soft-margin linear SVM. Let $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ be the Lagrange multipliers for x_1, x_2, x_3, x_4 respectively. And also let the regularization parameter C be 100.

(1) Write down the dual optimization formulation for this problem

Solutions:

$$\begin{aligned} &\text{argmax}_{\alpha_1,\alpha_2,\alpha_3,\alpha_4} \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - 1/2(\alpha_1^2 + 4\alpha_2^2 + \alpha_3^2 + 4\alpha_4^2 + 4\alpha_2\alpha_3 + 4\alpha_1\alpha_4) \\ &\text{Subject to :} 0 \leq \alpha_1, \alpha_2, \alpha_3, \alpha_4 \leq 100 \\ &-\alpha_1 + \alpha_2 + \alpha_3 - \alpha_4 = 0 \quad (3) \end{aligned}$$

(2) Suppose we initialize $\alpha_1 = 5, \alpha_2 = 4, \alpha_3 = 8, \alpha_4 = 7$. And we want to update α_1 and α_4 (keep α_2 and α_3 fixed) in the 1st iteration. Derive the update equations for α_1 and α_4 (in terms of α_2 and α_3). What are the values for α_1 and α_4 after update?

Solutions: $\alpha_1 = \alpha_2 + \alpha_3 = 12$ and $\alpha_4 = 0$.

(3) Now fix α_1 and α_4, derive the update equations for α_2 and α_3 (in terms of α_1 and α_4). What are the values for α_2 and α_3 after update?

Solutions: $\alpha_3 = \alpha_1 + \alpha_4 = 12$ and $\alpha_2 = 0$.

4 More on SVM [20 points]

1. [4 points] Suppose we are using a linear SVM (i.e., no kernel), with some large C value, and are given the following data set.

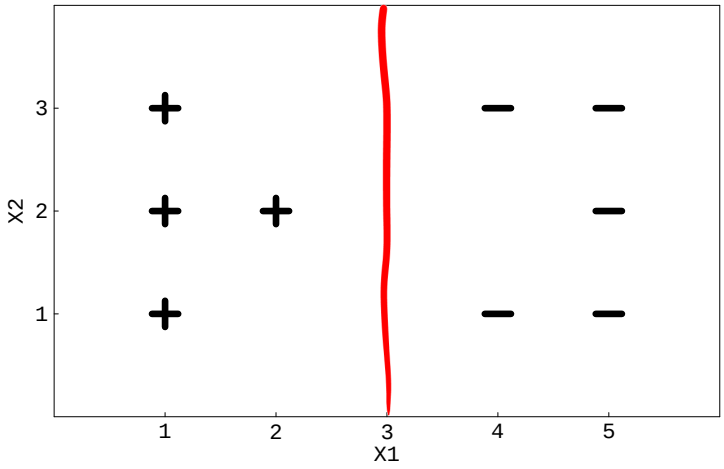

Figure 3

Draw the decision boundary of linear SVM. Give a brief explanation.

Solution. Because of the large C value, the decision boundary will classify all of the examples correctly. Furthermore, among separators that classify the examples correctly, it will have the largest margin (distance to closest point).

2. [4 points] In the following image, circle the points such that after removing that point (example) from the training set and retraining SVM, we would get a different decision boundary than training on the full sample.

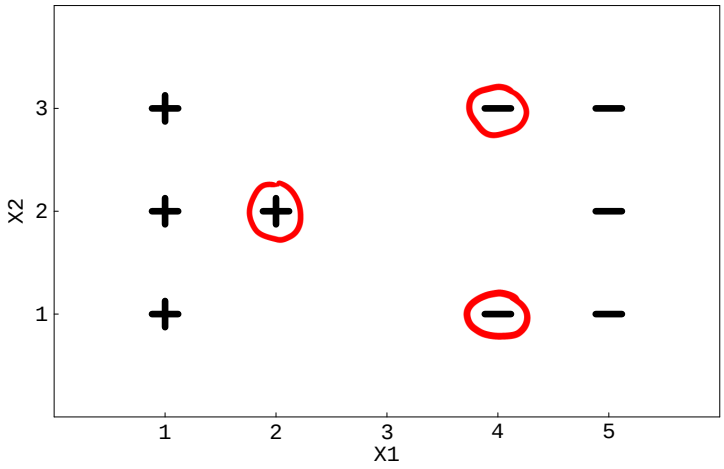

Figure 4

Justify your answer.

Solution. These examples are the support vectors; all of the other examples are such that their corresponding constraints are not tight in the optimization problem, so removing them will not create a solution with smaller objective function value (norm of w). These three

examples are positioned such that removing any one of them introduces slack in the constraints, allowing for a solution with a smaller objective function value and with a different third support vector; in this case, because each of these new (replacement) support vectors is not close to the old separator, the decision boundary shifts to make its distance to that example equal to the others.

