Assignment 1

Aleksandr Salo

Due March 3, 2015

1 Ridge regression (20 points)

1. Using vector notation, find a closed-form solution for the θ that minimizes the ridge regression cost function.

   Rules of derivation and trace operator that we are going to use:

   (a) $\frac{\partial}{\partial X}(Tr(X^T BX)) = BX + B^T X$. Note: if $B = I$ then $BX + B^T X = 2X$

   (b) $tr(X^T) = tr(X)$

   (c) $tr(ABC) = tr(CAB) = tr(BCA)$

   (d) $\nabla_A tr(AB) = B^T$

   Minimizing the following cost function:

$$J(\theta) = \frac{1}{2}(X\theta - y)^T (X\theta - y) + \frac{\lambda}{2}\theta^T \theta$$

   We first take the gradient With respect to θ:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \frac{1}{2}\nabla_\theta(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y) + \nabla_\theta(\frac{\lambda}{2}\theta^T\theta) \\
&= \frac{1}{2}\nabla_\theta(tr(\theta^T X^T X\theta) - 2tr(y^T X\theta)) + \frac{\lambda}{2}\nabla_\theta(tr(\theta^T\theta)) \\
&= \frac{1}{2}(X^T X\theta + X^T X\theta - 2X^T y) + \frac{\lambda}{2}(2\theta) \\
&= X^T X\theta - X^T y + \lambda\theta
\end{aligned}$$

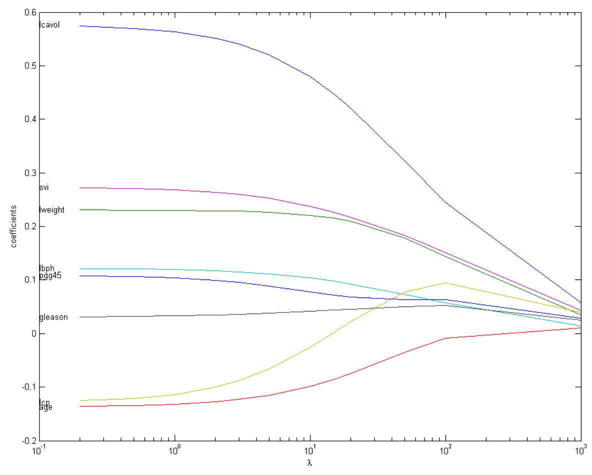
   Then setting gradient to zero:

$$\begin{gathered}
X^T X\theta + \lambda\theta = X^T y \\
(X^T X + \lambda I)\theta = X^T y \\
(X^T X + \lambda I)^{-1}(X^T X + \lambda I)\theta = (X^T X + \lambda I)^{-1} X^T y
\end{gathered}$$

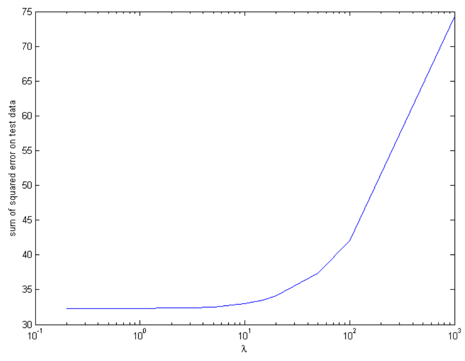
   We find θ that solves the equation:

$$\theta = (X^T X + \lambda I)^{-1} X^T y$$

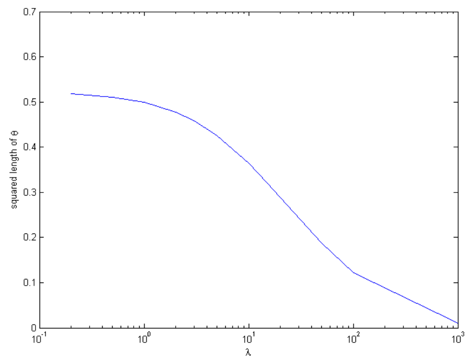
2. Implement ridge regression using the functions hwk ridge.m and hwk zscore.m on the prostate data provided. Show the effect of the parameter λ on three things:



(a) Values of the individual components of θ, λ is log scaled



(b) Non-regularized cost function



(c) Value of $\theta^T\theta$

(a) the values of the individual components of θ:
λ restricts the growth rate of θ components. The larger the value of λ, the greater the amount of shrinkage (that is to say that the coefficients become more robust to collinearity). Ultimately, with $\lambda = 1000$ for provided dataset, all coefficients converge to a single value of zero.

