9.520 Problem Set 2

Due Wednesday, April 14, 2010

Note: there are six problems total in this set.

Problem 1 One common preprocessing in machine learning is to center the data. In this problem we will see how this can be related to working with an (unpenalized) off-set term in the solution. Consider the usual Tikhonov regularization with a linear kernel, but assume that there is an unpenalized offset term b,

$$\min_{w\in\mathbb{R}^d,\, b\in\mathbb{R}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \big(\langle w, x_i\rangle + b - y_i\big)^2 + \lambda\|w\|^2 \right\}$$

and let (w^*, b^*) be the solution of the above problem.

For $i = 1, \ldots, n$, denote by $x_i^c = x_i - \bar{x}$, $y_i^c = y_i - \bar{y}$ the centered data, where $\bar{y}, \bar{x}$ are the output and input means respectively. Show that w^* also solves

$$\min_{w\in\mathbb{R}^d} \left\{ \frac{1}{n}\sum_{i=1}^{n} \big(\langle w, x_i^c\rangle - y_i^c\big)^2 + \lambda\|w\|^2 \right\},$$

and determine b^*.

Problem 2 In classification problems where the data are unbalanced (there are many more examples of one class than of the other one) a common strategy to obtain effective solution is *weighting* the loss function so that the errors in one class are counted more than errors in the other class. In the case of RLS this corresponds to solving the following problem

$$\min_{w\in\mathbb{R}^d} \left\{ \sum_{i=1}^{n} \gamma_i \big(\langle w, x_i\rangle - y_i\big)^2 + \lambda\|w\|^2 \right\}$$

where $\sum_{i=1}^n \gamma_i = 1$ and $\gamma_i > 0$ for all $i = 1, \ldots, n$.

(a) Derive the explicit form of the minimizer w^* of the above problem.

(b) Consider the case where we have a weighted loss function and we also an offset,

$$\min_{w\in\mathbb{R}^d,\, b\in\mathbb{R}} \left\{ \sum_{i=1}^{n} \gamma_i \big(\langle w, x_i\rangle + b - y_i\big)^2 + \lambda\|w\|^2 \right\}$$

where $\sum_{i=1}^n \gamma_i = 1$ and $\gamma_i > 0$ for all $i = 1, \ldots, n$. Using previous results, derive the explicit form of the minimizers w^*, b^* of the above problem.

(c) What do you think could be a good (optimal) way to choose the weights?
Note that this last question is admittedly a bit open-ended. There is not necessarily a right or wrong answer. We are just interested in seeing what ideas you might have.

Problem 3 Consider a bounded loss function $V : \mathbb{R} \times \mathbb{R} \to (0, M]$ and a hypothesis space comprised of N distinct functions, $\mathcal{H} = \{f_1, \ldots, f_N\}$.

(a) Prove that for all $\epsilon > 0$, the following bound holds

$$\Pr\Big(\sup_{f\in\mathcal{H}} \big|I_S[f] - I[f]\big| \geq \epsilon\Big) \leq \frac{CNM^2}{n\epsilon^2} \tag{1}$$

where $C > 0$ is some constant. What is the best C that you can get?
(Hint: use Chebychev's inequality and union bound)

(b) Show that, if f_S is the minimizer of the empirical risk on $\mathcal{H}$, then the above inequality implies that with probability $1 - \eta$ we have

$$I[f_S] \leq I_S[f_S] + \epsilon(n, \eta, N)$$

where $\epsilon(n, \eta, N) = \sqrt{\frac{CNM^2}{\eta n}}$ and $0 < \eta \leq 1$. Discuss the behavior of $I_S[f_S]$, $\epsilon(n, \eta, N)$ and their sum as functions of N.

(c) Denote with f_S and f^* the minimizers on $\mathcal{H}$ of the empirical and expected risks, respectively. Given (1), show that

$$I[f_S] - I[f^*] \leq 2\epsilon(n, \eta, N).$$

(Hint: add and subtract the empirical risks of f_S and f^* in the left hand side of the above inequality. Recall that by definition f_S minimizes the empirical risk.)

Problem 4 A reproducing kernel K is a *Mercer kernel* if it is continuous, symmetric, and positive semidefinite. Translation invariant kernels are those kernels given by $K(x, y) = k(x - y)$ where k is an even function on $\mathbb{R}^n$. In the context of RLS, we've seen that using the translation invariant Gaussian kernel leads to a solution which can be described as a superposition of "bumps" centered on the training points.

