Stats 201B: Statistical Modeling and Learning
Problem Set 4

Feb 19, 2016

Due 26 Feb

1. **Generative Models for Classification. (18 points)**
   Consider a sample of observations (Y_i, X_i), where Y_i is a class label $1, 2, ..., G$ and X_i is a vector of predictors.

   (a) *(5 pts.)* Suppose you wish to estimate a classifier for this problem by using a "Bayes classifier", that is, one that estimates the (posterior) probability $p(Y_i = g|X_i)$ for each g, and then chooses the g that maximizes this probability.

   Describe how *generative models* approach this problem. (That is, how might you theoretically estimate $p(Y_i = g|X_i)$ for each class g?)

   Suppose Y_i has a prior distribution with density function $p(y)$.

   $$P(Y_i = g|X_i) = \frac{f(x_i|g)p(g)}{\sum_{i=1}^{G} f(x_i|g)p(g)}$$

   To calculate maximum $P(Y_i = g|X_i)$, we don't need to calculate the denominator. So g is chosen by:

   $$\underset{g}{arg} \max f(x_i|g)p(g)$$

   (b) *(5 pts.)* Propose the most non-parametric, assumption-free way you can think of to actually estimate the posterior probability $p(Y_i = g|X_i)$ (for each g) by a generative approach. Why might this not be a good idea in practice?

   Assume the prior distribution is evenly distributed, namely $P(Y = g) = p(g) = \frac{1}{G}$. Given data set (X_i, Y_i), calculate the kernel density,

   $$f(x|Y = g) = \frac{1}{\#\{Y_i|Y_i = g\}h} \sum_{Y_i=g} K(\frac{x - x_i}{h}).$$

   Then choose g by $arg_g \max f(x_i|g)p(g)$. when Y_i is skew-distributed, we cannot get some proper distributions for $f(x|Y = g)$. When the dimension of X is increasing, the amount of data which guarantees the accuracy is increasing exponentially.

(c) *(5 pts.)* Explain how LDA, QDA, and Naive Bayes classifiers each tackle this problem. What is each doing, and how does each differ in their assumptions?

- LDA – Given that $\forall g, f(x|Y=g)$ have the same covariance matrix, we get g such that $arg_g \max f(x_i|g)$.
- QDA – Given that $\forall (X_i, Y_i)$, $f(x|Y=g)$ follows the same distribution with g. we get g such that $arg_g \max f(x_i|g)$.
- Naive Bayes – Given prior distribution $p(y)$, we get g such that $arg_g \max f(x_i|g)p(g)$.

Assume that $f(x|Y=g)$ is multi normal distribution.

- LDA – It is equal to compare some linear combination of X.
- QDA – It is equal to compare some quadratic combination of X.
- Naive Bayes classifiers – It depends on how we assume prior distribution and how to estimate μ, Σ for $f(x|Y=g) \sim N(\mu, \Sigma)$.

(d) *(3 pts.)* Propose some guidelines for when someone might prefer LDA, QDA, or Naive Bayes classifiers relative to each other, and then why you might sometimes prefer a non-generative approach such as logit or SVM.

- LDA – There is a limited amount of data so that QDA won't have good effect. I can infer that for all $f(x|Y=g)$, they have common covariance. Less calculation speed is required.
- QDA – For some $f(x|Y=g)$, they don't have similar covariance. When LDA performs poor, it can be tried.
- Naive Bayes classifiers – Certain prior knowledge is known. The dimension of X is not high or assumption that $f(x|Y=g) = \prod_i f(x_{(i)}|Y=g)$ is true.
- logit – Deal with some situation where the odds ratio of X don't change with X
- SVM – The above methods may need to establish some models and calculate some ancillary parameters, but we only need to know the classification of Y_i.

2. **Relationship between similarity (in X) and distance in $\phi(x)$. (8pts)**
This question will give you some practice working with inner-products, feature-spaces, and kernels. It also aims to show an initimate link between how a kernel measures similarity (in the input space, x) and how it measures distance (in the feature space, $\phi(x)$). Throughout the question, assume you have some positive semi-definite kernel $k(\cdot,\cdot)$ such that $k(X_i, X_j) = \langle \phi(X_i), \phi(X_j) \rangle$

(a) (2pts) Suppose you have vectors X_i and X_j. Write down the squared Euclidean distance between them as a norm using the notation $||\cdot||^2$.

$$SquaredDistance = ||X_i - X_j||^2$$

(b) (2pts) Now write a version of the expression above that instead takes the squared Euclidean distance in the feature space. That is, map each point to the feature space and write down the Euclidean distance there.