(b) the non-regularized cost function on held-out test data:
Naturally, with bigger λ and less diverse and less accurate θ cost function grows.
However, when looking at te detailed behavior, it's clear that for provided dataset cost function really start to grow only since $\lambda = 0.5$. Thus not significantly decreasing the accuracy of prediction by penalizing to high θ values.

(c) the value of $\theta^T\theta$:
Obviously, at the same time the squared length of theta decreases.

Thus ridge regression technique allows to penalize high theta coefficients (that appeared due to potential collinearity in the data) while with moderate values of parameter λ it would not significantly decrease predictive power (in other words would not increase cost function). By doing that we avoid blown-up theta values that potentially lead to calculus errors.

2 Gaussian Discriminant Analysis (30 points)

1. Find optimal μ, ϕ by maximizing log-likelihood of the following input data model:

$$p(x|y=0;\mu^0,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu^0)^T\Sigma^{-1}(x-\mu^0))$$
$$p(x|y=1;\mu^1,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(x-\mu^1)^T\Sigma^{-1}(x-\mu^1))$$
$$p(y;\phi) = \phi^y(1-\phi)^{1-y}$$

Which gives the following joint likelihood:

$$\begin{aligned} L(\phi,\mu^0,\mu^1,\Sigma) &= \prod_{i=1}^m p(x^{(i)},y^{(i)};\phi,\mu^0,\mu^1,\Sigma) \\ &= \prod_{i=1}^m p(x^{(i)}|y^{(i)};\mu^0,\mu^1,\Sigma)p(y^{(i)};\phi) \end{aligned}$$

Taking the log-likelihood allows us to break product into sum:

$$\ell(\phi,\mu^0,\mu^1,\Sigma) = \sum_{i=1}^m p(x^{(i)}|y^{(i)};\mu^0,\mu^1,\Sigma)p(y^{(i)};\phi)$$

Now we take partial derivatives and set it to zero to find the MLEs.
With respect to ϕ:

$$\begin{aligned} 0 &= \frac{\partial}{\partial\phi}\sum_{i=1}^m (y^{(i)}log\phi + (1-y^{(i)})log(1-\phi)) \\ &= \frac{1}{\phi}\sum_{i=1}^m y^{(i)} - \frac{1}{1-\phi}\sum_{i=1}^m (1-y^{(i)}) \\ \frac{1}{\phi}\sum_{i=1}^m y^{(i)} &= \frac{1}{1-\phi}\sum_{i=1}^m (1-y^{(i)}) \\ \phi\sum_{i=1}^m 1 - \phi\sum_{i=1}^m y^{(i)} &= \sum_{i=1}^m y^{(i)} - \phi\sum_{i=1}^m y^{(i)} \\ \phi &= \frac{\sum_{i=1}^m y^{(i)}}{m} \\ \phi_{optimal} &= \frac{1}{m}\sum_{i=1}^m 1\{y^{(i)}=1\} \end{aligned}$$

With respect to μ^0:

$$\begin{aligned}0 &= \frac{\partial}{\partial \mu^0}\sum_{i=1}^{m} -\frac{1}{2}(x^{(i)}-\mu^0)^T\Sigma^{-1}(x^{(i)}-\mu^0)\\ &= \sum_{i=1}^{m} -\frac{1}{2}(x^{(i)}-\mu^0)^T(\Sigma^{-1}+(\Sigma^{-1})^T)(-1)\\ &= \sum_{i=1}^{m} \Sigma^{-1}(x^{(i)}-\mu^0)^T\end{aligned}$$

Note, that:

(a) $\frac{\partial}{\partial x}x^TAx = x^T(A+A^T)$

(b) Covariance matrix Σ is symmetric around main diagonal, hence $\Sigma = \Sigma^T$

2. Prove that for two-class Gaussian Discriminant Analysis, with common covariance Σ, the posterior probability $p(y=1|x;\phi,\mu^0,\mu^1,\Sigma)$ is a logistic sigmoid function $g(z) = 1/(1+exp(-z))$.
   By Bayes' Rule:
   $$p(y=1|x) = \frac{p(x|y=1)p(y=1)}{p(x|y=1)p(y=1)+p(x|y=0)p(y=0)}$$
   Let:

   (a) $p(x|y=1)p(y=1) = A$

   (b) $-\frac{1}{2}(x-\mu^0)^T\Sigma^{-1}(x-\mu^0) = -z$

   (c) $\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} = c$

   Then:
   $$= \frac{A}{A+c\cdot e^{-z}}$$
   Now divide the entire fraction by A:
   $$= \frac{1}{\frac{A+c\cdot e^{-z}}{A}}$$
   Divide denominator by parts:
   $$= \frac{1}{1+c\cdot e^{-z}}$$
   Which is exactly sigmoid function.