3. [4 points] Suppose instead of SVM, we use regularized logistic regression to learn the classifier. That is,

$$(w, b) = \arg \min_{w \in \mathbb{R}^2, b \in \mathbb{R}} \frac{\|w\|^2}{2} - \sum_i \mathbb{1}[y_i = 0] \ln \frac{1}{1 + e^{(w \cdot x_i + b)}} + \mathbb{1}[y_i = 1] \ln \frac{e^{(w \cdot x_i + b)}}{1 + e^{(w \cdot x_i + b)}}.$$

In the following image, circle the points such that after removing that point (example) from the training set and running regularized logistic regression, we would get a different decision boundary than training with regularized logistic regression on the full sample.

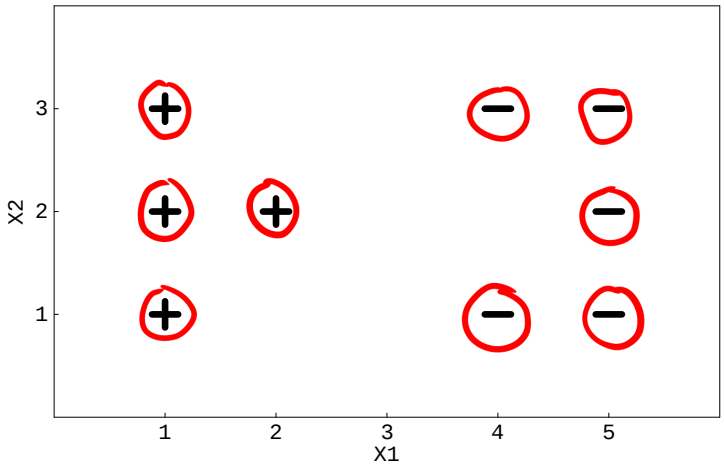

Figure 5

Justify your answer.

Solution. Because of the regularization, the weights will not diverge to infinity, and thus the probabilities at the solution are not at 0 and 1. Because of this, *every* example contributes to the loss function, and thus has an influence on the solution.

4. [4 points] Suppose we have a kernel $K(\cdot, \cdot)$, such that there is an implicit high-dimensional feature map $\phi : \mathbb{R}^d \to \mathbb{R}^D$ that satisfies $\forall x_i, x_j \in \mathbb{R}^d$, $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$, where $\phi(x_i) \cdot \phi(x_j) = \sum_{l=1}^{D} \phi(x_i)^l \phi(x_j)^l$ is the dot product in the D-dimensional space, and $\phi(x_i)^l$ is the l^{th} element/feature in the D-dimensional space.

Show how to calculate the Euclidean distance in the D-dimensional space

$$\|\phi(x_i) - \phi(x_j)\| = \sqrt{\sum_{l=1}^{D} (\phi(x_i)^l - \phi(x_j)^l)^2}$$

without explicitly calculating the values in the D-dimensional space. For this question, **please provide a formal proof**.
Hint: Try converting the Euclidean distance into a set of inner products.

Solution.

$$\begin{aligned}
\|\phi(x_i) - \phi(x_j)\| &= \sqrt{\sum_{l=1}^{D} (\phi(x_i)^l - \phi(x_j)^l)^2} \\
&= \sqrt{\sum_{l=1}^{D} (\phi(x_i)^l)^2 + (\phi(x_j)^l)^2 - 2\phi(x_i)^l \phi(x_j)^l} \\
&= \sqrt{\left(\sum_{l=1}^{D} (\phi(x_i)^l)^2\right) + \left(\sum_{l=1}^{D} (\phi(x_j)^l)^2\right) - \left(\sum_{l=1}^{D} 2\phi(x_i)^l \phi(x_j)^l\right)} \\
&= \sqrt{\phi(x_i) \cdot \phi(x_i) + \phi(x_j) \cdot \phi(x_j) - 2\phi(x_i) \cdot \phi(x_j)} \\
&= \sqrt{K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)}.
\end{aligned}$$

5. [2 points] Assume that we use the RBF kernel function $K(x_i, x_j) = \exp(-\frac{1}{2}\|x_i - x_j\|^2)$. Also assume the same notation as in the last question. Prove that for any two input examples x_i and x_j, the squared Euclidean distance of their corresponding points in the high-dimensional space $\mathbb{R}^D$ is less than 2, i.e., prove that $\|\phi(x_i) - \phi(x_j)\|^2 < 2$.

   Solution. This inequality directly follows from the result from the last question.

6. [2 points] Assume that we use the RBF kernel function, and the same notation as before. Consider running One Nearest Neighbor with Euclidean distance in both the input space $\mathbb{R}^d$ and the high-dimensional space $\mathbb{R}^D$. Is it possible that One Nearest Neighbor classifier achieves better classification performance in the high-dimensional space than in the original input space? Why?

