Homework 1, Problems 1 and 2

Name: Shashank Singh[1] [2]

10-715 Advanced Introduction to Machine Learning

Due: Thursday, Wednesday October 1, 2014

Collaborators: Bryan Hooi

1 Regression (Samy)

1.1 Multi-Task Regression

1. Since $\|Y - \Theta X\|_F^2 = \sum_{\ell=1}^k \|\vec{y}_\ell - \vec{\theta}_\ell^T X\|_F^2$ and no element of Θ is involved in more than one task, the total cost is minimized precisely by minimizing the cost on each individual task.

2. (a) Since $\|\Theta\|_F^2 = \sum_{\ell=1}^k \|\vec{\theta}_\ell\|_F^2$, the tasks are still independent. Also, as a norm, $\|\cdot\|_F^2$ is convex. These properties make J easy to minimize as k convex problems in d dimensions.

   (b) $\mathcal{C}$ couples the tasks, but, as a sum of norms, $\mathcal{C}$ is convex, so $\hat{\Theta}$ can still be found efficiently. Similar to the group lasso, $\mathcal{C}$ encourages columns of $\hat{\Theta}$ to be zero, and hence identifies components of X that are most informative.

   (c) The tasks are neither independent nor convex. However, by encouraging a low-rank $\hat{\Theta}$, $\mathcal{C}$ may identify certain tasks or components of X that are most informative, and hence allow for a sparse representation after a change of coordinates.

1.2 Shrinkage in Ridge Regression

1. Since the objective function is smooth and convex,

$$0 = \nabla_{\hat{\beta}} \|X\hat{\beta} - y\|_2^2 + \lambda\|\hat{\beta}\|_2^2 = 2(X\beta - y)^T X + 2\lambda\hat{\beta}^T.$$

   Hence, since U and V are orthogonal and Σ is diagonal,

$$\begin{aligned} y^T X = (X\hat{\beta})^T X + \lambda\hat{\beta} &= (U\Sigma V^T\hat{\beta})^T U\Sigma V^T + \lambda\hat{\beta}^T \\ &= \hat{\beta}^T V\Sigma U^T U\Sigma V^T + \lambda\hat{\beta}^T \\ &= \hat{\beta}^T V\Sigma^2 V^T + \lambda\hat{\beta}^T = \hat{\beta}^T\Sigma^2 + \lambda\hat{\beta}^T = (\Sigma^2 + \lambda I)\hat{\beta}^T. \end{aligned}$$

   Thus, noting that both λ and the singular values of X are nonnegative (so $\Sigma^2 + \lambda I$ is invertible),

$$x_*^T\hat{\beta} = \hat{\beta}^T x_* = (\Sigma^2 + \lambda I)^{-1} y^T U\Sigma V^T V z_* = (\Sigma^2 + \lambda I)^{-1} y^T U\Sigma z_* = \sum_{i=1}^d \frac{z_{*i}\sigma_i}{\sigma_i^2 + \lambda} u_i^T y. \quad \blacksquare$$

2. Without the L_2 penalty (i.e., when $\lambda = 0$), $x_*^T\hat{\beta} = \sum_{i=1}^d \frac{z_{*i}}{\sigma_i} u_i^T y$, and so the prediction is highly sensitive to small singular values of X (e.g., when covariates are correlated).

[1] sss1@andrew.cmu.edu

[2] Machine Learning Department & Department of Statistics

1.3 Local/Weighted Linear Regression

1. Since the objective function is smooth and convex,

$$0 = \nabla_{\hat{\beta}} \sum_{i=1}^{n} w_i(x)(y_i - \hat{\beta} x_i)^2 = 2 \sum_{i=1}^{n} w_i(x)(\hat{\beta} x_i - y_i) x_i = 2(X\hat{\beta} - y)^T W X.$$

   Solving for $\hat{\beta}$ gives $\hat{\beta} = (X^T W X)^{-1} X^T W y$, so $\hat{f}(x) = x^T \hat{\beta} = x^T (X^T W X)^{-1} X^T W y$. ■

2. Since the objective function is smooth and convex,

$$0 = \frac{d}{d\hat{\theta}} \sum_{i=1}^{n} w_i(x)(\hat{\theta} - y_i)^2 = 2 \sum_{i=1}^{n} w_i(x)(\hat{\theta} - y_i).$$