Now consider using a different kind of bump, defined by

$$K(x, y) = \begin{cases} 1 - \frac{|x-y|}{2} & \text{if } |x - y| \leq 2, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We would like to know if this is indeed a valid Mercer kernel.

(a) Sketch the bump in Equation (2) as a function of the variable $z = x - y$.

(b) Suppose $g \in L_2(\mathbb{R}^n)$ is continuous and even, and suppose the Fourier transform $\widehat{g}$ of g is nonnegative (that is, $\widehat{g}(\omega) \geq 0$). Show that under these conditions, the kernel $G(x, y) = g(x - y)$ is a Mercer kernel.
(Hint: use Fourier transforms)

(c) Given the result you proved in (b), show that the kernel K defined in Equation (2) above is a Mercer kernel.

Problem 5 In this excercise you will implement and use regularized least squares with an un-regularized bias term and weighted loss function, in an artificial classification problem with unbalanced classes. You should use your solution for Problem 2 in this problem.

Weighted RLS. You will modify your function `rlsTrain` from the previous problem set, now to take a new parameter `gamma`, which is a vector of weights to use in weighting the loss function. More specifically:

- `rlsTrain(Ytrain,Xtrain,whichKernel,lambda,gamma)` takes five inputs:
  - `Ytrain` the training labels;
  - `Xtrain` the training inputs;
  - `whichKernel` the kernel to use, e.g. `'linear'`; for this problem, *your function need only implement the weighted loss and bias term for the linear kernel.*
  - `lambda` the regularization parameter;
  - `gamma` the vector of weights on the corresponding terms in the loss function, of the same length as the number of training inputs

  and returns two outputs
  - `coeffs` the optimal RLS coefficients
  - `b` the optimal RLS bias

In addition, you will write a function `rlsHoldOut(Ytrain,Xtrain,Yvalidation,Xvalidation,lambdas,gamma)` that evaluates the weighted error on a validation set, for multiple values of the regularization parameter:

- `rlsHoldOut(Ytrain,Xtrain,Yvalidation,Xvalidation,lambdas,gamma)` takes 6 inputs:
  - `Ytrain` the training labels;
  - `Xtrain` the training inputs;
  - `Yvalidation` the validation labels;
  - `Xvalidation` the validation inputs;
  - `lambdas` a vector of values to try for the regularization parameter λ
  - `gamma` the vector of weights on the corresponding terms in the loss function, of the same length as the number of validation inputs

  and returns one output
  - `valerr` a vector of (weighted) errors on the validation set – one error value for each RLS solution corresponding to a value of λ in `lambdas`
  - (You can also return the learned coefficients, if you'd like.)

Data Generation. You will generate a toy dataset for binary classification, in which the points are sampled from a mixture of two Gaussians. The points of class -1 are sampled from a two-dimensional Gaussian with mean $(-2, 0)$ and the points of class $+1$ are sampled from a two-dimensional Gaussian with mean $(1, 0)$ – both Gaussians should be spherical with variance 1. To play with the balance of the data, we will vary the probability of sampling from the positive versus the negative class. Write a function `SampleGaussians2D(nPoints,pPos)` to do this:

- `SampleGaussians2D(nPoints,pPos)` takes two arguments
  - `nPoints` the total number of points to sample;
  - `pPos` the probability of each point being in the positive class (they should be i.i.d.)

  and returns two outputs

– `X` a $nPoints \times 2$ matrix in which each row gives one sampled point;
– `Y` a vector of length `nPoints` giving the labels corresponding to the rows in `X`.

Experimental protocol. We want to see the effect of the un-regularized bias term and the weights on the loss function, individually and combined, when we vary the balance of the data. First, you should sample a single test set, balanced between the classes. Then sample both a training set and validation set, for each of a number of settings of `pPos`, the probability of the positive class. Vary the probability of the positive class from 0.5 to (nearly) 1.

(a) You should produce four tables of the following form:

`pPos`	bias	weights	best λ	training accuracy	validation accuracy	test accuracy
0.5	0.12	...	2.78	0.96	0.83	0.79
		...				

Produce one table for each of:

- Linear RLS;
- Linear RLS with a bias term;
- Linear RLS with a weighted loss function;
- Linear RLS with both a bias term and a weighted loss function.

