MIT 9.520: Statistical Learning Theory **Fall 2014**

Problem Set 1

Student: Brando Miranda

Problem 1 The optimal solution to the Tikhonov regularization with offset (but not penalized) and linear kernel must satisfy the following constraints for minimization/optimality:

$$\frac{d}{db}\frac{1}{n}\sum_{i=1}^{n}(\langle w, x_i\rangle_{\mathbb{R}^d} + b - y_i)^2 + \lambda||w||^2_{\mathbb{R}^d} = 0$$

$$b = \frac{1}{n}\sum_{i=1}^{n}(y_i - \langle w, x_i\rangle_{\mathbb{R}^d}) = \frac{1}{n}\sum_{i=1}^{n} y_i - \langle w, \frac{1}{n}\sum_{i=1}^{n} x_i\rangle_{\mathbb{R}^d}$$

let $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ then b:

$$b = \bar{y} - \langle w, \bar{x}\rangle_{\mathbb{R}^d}$$

If the above must be satisfy, then we can just "remove" the dependence of the offset b, of the original minimization by substituting the above value of b. That leads to the following:

$$min_{w\in\mathbb{R}^d}\{\frac{1}{n}(\langle w, x_i\rangle_{\mathbb{R}^d} + \bar{y} - \langle w, \bar{x}\rangle_{\mathbb{R}^d} - y_i)^2 + \lambda||w||^2_{\mathbb{R}^d})\}$$

by using linearity of the inner product and re-arranging the y terms, we get the following expression:

$$min_{w\in\mathbb{R}^d}\{\frac{1}{n}(\langle w, x_i - \bar{x}\rangle_{\mathbb{R}^d} - (y_i - \bar{y}))^2 + \lambda||w||^2_{\mathbb{R}^d})\}$$

$$min_{w\in\mathbb{R}^d}\{\frac{1}{n}(\langle w, x_i^c\rangle_{\mathbb{R}^d} - y_i^c)^2 + \lambda||w||^2_{\mathbb{R}^d})\}$$

Which if we just substitute the definitions of x_i^c and y_i^c, we notice that it is the same minimization problem as with centered data! Thus, the solution for w^* for the original problem with unpenalized offset, is the same as when the data is centered. QED.

Let X^c be defined as the data matrix but with the centered data as the rows, instead of the input data. Similarly define Y^c. Then we can express w^* in terms of X^c and Y^c

$$w^* = ((X^c)^T X^c + \lambda n I)^{-1}(X^c)^T Y^c$$

Which we can then plug back in to out original formulation for b from the beginning yields:

$$b = \bar{y} - \langle w^*, \bar{x}\rangle_{\mathbb{R}^d}$$

We can do this because the optimal value for w^* must also satisfy that the derivative wrt b is zero (which is the same as the above equation for b).

$$b = \bar{y} - \langle((X^c)^T X^c + \lambda n I)^{-1}(X^c)^T Y^c, \bar{x}\rangle_{\mathbb{R}^d}$$

$$b = \bar{y} - ((X^c)^T X^c + \lambda n I)^{-1}(X^c)^T Y^c)^T \bar{x}$$

Problem 2 Lets try to express the minimization problem in terms of matrices:

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

Let

$$\hat{\gamma} = \begin{pmatrix} \gamma_1 & 0 & 0 & 0 \\ 0 & \gamma_2 & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_n \end{pmatrix}$$

and lets try to re-write the minimization problem in terms of the above gamma matrix:

$$(\hat{\gamma}(Xw - Y)) \cdot (Xw - Y) + \lambda||w||^2_{\mathbb{R}^d}$$

$$(\hat{\gamma}(Xw - Y))^T(Xw - Y) + \lambda||w||^2_{\mathbb{R}^d}$$

$$\langle\hat{\gamma}(Xw - Y), Xw - Y\rangle_{\mathbb{R}^n} + \lambda||w||^2_{\mathbb{R}^d}$$

Lets try to minimize it now by taking the gradient wrt w:

$$\bigtriangledown_w \left(\langle\hat{\gamma}(Xw - Y), Xw - Y\rangle_{\mathbb{R}^n} + \lambda||w||^2_{\mathbb{R}^d}\right) = 0$$