   Solving for $\hat{\theta}$ gives $\hat{f}(x) = \hat{\theta} = \frac{\sum_{i=1}^{n} w_i(x) y_i}{\sum_{i=1}^{n} w_i(x)}$, which is the Nadaraya-Watson Estimator. Since the $w_i(x) / \sum_{i=1}^{n} w_i(x)$ coefficients are non-negative and add to 1, $\hat{f}(x)$ is a convex combination of y_i's, and so $\hat{f}(x) \in (\min_i y_i, \max_i y_i)$. ■

1.4 Least Norm Solution

1. Let $\boxed{\hat{\beta} := X^T (X X^T)^{-1} y}$ (noting that, since X has full rank, $X X^T$ is invertible). Then, $X\hat{\beta} = X X^T (X X^T)^{-1} y = y$. Also, for all β with $y = X\beta$,

$$(\beta - \hat{\beta})^T \hat{\beta} = (\beta - \hat{\beta})^T X^T (X X^T)^{-1} y = (X\beta - X\hat{\beta})(X X^T)^{-1} y$$

   By the Pythagorean Theorem, $\|\beta\|_2^2 = \|\beta - \hat{\beta}\|_2^2 + \|\hat{\beta}\|_2^2 \geq \|\beta\|_2^2$, so $\hat{\beta}$ is the least norm solution.

2. Note that, since $X X^+$ is symmetric and $X X^+ X = X$,

$$(X\beta)^T X = (X X^+ y)^T X = y^T X X^+ X = y^T X.$$

   Since $\|X\beta - y\|_2^2$ is smooth and convex in β, to see that $\hat{\beta}$ minimizes $\|X\beta - y\|_2^2$, is suffices to observe that

$$\nabla_\beta \|X\beta - y\|_2^2 \big|_{\beta = \hat{\beta}} = 2(X\hat{\beta} - y)^T X = 2(X\beta)^T X - 2y^T X = 0.$$

   Now suppose $\beta \in \mathbb{R}^d$ also minimizes $\|y - X\beta\|_2^2$. Then, both $X\beta$ and $X\hat{\beta}$ are the projection of y onto the column space of X, so that $X\beta = X\hat{\beta}$, and hence $\beta - \hat{\beta}$ is in the null space of X. Since $\hat{\beta} = X^+ y$ is in the column space of X^+, which is the column space of X^T, $\hat{\beta}$ is orthogonal to $\beta - \hat{\beta}$. Thus, by the Pythagorean Theorem,

$$\|\beta\|_2^2 = \|\beta - \hat{\beta} + \hat{\beta}\|_2^2 = \|\beta - \hat{\beta}\|_2^2 + \|\hat{\beta}\|_2^2 \geq \|\hat{\beta}\|_2^2.$$

   Hence $\hat{\beta}$ is also the least-norm minimizer of $\|y - X\beta\|_2$. ■

2 Pólya Discriminant Analysis (Samy)

2.1 Model

1. Note that, whenever $i \neq j$, $(x_i, y_i) \perp\!\!\!\perp (x_j, y_j) | \theta, \alpha_1, \ldots, \alpha_K$, and that $x_i \perp\!\!\!\perp \theta, \alpha_j | y_i$. Hence,

$$\begin{aligned} L(\theta, \alpha_1, \ldots, \alpha_K) &= p(x_1, \ldots, x_n, y_1, \ldots, y_n | \theta, \alpha_1, \ldots, \alpha_K) \\ &= \prod_{i=1}^{n} p(x_i, y_i | \theta, \alpha_1, \ldots, \alpha_K) \\ &= \prod_{i=1}^{n} p(x_i | y_i, \theta, \alpha_1, \ldots, \alpha_K) p(y_i | \theta, \alpha_1, \ldots, \alpha_K) = \prod_{i=1}^{n} p(x_i | \alpha_{y_i}) p(y_i | \theta). \end{aligned}$$

by definition of condition probability. Since the Dirichlet distribution is the conjugate prior to the multinomial distribution, given x_i and α_{y_i}, $p \sim \text{Dir}(\alpha_{y_i} + x_i)$. Hence, using the definition of conditional probability and then plugging in the appropriate probability density functions and simplifying,

$$p(x_i | \alpha_{y_i}) = \frac{p(x_i | p, \alpha_{y_i}) p(p | \alpha_{y_i})}{p(p | x_i, \alpha_{y_i})} = \frac{\Gamma\left(\sum_{j=1}^{V} \alpha_{y_i,j} + x_{i,j}\right)}{\Gamma\left(\sum_{j=1}^{V} \alpha_{y_i,j}\right)} \prod_{j=1}^{V} \frac{\Gamma(\alpha_{y_i,j})}{\Gamma(\alpha_{y_i,j} + x_{i,j})}$$