3. Implementation of Gaussian discriminant analysis gives the following results:
   Base Rate Accuracy: 0.0909 which means, that if we are just to guess the class label we would have 9 % chance to guess correctly.
   Training Data Accuracy: 0.6818 that is to say, that when we try to predict on data that we used for learning, we get 68% chance of predicting correctly. That is on 59% better than the base case, which is significant learning result
   Test Data Accuracy: 0.4437 that is to say, that when we try to predict on the brand new data we get 44.5% chance of predicting correctly. That is on 36.5% better than the base case, yet it significantly worse than predicting on train data
   Per class accuracy: Not surprisingly, precision differs for different classes:

Class	Train Data	Test Data
1	0.6458	0.6667
2	0.5833	0.3810
3	0.8750	0.3810
4	0.7500	0.7857
5	0.6875	0.1667
6	0.4792	0.4524
7	0.6875	0.2619
8	0.7083	0.5476
9	0.6042	0.3571
10	0.6875	0.3095
11	0.7917	0.5714

3 Optimal margin classifier (10 points)

The optimal margin classifier is the one which maximizes the geometric margin separating examples from the two classes. This is defined as:

$$\min_{w,b} ||w||^2$$

$$\forall i : y^{(i)}(w^T x^{(i)} + b) \geq 1$$

1. Prove that w is orthogonal to the decision hyperplane (the hyperplane is the set of points $x|w^T x + b = 0$). Recall that two vectors u and v are orthogonal iff $u^T v = 0$.
   (a) Vector is orthogonal to a plane if it is orthogonal to a vector that belongs to a plane
   (b) Rewrite the equation of hyperplane: $w^T x = -b$
   (c) When $b = 0$, the hyperplane H is simply set of points that are orthogonal to w; when $b \neq 0$, the hyperplane is a translation, along the direction w, of that set.
   (d) For the equation above, w^T and b are given, and then different x's, that solve the equation, for the hyperplane H: if $x_0 \in H$ then for any other $x \in H$:

$$\begin{aligned} -b = w^T x_0 &= w^T x \\ -b - w^T x_0 &= w^T x - w^T x_0 \\ -b - (-b) &= w^T (x - x_0) \\ 0 &= w^T (x - x_0) \end{aligned}$$

   Thus a the hyperplane can be expressed as the set of vectors x s.t. $x - x_0$ is orthogonal to w
   (e) Taken into account this description, *any* vector from hyperplane is orthogonal to w, hence w is orthogonal to hyperplane itself.

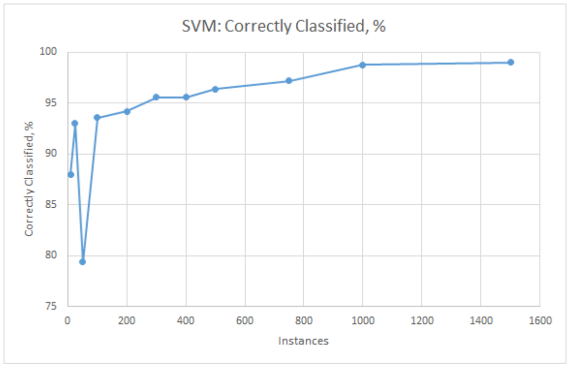
2. The equation $w^T x + b = 0$ forms a single hyperplane in the feature space of the variables x, with normal vector w. However, when thinking about minimizing the objective $w^T w$ subject to the m constraints $y^{(i)}(w^T x^{(i)} + b) \geq 1$, we said that each constraint forms a (different) hyperplane that separates the feasible and non-feasible space of solutions. Explain why each constraint's shape is a hyperplane, and why each of the m constraints makes a different hyperplane (when we have a single hyperplane for the decision boundary).
   (a) Since a hyperplane is a the set of points x s.t. $w^T x + b = 0$ if we have an m different equations $w^T x^{(i)} + b = 0$ we have m different hyperplanes (for m different x's).

(b) For the decision boundary we have a single hyperplane because we form it in the space of X thus making one optimal w.
On contrary, in space of w, each example of X forms its own hyperplane. Different x's hyperplanes intersect and form the space in which w could take its value to be optimal.
So answer is really is: because we have multiple x vectors and only one w vector.