$$SquaredDistance = ||\phi(X_i) - \phi(X_j)||^2$$

(c) (4pts) Rewrite this using inner product notation, $\langle \cdot, \cdot \rangle$. Suppose now that you are using the $\phi(X)$ corresponding to a Gaussian kernel. Carry through the inner-product computation and simplify the expression as much as possible using properties of the Gaussian kernel. Discuss how the distance you have computed in the feature space relates back to how the kernel measured similarity in the input space. (You may find it useful to think of $1 - k(X_i, X_j)$ as a "dissimilarity" or difference measure.)

$$\begin{aligned} SquaredDistance &= \langle \phi(X_i) - \phi(X_j), \phi(X_i) - \phi(X_j) \rangle \\ &= \langle \phi(X_i), \phi(X_i) \rangle + \langle \phi(X_j), \phi(X_j) \rangle - \langle \phi(X_i), \phi(X_j) \rangle - \langle \phi(X_j), \phi(X_i) \rangle \\ &= k(0,0) + k(0,0) - k(X_i, X_j) - k(X_j, X_i) \\ &= 2 - 2e^{\frac{-||X_i - X_j||^2}{\sigma^2}} \end{aligned}$$

When X_i and X_j are similar in Euclidean distance, $||X_i - X_j||$ is close to 0 so that $2 - 2e^{\frac{-||X_i - X_j||^2}{\sigma^2}}$ is close to 0. This means X_i and X_j are similar in distance of the feature space. We should notice that exponential makes $\phi(X_i)$ and $\phi(X)_j$ much more dissimilar with the Euclidean distance $||X_i - X_j||^2$ increasing a little. So conversely, the more dissimilar the $\phi(X_i)$ and $\phi(X)_j$ are, the larger distance between X_i and X_j is.

3. **Kernel Ridge Regression: Setup. (12 points)**

One benefit of kernelized ridge regressions (such as Kernel Regularized Least Squares, KRLS) is that even though it fits highly non-linear functions, it has a closed-form analytical solution. Recall from lecture that KRLS can be formulated as a *linear* problem with the functional form $\mathbf{Y} = \mathbf{Kc} + \epsilon$:

$$\begin{bmatrix} f(X_1) \\ f(X_2) \\ \vdots \\ f(X_N) \end{bmatrix} = \begin{bmatrix} k(X_1, X_1) & k(X_1, X_2) & \cdots & k(X_1, X_N) \\ k(X_2, X_1) & k(X_2, X_2) & \cdots & k(X_2, X_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(X_N, X_1) & K(X_N, X_2) & \cdots & k(X_N, X_N) \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$$

where

- $f(X)$ is the value at X of the function we are trying to approximate.
- $\mathbf{K}$ is the *kernel matrix* whose entries $k(X_i, X_j)$ quantify, in some way, the similarities between the values of the independent variables for two different data observations i and j, and

- $\mathbf{c} = [c_1, \ldots, c_N]^T$ is the column-vector of choice coefficients which we want to solve for.

We want a model that satisfies two main criteria: we want to minimize the prediction error, and we want to also favor simpler functions over more complicated ones. To do this, we use (Tikhonov) Regularization. Skipping over some details, this implies choosing coefficients $\mathbf{c}$ to solve the following:

$$\underset{\mathbf{c}\in\mathbb{R}^N}{\operatorname{argmin}} (\mathbf{Y} - \mathbf{Kc})^T(\mathbf{Y} - \mathbf{Kc}) + \lambda \mathbf{c}^\top \mathbf{Kc} \tag{1}$$

where $\mathbf{c}^\top\mathbf{Kc}$ is an estimated norm of our function (a measure of its "complexity") and λ is our choice about how much we penalize complicated functions versus how much we penalize prediction error.

(a) *(10 pts.)* Find the closed-form solution for $\hat{\mathbf{c}}$, the minimizer of $(\mathbf{Y} - \mathbf{Kc})^T(\mathbf{Y} - \mathbf{Kc}) + \lambda(\mathbf{c}^\top\mathbf{Kc})$, showing your work.

$$Loss = Y^tY - 2c^tKY + c^tK^2c + \lambda c^tKc$$
$$\frac{\partial Loss}{\partial c} = -2KY + K^2c + \lambda Kc = 0$$
$$\Rightarrow \widehat{c} = \left(\lambda K + K^2\right)^{-1} KY = (K + \lambda I_N)^{-1}Y$$

(b) *(2 pts.)* In general, what is the effect of making λ larger or smaller? What is a reasonable approach to choosing λ?
When λ is small, $K + \lambda I_N \approx K$, so $\widehat{c} \approx K^{-1}Y$ and $||\widehat{c}||^2$ could be large, which means some element could be large. The line of real value versus fitted value could be sharp. When λ is large, $K + \lambda I_N$ is dominated by λI_N, so $||\widehat{c}||^2$ be small. The line of real value versus fitted value could be smooth. Usually I will use minimum cross validation to choose the λ.