Thus,

$$\begin{aligned} \ell(\theta, \alpha_1, \ldots, \alpha_K) &= \log L(\theta, \alpha_1, \ldots, \alpha_K) \\ &= \sum_{i=1}^{n} \log \theta_{y_i} + \log \Gamma\left(\sum_{j=1}^{V} \alpha_{y_i,j} + x_{i,j}\right) - \log \Gamma\left(\sum_{j=1}^{V} \alpha_{y_i,j}\right) \\ &\quad + \sum_{j=1}^{V} \log \Gamma(\alpha_{y_i,j}) - \log \Gamma(\alpha_{y_i,j} + x_{i,j}). \end{aligned} \tag{1}$$

2. For $i \in \{1, \ldots, K\}$, let c_i denote the number of documents in category i. Then, the component of ℓ depending on θ is

$$\ell_\theta(\theta) = \sum_{i=1}^{K} c_i \log \theta_i = \left(\sum_{i=1}^{K-1} c_i \log \theta_i\right) + c_K \log\left(1 - \sum_{i=1}^{K-1} \theta_i\right),$$

where the last step follows from $\theta \in \Delta^{K-1}$. Since ℓ_θ is smooth and convex in θ, if $\hat{\theta}$ is the maximum likelihood estimator of θ, for each $i \in \{1, \ldots, K-1\}$,

$$0 = \nabla_{\hat{\theta}_i} \ell_\theta(\hat{\theta}) = \frac{c_i}{\theta_i} - \frac{c_K}{1 - \sum_{i=j}^{K-1} \theta_j} = \frac{c_i}{\theta_i} - \frac{c_K}{\theta_K},$$

so that $\theta_i = c_i \theta_K / c_K$. Plugging this into $\sum_{i=1}^{K} \theta_i = 1$ and solving for θ_K gives $\theta_K = c_K / n$, and it follows that each $\boxed{\theta_i = c_i / n.}$

3. For each $r \in [n], s, t \in [V]$ with $(s \neq t)$, most terms in ℓ vanish upon differentiation by $\alpha_{i,j}$:

$$\frac{d\ell(\theta, \alpha_1, \ldots, \alpha_K)}{d\alpha_r^{(s)}} = \sum_{i:y_i=r}^{n} \psi\left(\sum_{j=1}^{V} \alpha_r^{(j)} + x_i^{(j)}\right) - \psi\left(\sum_{j=1}^{V} \alpha_r^{(j)}\right) + \psi\left(\alpha_r^{(s)}\right) - \psi\left(\alpha_r^{(s)} + x_i^{(s)}\right)$$

and so

$$\frac{d^2\ell(\theta, \alpha_1, \ldots, \alpha_K)}{\left(d\alpha_r^{(s)}\right)^2} = \sum_{i:y_i=r}^{n} \psi^{(1)}\left(\sum_{j=1}^{V} \alpha_r^{(j)} + x_i^{(j)}\right) - \psi^{(1)}\left(\sum_{j=1}^{V} \alpha_r^{(j)}\right) + \psi^{(1)}\left(\alpha_r^{(s)}\right) - \psi^{(1)}\left(\alpha_r^{(s)} + x_i^{(s)}\right)$$

and

$$\frac{d^2\ell(\theta, \alpha_1, \ldots, \alpha_K)}{d\alpha_r^{(s)} d\alpha_r^{(t)}} = \sum_{i:y_i=r}^{n} \psi^{(1)}\left(\sum_{j=1}^{V} \alpha_r^{(j)} + x_i^{(j)}\right) - \psi^{(1)}\left(\sum_{j=1}^{V} \alpha_r^{(j)}\right)$$