   Solution. No.

5 Naïve Bayes Classifier and Logistic Regression [20 points]

1. **Gaussian Naïve Bayes and Logistic Regression (10 points).** Suppose we want to train Gaussian Naïve Bayes to learn a boolean/binary classifier: $f : X \to Y$, where X is a vector of n dimensional real-valued features: $X =< X_1, ..., X_n >$; and Y is boolean class label (i.e., $Y = 1$ or $Y = 0$). Recall that in Gaussian Naïve Bayes, we assume all $X_i (i = 1, ..., n)$ are conditionally independent given the class label Y, i.e., $P(X_i|Y = k) \sim \mathcal{N}(\mu_{ik}, \sigma_i)$ $(k = 0, 1; i = 1, ..., n)$. We also assume that $P(Y)$ follows Bernoulli$(\theta, 1 - \theta)$ (i.e., $P(Y = 1) = \theta$).

   – How many independent model parameters are there in this Gaussian Naïve Bayes classifier?

Solutions: $3n + 1$ (2 for each Xi of a given label, but for the same Xi, it shares the same variance between two class labels, and the prior)

– Prove that the Gaussian Naïve Bayes assumption imply that $P(Y|X)$ follow the form of $P(Y = 1|X =< X_1, ..., X_n >) = \frac{1}{1+exp(w_0+\sum_{i=1}^{n} w_i X_i)}$. In particular, you need to express w_i $(i = 0, ..., n)$ by the model parameters (i.e., $\theta, \mu_{ik}, \sigma_i$ $(k = 0, 1; i = 1, ..., n)$).

Solutions:

$$\begin{aligned} P(Y=1|X) &= \frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1)+P(Y=0)P(X|Y=0)} \text{ (Bayes Rule)} \\ &= \frac{1}{1+\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}} \\ &= \frac{1}{1+exp(ln(\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}))} \\ &= \frac{1}{1+exp(ln(\frac{1-\theta}{\theta})+\sum_i ln(\frac{P(X_i|Y=0)}{P(X_i|Y=1)}))} \text{ (Naïve Bayes Assumption)} \end{aligned}$$

Since $P(X_i|Y = k) \sim \mathcal{N}(\mu_{ik}, \sigma_i)$ $(k = 0, 1; i = 1, ..., n)$, we have $P(X_i = x|Y = k) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{\frac{-(x-\mu_{ik})^2}{2\sigma_i^2}}$, which implies

$$ln(\frac{P(X_i|Y=0)}{P(X_i|Y=1)}) = \sum_i (\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}X_i + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2})$$

which completes the proof, with $w_0 = ln(\frac{1-\theta}{\theta}) + \sum_i \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}$ and $w_i = \frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}$.

2. **Boolean Naïve Bayes and Logistic Regression (10 points).** Now consider X being a vector of boolean variables (Y still being the boolean class label). We still want to train a Naïve classifier.

– How many independent model parameters are there in this boolean Naïve Bayes classifier?

Solutions: $2n + 1$ (1 for each Xi of a given label, and the prior)

– Let $P(X_i|Y = k) = \theta_{ik}$ $(k = 0, 1)$ and $P(Y = 1) = \theta$. Prove that the boolean Naïve Bayes assumption imply that $P(Y|X)$ follow the form of $P(Y = 1|X =< X_1, ..., X_n >) = \frac{1}{1+exp(w_0+\sum_{i=1}^{n} w_i X_i)}$. In particular, you need to express w_i $(i = 0, ..., n)$ by the model parameters (i.e., $\theta, \mu_{ik}, \sigma_i$ $(k = 0, 1; i = 1, ..., n)$).

Solutions:

Follow the same procedure as above, except that now $P(X_i|Y = k) = \theta_{ik}^{X_i}(1-\theta_{ik})^{1-X_i}$ $(k = 0, 1)$. This leads to

$$ln(\frac{P(X_i|Y=0)}{P(X_i|Y=1)}) = \sum_i (ln\frac{\theta_{i0}}{\theta_{i1}} - ln\frac{1-\theta_{i0}}{1-\theta_{i1}})X_i + ln\frac{1-\theta_{i0}}{1-\theta_{i1}})$$

This completes the proof, with $w_0 = ln(\frac{1-\theta}{\theta}) + \sum_i ln\frac{1-\theta_{i0}}{1-\theta_{i1}}$ and $w + i = (ln\frac{\theta_{i0}}{\theta_{i1}} - ln\frac{1-\theta_{i0}}{1-\theta_{i1}}) = ln\frac{\theta_{i0}(1-\theta_{i1})}{\theta_{i1}(1-\theta_{i0})}$.