Lets handle the inner product first:

$$\bigtriangledown_w \left(\langle\hat{\gamma}(Xw - Y), Xw - Y\rangle_{\mathbb{R}^n}\right)$$

By applying linearity of the inner product we get:

$$\langle\hat{\gamma}Xw - \hat{\gamma}Y, Xw - Y\rangle_{\mathbb{R}^n} = \langle\hat{\gamma}Xw, Xw\rangle_{\mathbb{R}^n} - \langle\hat{\gamma}Xw, Y\rangle_{\mathbb{R}^n} - \langle\hat{\gamma}Y, Xw\rangle_{\mathbb{R}^n} + \langle\hat{\gamma}Y, Y\rangle_{\mathbb{R}^n}$$

$$\bigtriangledown_w((\hat{\gamma}Xw)^T Xw - (\hat{\gamma}Xw)^T Y - Y^T\hat{\gamma}Xw + (\hat{\gamma}Y)^T Y + \lambda||w||^2_{\mathbb{R}^d}) = 0$$

$$2X^T\hat{\gamma}Xw - X^T\hat{\gamma}Y - Y^T\hat{\gamma}X + 2\lambda w = 0$$

$$X^T\hat{\gamma}Xw + \lambda w = Y^T\hat{\gamma}X$$

$$(X^T\hat{\gamma}X + \lambda I)w = Y^T\hat{\gamma}X$$

$$w^* = (X^T\hat{\gamma}X + \lambda I)^{-1}Y^T\hat{\gamma}X$$

b)

In part a) we discovered that centering the data is the same thing as solving the optimization problem for RLS with an unpenalized offset. Thus, using the same idea, we can re-write the weighted minimization problem (in question 2) in terms of weighted centered data and then use our solution for part 2a).

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

For optimality, we need to satisfy the following constraint/equation for b:

$$\frac{d}{db} \sum_{i=1}^{n} \gamma_i \langle (w, x_i\rangle_{\mathbb{R}^d} + b - y_i)^2 + \lambda ||w||^2_{\mathbb{R}^d} = 0$$

$$b = \sum_{i=1}^{n} \gamma_i y_i - \sum_{i=1}^{n} \gamma_i \langle w, x_i\rangle$$

$$b = \sum_{i=1}^{n} \gamma_i y_i - \langle w, \sum_{i=1}^{n} \gamma_i x_i\rangle$$

let the weighted means be defined as $\bar{y}_\gamma = \sum_{i=1}^{n} \gamma_i y_i$ and $\bar{x}_\gamma = \sum_{i=1}^{n} \gamma_i x_i$. Using that we can re-express the above as:

$$b = \bar{y}_\gamma - \gamma_i \langle w, \bar{x}_\gamma\rangle$$

Now, since this equation for b represents one of the constraints we must satisfy (i.e. the constraint saying to minimize b), we can just plug b back in to the original minimization problem and re-express it in terms of the weighted means $\bar{x}_\gamma$ and $\bar{y}_\gamma$. After that we can just express the solution using part 2a) Lets do it:

$$min_{w\in\mathbb{R}^d} \sum_{i=1}^{n} \gamma_i (\langle w, x_i\rangle_{\mathbb{R}^d} + \bar{y}_\gamma - \langle w^*, \bar{x}_\gamma\rangle_{\mathbb{R}^d} - y_i)^2 + \lambda ||w||^2_{\mathbb{R}^d}$$

$$min_{w\in\mathbb{R}^d} \sum_{i=1}^{n} \gamma_i (\langle w, x_i - \bar{x}_\gamma\rangle_{\mathbb{R}^d} - \bar{y}_\gamma - y_i)^2 + \lambda ||w||^2_{\mathbb{R}^d}$$