where ψ denotes the digamma function (derivative of $\log \circ \Gamma$) and $\psi^{(1)}$ denotes its derivative. In particular, if ℓ_{α_r} denotes the component of ℓ that depends on α_r, then the Hessian of ℓ_{α_r} is $H_{\alpha_r}\ell(\theta, \alpha_1, \ldots, \alpha_K) = D_{\alpha_r} + c_{\alpha_r} 1_V 1_V^T$, where

$$D_{\alpha_r} = \sum_{i:y_i=r} \operatorname{diag}\left(\psi(\alpha_r^{(1)}) - \psi(\alpha_r^{(1)} + x_i^{(1)}), \ldots, \psi(\alpha_r^{(V)}) - \psi(\alpha_r^{(V)} + x_i^{(V)})\right),$$

$$c_{\alpha_r} = \sum_{i:y_i=r}^{n} \psi^{(1)}\left(\sum_{j=1}^{V} \alpha_r^{(j)} + x_i^{(j)}\right) - \psi^{(1)}\left(\sum_{j=1}^{V} \alpha_r^{(j)}\right),$$

and $1_V \in \mathbb{R}^V$ denotes the V dimensional column vector of all ones. Since D_{α_r} is diagonal, D_{α_r} can be represented and inverted in $\tilde{O}(V)$ time and space. Thus, we can use the Sherman-Morrison inversion formula:

$$[H\ell_{\alpha_r}(\alpha_r)]^{-1} = \left(D_{\alpha_r} + c_{\alpha_r} 1_V 1_V^T\right)^{-1} = D_{\alpha_r}^{-1} - \frac{D_{\alpha_r}^{-1} c_{\alpha_r} 1_V 1_V^T D_{\alpha_r}^{-1}}{1 + c_{\alpha_r} 1_V^T D_{\alpha_r}^{-1} 1_V}.$$

Hence, since the Newton update is

$$\hat{\alpha}_r \to \hat{\alpha}_r - [H\ell_{\alpha_r}(\hat{\alpha}_r)]^{-1} \nabla \ell_{\alpha_r}(\hat{\alpha}_r) = \hat{\alpha}_r - \left(D_{\hat{\alpha}_r}^{-1} - \frac{D_{\hat{\alpha}_r}^{-1} c_{\hat{\alpha}_r} 1_V 1_V^T D_{\hat{\alpha}_r}^{-1}}{1 + c_{\hat{\alpha}_r} 1_V^T D_{\hat{\alpha}_r}^{-1} 1_V}\right) \nabla \ell_{\alpha_r}(\hat{\alpha}_r).$$

Note that, if the multiplication by $\nabla \ell_{\alpha_r}(\hat{\alpha}_r)$ is distributed over the subtraction, as long as D is stored as a V dimensional vector and all multiplications are performed from right to left, the above update requires only 8 vector arithmetic operations on V-dimensional vectors, while storing at most 2 V-dimensional vectors at a time. Hence, the update can be performed in $\tilde{O}(V)$ time and space. ■

4. I didn't have time to finish this. ☹

5. I didn't have time to finish this. ☹

2.2 Experiment

I didn't have time to get this working. ☹

Homework 1, Problems 3 and 4

Name: Shashank Singh[1] [2]

10-715 Advanced Introduction to Machine Learning

Due: Thursday, Wednesday October 1, 2014

3 Duality (Veeru)

3.1 Weak Duality

1. $L(x, \lambda, u) = f(x) + \lambda h_1(x) + u h_2(x)$.
2. $g(\lambda, u) = \inf_{x \in \mathbb{R}^d} L(x, \lambda, u) = f(x) + \lambda h_1(x) + u h_2(x)$.
3. For any primal feasible x and dual feasible λ and u, $h_1(x) \leq 0$, $h_2(x) = 0$, and $\lambda \geq 0$. Thus,

$$g(\lambda^*, u^*) \leq L(x^*, \lambda^*, u^*) = f(x^*) + \lambda h_1(x^*) + u h_2(x^*) \leq f(x^*). \quad \blacksquare$$