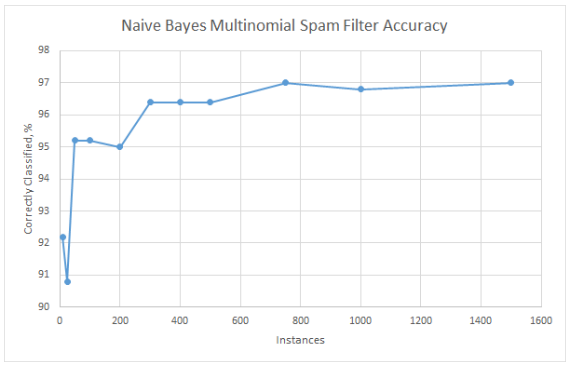
4 Naive Bayes and SVMs for Text Classiffication (20 points)

1. Naive Bayes Multinomial classifier performance using each of the different training files (but each time using the same test file) give the following error rate.

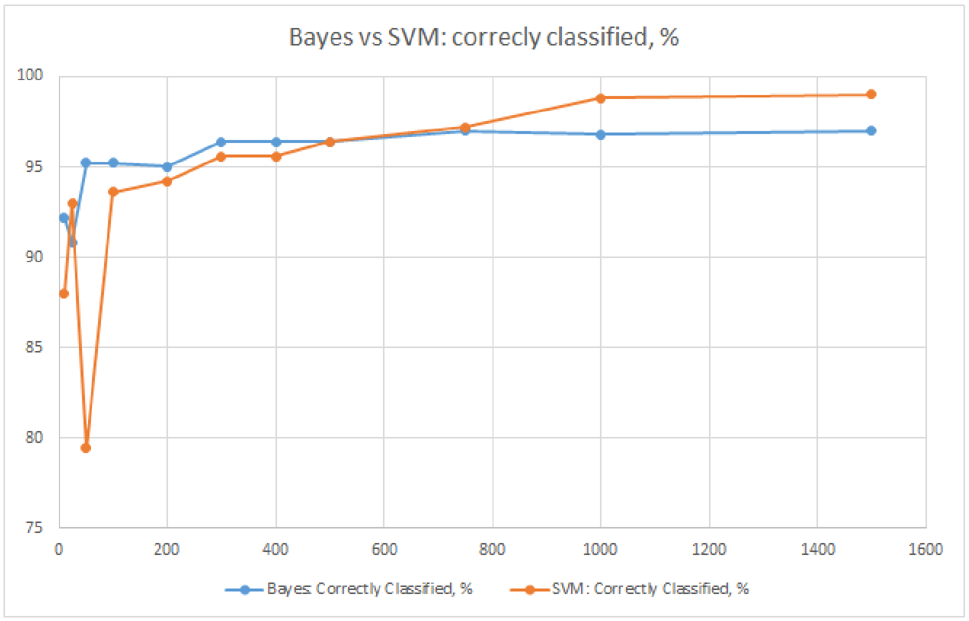
2. The same for SVM classifier + comparison of classifiers:



(a) Naive Bayes Spam Filter



(b) SVM Spam Filter



(c) Naive Bayes vs SVM accuracy

For Naive Bayes, accuracy is good from the beginning (with only 10 training examples), gets even better after 300 examples, and then don't improve significantly.

For SVM, although accuracy is good from the beginning, it seems like it less robust with small amount of examples. However, after 500 training examples, SVM significantly outperforms Bayes classified in prediction accuracy.

3. Consider the issue of estimating the multinomial probability of seeing a word w by using a training set with m examples.

If the training contains b(w) occurrences of w, then the MLE is:

$$P(w) = \frac{b(w)}{m}$$

If b(w)=0 then P(w) = 0, thus adding Laplace-smoothing:

$$P(w)_{laplace} = \frac{b(w) + 1}{m + d}$$

Thus, Laplace smoothing raises the probability of words that would have zero probability.
Intuitively, for other words in the dictionary it could significantly bring probability down. For example, we have a email which contains: "The quick brown fox", and our dictionary consists of 30000 words.
That gives:

$$P(''fox'') = \frac{1}{4}; P(''propagotary'') = \frac{0}{4} = 0$$

$$P(''fox'')_{laplace} = \frac{2}{30004}; P(''propagotary'')_{laplace} = \frac{1}{30000}$$

Prior the Laplace Smoothing, word "fox" was infinitly more likely to occur than "propagotary" according to our model. After smoothing, it became twice more likely to occur, which makes sense.