Let $\bar{x}_\gamma = x_i - \bar{x}_\gamma$ and let $\bar{y}_\gamma = y_i - \bar{y}_\gamma$. Also, let X^c_γ and Y^c_γ be the matrices containing the weighted centered data instead of the original inputs. Now lets express the minimization problem above with those:

$$min_{w\in\mathbb{R}^d, b\in\mathbb{R}} \sum_{i=1}^{n} \gamma_i (\langle w, \bar{x}_\gamma\rangle_{\mathbb{R}^d} - (\bar{y}_\gamma))^2 + \lambda ||w||^2_{\mathbb{R}^d}$$

Thus, because the above is in the same format as the minimization problem in part 2a) but with the weighted mean instead of the input data points, we can just use its solution using the weighted centered mean data points. The solutions look as follow:

$$w^* = ((X_\gamma^c)^T \hat{\gamma} X_\gamma^c + \lambda I)^{-1} (Y_\gamma^c)^T \hat{\gamma} X_\gamma^c$$

$$b = \bar{y}_\gamma - \langle w^*, \bar{x}_\gamma \rangle_{\mathbb{R}^d}$$

or

$$b = \bar{y} - ((X_\gamma^c)^T X_\gamma^c + \lambda n I)^{-1} (X_\gamma^c)^T Y_\gamma^c)^T \bar{x}$$

2c)

One way that one could consider doing the weighting is to have higher weights to the class that has less examples and lower to the one that has more examples. This is because of the following intuition. Imagine that we have a lot of wrong examples for the same data point that we happened to have too many times. In this case the error contribution from this one point (or set of points) would be extremely big. However, this might be undesirable because we already had an indication that we were wrong on it. i.e. the error on this data point my obscure our mistakes on other data points. Thus, to balance it out and make sure that our predictor doesn't get fixed in only fixing a few data points, we could distribute this weighting according the inverse proportion of data points. i.e. we can simply weight very similar data points less if we have them in high quantities, than different ones.

However, the weighting could be a subtle and important thing to tune, because, one could argue that, if we get few examples of one class it might be because that is how the underlying distribution is, and we might be imposing to large of a prior on what we believe the distribution actually is. Thus, we have to do this with care to not out way what the data is actually telling us. For example, we should be careful not to accidentally weight outliers to high just because we had very few of them.

Problem 3 a) We want to show that $d(\Phi(x), \Phi(x')) = d_k(x, x')$.

$$d(\Phi(x), \Phi(x')) = ||\Phi(x) - \Phi(x')|| = \sqrt{\langle \Phi(x) - \Phi(x'), \Phi(x) - \Phi(x') \rangle}$$

by linearity and symmetry of the inner product:

$$\sqrt{\langle \Phi(x), \Phi(x) \rangle - 2\langle \Phi(x), \Phi(x') \rangle + \langle \Phi(x'), \Phi(x') \rangle}$$

by the reproducing property of the kernel K we know $K(x, x) = \langle K_x, K_x \rangle = \langle \Phi(x), \Phi(x) \rangle$:

$$\sqrt{K(x, x) + K(x', x') - 2K(x, x')}$$

Thus:

$$d(\Phi(x), \Phi(x')) = d_k(x, x') = \sqrt{K(x, x) + K(x', x') - 2K(x, x')}$$

Which is only a function of the input vector x and does not need the explicit representation of the feature map $\Phi(x)$

b) Let x_+ denote positively label $y = +1$, X_+ the set containing them and $\mu_{I_+} = \frac{1}{n_+}\sum_{x_+ \in X_+} \Phi(x)$. Similarly, define x_-, X_- and μ_{I_-}. Let the average distance in the feature space be:

$$d(\Phi(x), \mu_{I_+}) = \sqrt{\langle \Phi(x) - \frac{1}{n_+}\sum_{x_+ \in X_+} \Phi(x_+), \Phi(x) - \frac{1}{n_+}\sum_{x_+ \in X_+} \Phi(x_+)\rangle}$$

by linearity and symmetry of inner products we get:

$$\sqrt{\langle \Phi(x), \Phi(x)\rangle - \frac{1}{n_+}\sum_{x_+ \in X_+} 2\langle \Phi(x), \Phi(x_+)\rangle - \frac{1}{n_+^2}\sum_{x_+ \in X_+}\sum_{x'_+ \in X_+} \langle \Phi(x'_+), \Phi(x_+)\rangle}$$

by the reproducing property of the kernel K we know $K(x,x) = \langle K_x, K_x\rangle = \langle \Phi(x), \Phi(x)\rangle$:

$$d(x, \mu_+) = \sqrt{K(x,x) - \frac{1}{n_+}\sum_{x_+ \in X_+} 2K(x, x_+) - \frac{1}{n_+^2}\sum_{x_+ \in X_+}\sum_{x'_+ \in X_+} K(x'_+, x_+)}$$