3.2 Optimal Coding

1. Since any exponential function and any sum of convex functions are necessarily convex, the first constraint is convex. Since the second constraint is trivially convex and the intersection of two convex sets is convex, the feasible region is convex. $\blacksquare$
2. Note that the objective is non-increasing in each component of x, and that, by the first constraint, any feasible x has only strictly positive components. Thus, for all $x \in \mathbb{R}^n_+$ x_i, unless equality holds in the first constraint, we can reduce the objective function by replacing x_i by $x_i - \varepsilon$ for some $\varepsilon > 0$ (since the first constraint is continuous in x). Hence, x is optimal only if the first constraint is tight. $\blacksquare$
3. By stationarity, $\exists u \in \mathbb{R}$ such that, for each $i \in \{1, \ldots, n\}$,

$$0 = \frac{d}{dx_i} \sum_{i=1}^n p_i x_i + u \left(\sum_{i=1}^n 2^{-x_i} - 1 \right) = p_i - u \ln(2) 2^{-x_i}.$$

Hence, $2^{-x_i} = \frac{p_i}{u \ln(2)}$. Since the sum of p_i's is 1,

$$1 = \sum_{i=1}^n 2^{-x_i} = \sum_{i=1}^n \frac{p_i}{u \ln(2)} = \frac{1}{u \ln(2)},$$

and so $u = 1/\ln(2)$. Thus, $\boxed{x_i = -\log_2 p_i.}$

[1] sss1@andrew.cmu.edu

[2] Machine Learning Department & Department of Statistics

4 SVM and Perceptron (Veeru)

4.1 Finding support vectors from dual variables

The KKT conditions give that, for some $\lambda_1, \ldots, \lambda_n, \mu_1, \ldots, \mu_n \leq 0$, $u \in \mathbb{R}$, for each $i \in [n]$, by Stationarity,

$$\begin{aligned}
0 &= \frac{1}{2}\sum_{j \neq i} \alpha_j y_i y_j \langle x_i, x_j \rangle + \alpha_i y_i^2 \|x_i\|^2 - 1 + u y_i + \lambda_i - \mu_i \\
&= \frac{1}{2} y_i \langle x_i, \sum_{j \neq i} \alpha_j y_j x_j \rangle + \alpha_i \|x_i\|^2 - 1 + u y_i + \lambda_i - \mu_i \\
&= \frac{1}{2} y_i \langle x_i, w - x_i \rangle + \alpha_i \|x_i\|^2 - 1 + u y_i + \lambda_i - \mu_i \\
&= \frac{1}{2} y_i \langle x_i, w \rangle + (\alpha_i - y_i/2)\|x_i\|^2 - 1 + u y_i + \lambda_i - \mu_i \\
&= \frac{1}{2} y_i (f(x_i) - b) + (\alpha_i - y_i/2)\|x_i\|^2 - 1 + u y_i + \lambda_i - \mu_i,
\end{aligned}$$

since $y_i f(x_i) = y_i \langle w, x_i \rangle + y_i b$. By complementary slackness, if $\alpha_i < C$, then $\mu_i = 0$, and so

$$y_i f(x_i) = b + (y_i - 2\alpha_i)\|x_i\|^2 + 2 - u y_i - \lambda_i \geq b + (y_i - 2\alpha_i)\|x_i\|^2 + 2 - u y_i,$$

since $\lambda_i \leq 0$. Similarly, if $\alpha_i > 0$, then $\lambda_i = 0$, so, since $\mu_i \leq 0$,

$$y_i f(x_i) = b + (y_i - 2\alpha_i)\|x_i\|^2 + 2 - u y_i + \mu_i \leq b + (y_i - 2\alpha_i)\|x_i\|^2 + 2 - u y_i.$$

Hence, it suffices to show $b + (y_i - 2\alpha_i)\|x_i\|^2 - u y_i = -1$. Didn't have time to go further. ☹

4.2 Using libsvm

See Figures 1, 2, and 3 below. Code is attached on the last page.

4.3 Mistake bound for Perceptron

1. Clearly, for $k = 0$, $\langle w_0, w_* \rangle = \langle 0, w_* \rangle = 0 = k\delta$. If $\langle w_k, w_* \rangle \geq k\delta$ for some $k \geq 0$, then,
   $\langle w_{k+1}, w_* \rangle = \langle w_k + y_{i(k+1)} x_{i(k+1)}, w_* \rangle = \langle w_k, w_* \rangle + y_{i(k+1)} \langle x_{i(k+1)}, w_* \rangle \geq k\delta + \delta = (k+1)\delta.$ ■