4. **Kernel Ridge Regression: Foundation (18 points)**.
We are now going to practice deriving how you would have arrived at the 1 to begin with. An important interpretation of kernelization is that it effectively maps our examples X_i to a higher dimensional space, $\phi(X_i) \in \mathcal{R}^P$, and allows us to work with models that are linear in these new features (i.e. functions of the form $\phi(X_i)^\top\theta$ for $\theta \in \mathcal{R}^P$). Consider a positive semi-definite kernel $k(\cdot,\cdot)$ such that $k(X_i, X_j) = \langle\phi(X_i), \phi(X_j)\rangle$. Now, we will work with linear models in $\phi(X_i)$, specifically, $Y_i = \phi(X_i)^\top\theta + \epsilon_i$. Your goal is to now fit this model using ridge regression, i.e., find:

$$\underset{\theta\in\mathcal{R}^P}{argmin} \sum_{i=1}^{N} (Y_i - \phi(X_i)^\top\theta)^2 + \lambda||\theta||^2$$

where $||\theta||^2$ is simple $\theta^\top\theta$.

(a) *2 pts.)* Show that our model, $Y_i = \phi(X_i)^\top\theta + \epsilon_i$, implies $\mathbb{E}[Y_i|X_i] = \phi(X_i)^\top\theta$. State any assumptions you require on ϵ_i for this to hold.

Assumption: ϵ_i is a random noise, independent of choosing X_i, with mean 0.

$$\begin{aligned}\mathbb{E}[Y_i|X_i] &= \mathbb{E}[\phi(X_i)^\top\theta + \epsilon_i|X_i] \\ &= \mathbb{E}[\phi(X_i)^\top\theta|X_i] + \mathbb{E}[\epsilon_i|X_i] \\ &= \phi(X_i)^\top\theta + 0 = \phi(X_i)^\top\theta\end{aligned}$$

(b) *(8 pts.)* Next, show that in solving this minimization problem, you get $\theta = \sum_i c_i\phi(X_i)$ and specify what c_i is equal to in terms of the quantities in the minimization probolem.

we choose θ with square errors and L-2 norm penalty. That is

$$\underset{\theta}{arg\min}\left[(Y_i - \phi(X_i)^\top\theta)^2 + \lambda\theta^\top\theta\right]$$

$$\begin{aligned}&\frac{\partial Loss}{\partial\theta} = -2\sum_{i=1}^{N}\left[(Y_i - \phi(X_i)^\top\theta)\phi(X_i)\right] + 2\lambda\theta = 0 \\ &\Rightarrow \theta = \frac{1}{\lambda}\sum_{i=1}^{N}\left[(Y_i - \phi(X_i)^\top\theta)\phi(X_i)\right] = \sum_{i=1}^{N}c_i\phi(X_i) \qquad \text{where } c_i = \frac{Y_i - \phi(X_i)^\top\theta}{\lambda}\end{aligned}$$

(c) *(4 pts.)* Using this result, show that your model for $\mathbb{E}[Y_i|X_i]$ becomes

$$\sum_{j=1}^{N}c_jk(X_i, X_j)$$

.

$$\begin{aligned}\mathbb{E}[Y_i|X_i] &= \phi(X_i)^\top\theta \\ &= \phi(X_i)^\top\left(\sum_{j=1}^{N}c_j\phi(X_j)\right) \\ &= \sum_{j=1}^{N}\left[\phi(X_i)^\top\phi(X_j)c_j\right] \\ &= \sum_{j=1}^{N}c_jk(X_i, X_j)\end{aligned}$$

(d) *(4 pts.)* Show that for such a model, $||\theta||^2 = \mathbf{c}^\top \mathbf{K}\mathbf{c}$.

$$\begin{aligned}
||\theta||^2 &= \theta^\top \theta \\
&= \left(\sum_{i=1}^{N} c_j \phi(X_i)\right)^\top \left(\sum_{j=1}^{N} c_j \phi(X_j)\right) \\
&= \sum_{i,j=1}^{N} c_i c_j \phi(X_i)^\top \phi(X_j) \\
&= \sum_{i,j=1}^{N} c_i c_j k(X_i, X_j) \\
&= c^\top K c
\end{aligned}$$