Therefore, I will use the previous distance function in the feature space $d(x, \mu_+)$ to express the classifier in terms of the distance to the mean.

The classifier in terms of the sign function and kernel products is then:

$$c(x) = sign(d(x, \mu_-) - d(x, \mu_+))$$

since $d(x, \mu_-)$ is only in terms of kernel products as expressed above, this is the classifier. The above equation tries to capture the intuition "choose the sign of whichever mean of the feature space you are closest to."

Problem 4 a) To check that the square loss function can be written as $\mathcal{L}(-yf(x))$ lets expand $\|f(x) - y\|^2$:

$$(y - f(x))^2 = (1 - 2yf(x) + f(x)^2)$$

but $y^2 = 1$ thus:

$$\mathcal{L}(-yf(x)) = (1 - 2yf(x) + (yf(x))^2)$$

To find the minimizer c(x) we need to minimize:

$$\mathbb{E}_{x,y}[(y - f(x))^2]$$

and specify the function that achieves this minimum. Lets find it by taking the derivative of the above wrt to f(x) and setting it to zero:

$$\frac{d}{df(x)}\mathbb{E}_x\mathbb{E}_{y|x}[(y - f(x))^2] = \mathbb{E}_x\frac{d}{df(x)}\mathbb{E}_{y|x}[(y - f(x))^2]$$

which can be minimized by finding the minimum of $\frac{d}{df(x)}\mathbb{E}_{y|x}[(y - f(x))^2]$:

$$\frac{d}{df(x)}\mathbb{E}_{y|x}[(y-f(x))^2] = \mathbb{E}_{y|x}\left[\frac{d}{df(x)}(y-f(x))^2\right] = 0$$

$$\mathbb{E}_{y|x}[2(y-f(x))] = 0$$

$$\mathbb{E}_{y|x}[y] = \mathbb{E}_{y|x}[f(x)]$$

$$\mathbb{E}_{y|x}[y] = f(x)\mathbb{E}_{y|x}[1]$$

$$\mathbb{E}_{y|x}[y] = f(x)$$

$$p_{y|x}(1|x) - p_{y|x}(-1|x) = f(x)$$

Since $p_{y|x}(1|x) + p_{y|x}(-1|x) = 1$ then:

$$2p_{y|x}(1|x) - 1 = f(x)$$

b) We want to solve:

$$f^*(x) = argmin_{f(x)}\mathbb{E}_{x,y}[e^{-yf(x)}]$$

$$\frac{d}{df(x)}\mathbb{E}_x\mathbb{E}_{y|x}[e^{-yf(x)}] = 0$$

Similar reasoning as the previous question we have:

$$\mathbb{E}_{y|x}\left[\frac{d}{df(x)}e^{-yf(x)}\right] = 0$$

$$\sum_{y\in\{1,-1\}} p_{y|x}(y|x)ye^{-yf(x)} = p_{y|x}(1|x)e^{-f(x)} - p_{y|x}(-1|x)e^{f(x)}$$

$$p_{y|x}(1|x) - p_{y|x}(-1|x)e^{2f(x)} = 0$$

$$p_{y|x}(1|x) = p_{y|x}(-1|x)e^{2f(x)}$$

$$\frac{p_{y|x}(1|x)}{p_{y|x}(-1|x)} = e^{2f(x)}$$

$$\frac{1}{2}log\left(\frac{p_{y|x}(1|x)}{p_{y|x}(-1|x)}\right) = f(x)$$

or

$$\frac{1}{2}log\left(\frac{p_{y|x}(1|x)}{1-p_{y|x}(1|x)}\right) = f(x)$$

c)