Note:

- For each experiment, you should be choosing the value of λ to minimize the weighted loss on the corresponding validation set.
- If $f^*(\cdot)$ is the optimal function chosen by RLS, the classification function should be $\text{sgn}(f^*(\cdot))$.
- Please report the accuracy on the training, validation and test sets all in terms of the percentage of points correctly classified.
- For the 'weights' column, please describe their numerical values as concisely as possible.

(b) You should also produce five figures:

- One figure for each of the tables, showing the test set and the decision boundaries for the different settings of `pPos`. Use the `legend` function to label the different decision boundaries. Matlab's `contour` function might be helpful here.
- One figure plotting the test set accuracy (percentage of points correctly classified) versus `pPos` for each of the four tables. (Label the curves, please.)

Weights. If N, N^-, and N^+ are the total number of points, the number of points in class -1, and the number in $+1$, respectively, you might choose the weights for the weighted loss to be $\gamma_+ = N^-/N$ for class $+1$ and $\gamma_- = N^+/N$ for class -1. Feel free to experiment with other weighting schemes as well.

Writeup. Please include the four tables and the five figures above, as well as all of your code.

Problem 6 In this problem, we will look at gradient descent first as a regularized algorithm based on early-stopping and then as a tool for solving RLS problems. We also recall the successive approximation scheme associated to a contractive map.

Recall that, by the fixed point theorem (see e.g. [1] or [2]), if $c^* \in \mathbb{R}^n$ satisfies

$$c^* = T(c^*),$$

where the map T is a contraction, i.e. $\|Tc - Tc'\|_2 \leq L\|c - c'\|_2$ with $0 < L < 1$, for all c, c', then the iteration

$$c_i = T(c_{i-1}) \tag{3}$$

with $c_0 = 0$ converges to c^*. Here we have temporarily used subscripts to denote iterates (not elements).

We have seen that the solution of ERM on a RKHS can be written as $f = \sum_{i=1}^n c_i K(\cdot, x_i)$ where the x_i belong to the training set. Then the ERM problem can be written as

$$\min_{c \in R^n} \|Y - \mathbf{K}c\|_2^2 \tag{4}$$

where $\mathbf{K}$ is the $(n \times n)$ kernel matrix and c, Y are the (n-dimensional) vectors of coefficients and labels respectively.

By assuming that $n \geq \|\mathbf{K}\|$ = maximum eigenvalue of the kernel matrix (always true for the Gaussian kernel), we can ensure convergence of the iterative procedures described below.

(a) Prove that if c^* minimizes the empirical risk in (4), if and only if it also satisfies $c^* = T(c^*)$ with

$$T(c) := c - \frac{1}{n}(\mathbf{K}c - Y). \tag{5}$$

(b) Implement in Matlab the iteration (3) for the particular map given in (5). Construct your own toy dataset, and show empirically that if the number of iterations is chosen appropriately (that is, if we choose a good early stopping point), we can avoid overfitting. Use the Gaussian kernel with sigma equal to the average distance between the points in the training set. We could use other kernels of course, but the Gaussian kernel will conveniently allow us to define a stable learning rate of the form chosen in (5).

(c) Now recall Tikhonov regularization problem

$$\min_{c \in \mathbb{R}^n} \{\|Y - \mathbf{K}c\|_2^2 + \lambda c^t \mathbf{K} c\}. \tag{6}$$

Show that c_λ^* solves the problem (6) if and only if it also satisfies $c_\lambda^* = T(c_\lambda^*)$ with

$$T_\lambda(c) := c - \frac{1}{n + \lambda}((\mathbf{K} + \lambda I)c - Y). \tag{7}$$

(d) Implement in Matlab the iteration (3) for the map (7), and evaluate its behavior on the same dataset you used in (b). Discuss the (empirical) interplay between the number of iterations and the parameter λ, and provide evidence (plots etc.) to support your claims. For all experiments, use the Gaussian kernel with sigma chosen to be the average distance between the points in the training set.

References

[1] J. Hunter and B. Nachtergaele, *Applied Analysis*, World Scientific, 2001. (available online – see ch.3: http://www.math.ucdavis.edu/∼hunter/book/pdfbook.html)

[2] A. N. Kolmogorov and S. V. Fomin. *Elements of the Theory of Functions and Functional Analysis*, Dover Publications, 1999.

[3] V. N. Vapnik. Statistical Learning Theory. Wiley, 1998.