$$\frac{d}{df(x)}\mathbb{E}_{y|x}log(1+e^{-yf(x)})$$

$$\sum_{y\in\{-1,1\}}\frac{-ye^{-yf(x)}}{1+e^{-yf(x)}}p_{y|x}(y|x)=0$$

$$\frac{-e^{-1f(x)}}{1+e^{-f(x)}}p_{y|x}(1|x)+\frac{e^{f(x)}}{1+e^{f(x)}}p_{y|x}(-1|x)=0$$

$$\frac{e^{-1f(x)}}{1+e^{-f(x)}}p_{y|x}(1|x)=\frac{e^{f(x)}}{1+e^{f(x)}}p_{y|x}(-1|x)$$

$$e^{f(x)}p(-1|x)=p(1|x)$$

$$f(x)=log\left(\frac{p(1|x)}{p(-1|x)}\right)$$

$$\frac{1}{2}log\left(\frac{p_{y|x}(1|x)}{p_{y|x}(-1|x)}\right)=f(x)$$

d) Bayes decision rule is:

$$b(x)=sign(2p_{y|x}(1|x)-1)$$

From part a, b and c we have $f(x)$ expressed in terms of $p_{y|x}(1|x)$. Thus, we can just re-arrange those equations and make $p_{y|x}(1|x)$ the subject and therefore, express $p_{y|x}(1|x)$ as a function of $f(x)$. Then we can obviously plug them back to $b(x)$ that is a function of $p_{y|x}(1|x)$. To not bore you with the simple algebra I will just express the answers.

For part a) we have:

$$p_{y|x}(1|x)=\frac{f(x)+1}{2}$$

So the relation of the minimizer of the squared loss function to Bayes decision rule is:

$$\boxed{b(x)=sign(2\left(\frac{f(x)+1}{2}\right)-1)=sign(f(x))}$$

First lets express $p_{y|x}(1|x)$ interns of f(x):

$$p(1|x)=\frac{1}{1+e^{-f(x)}}$$

Now lets do the appropriate substitutions to Bayes decision rule from the predictors in b and c. For the logistic loss its:

$$\boxed{b(x)=sign\left(\frac{2}{1+e^{-f(x)}}-1\right)}$$

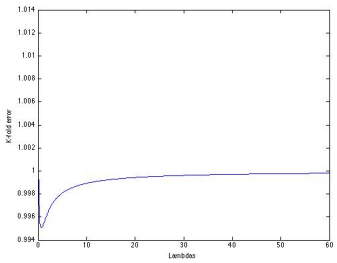

Figure 1.1. Lambdas vs k-fold error plot

For the exponential loss its:

$$p(1|x) = \frac{1}{1 + e^{-2f(x)}}$$

Which gives for the exponential loss:

$$\boxed{b(x) = sign\left(\frac{2}{1 + e^{-2f(x)}} - 1\right)}$$

Problem 5 (MATLAB) a)

$$\lambda = 0.6900$$

Training set:

$$accuracy_{trainingset} = 0.4700$$

Test set:

$$accuracy_{testset} = 0.5500$$

b)

$$\lambda = 0.0392$$

Considering that my algorithm for part a search for a lambda between 0 to 60 (max eigenvalue of kernel matrix), I would say 0.0392 is not that far from 0.69.

$$accuracy_{test} = 0.5500$$

Furthermore, I got that their accuracy on the test set is the same! Show the little difference between a 0.0392 lambda and 0.6900.

c)

$$\sigma = 0.2147$$

$$\lambda = 0.0127$$

$$accuracy_{train} = 1.0$$

$$accuracy_{test} = 0.9100$$

This one had a better train error. That should have been obvious because the RBK has much more complexity and can fit basically any data set given perfectly (it can overfit without being careful). However, from the accuracy result, we can say that the data was probably highly dimensional/complicated, and the RBK was able to realize that much better and thats why its accuracy is better. It could have overfitted, but probably didn't because of the regularization.