2. Clearly, for $k = 0$, $\|w_k\|^2 = 0 = kM^2$. If $\|w_k\|^2 \leq kM^2$ for some $k \geq 0$, then
   $$\begin{aligned}
   \|w_{k+1}\|^2 = \|w_k + y_{i(k+1)} x_{i(k+1)}\|^2 &= \|w_k\|^2 + 2\langle w_k, y_{i(k+1)} x_{i(k+1)} \rangle + |y_{i(k+1)}| \|x_{i(k+1)}\|^2 \\
   &= kM^2 + 2 y_{i(k+1)} \langle w_k, x_{i(k+1)} \rangle + M^2 \leq (k+1)M^2,
   \end{aligned}$$
   since w_k fails on $(y_{i(k+1)}, x_{i(k+1)})$, so that $2 y_{i(k+1)} \langle w_k, x_{i(k+1)} \rangle \leq 0$. ■

3. For all iterations k such that the algorithm still makes mistakes, by Cauchy-Schwarz,
   $$(k\delta)^2 \leq \langle w_k, w_* \rangle^2 \leq \|w_k\|^2 \|w_*\|^2 = \|w_k\|^2 \leq kM^2.$$
   Hence, $k_* \leq M^2/\delta^2$. ■

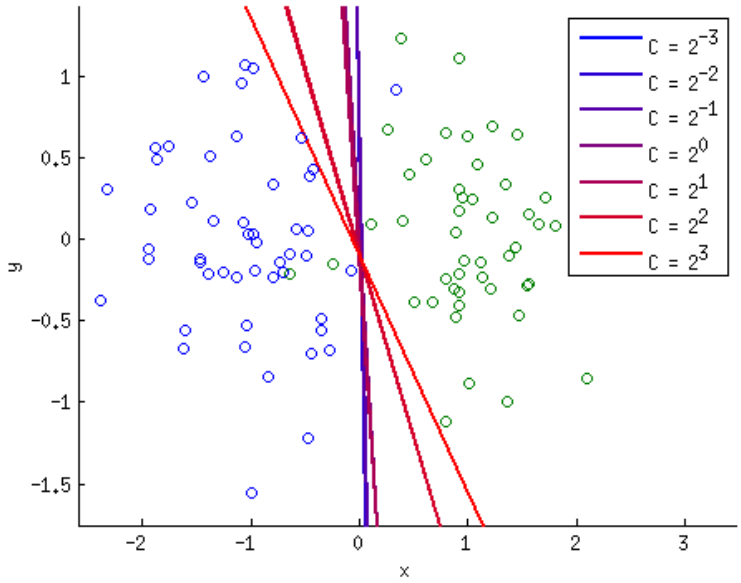

Figure 1: Decision boundaries of SVM for various values of C, over the training data.

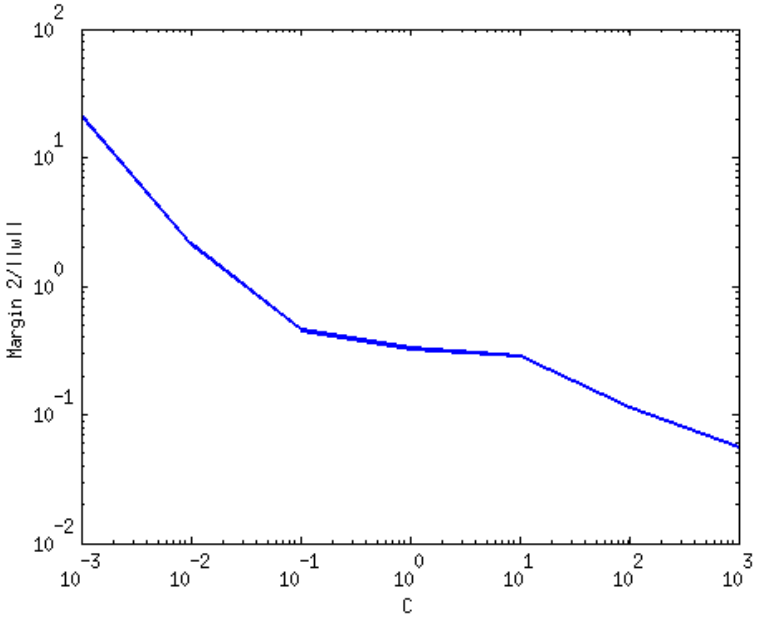

Figure 2: Log-log plot of width of SVM margin for various values of C.

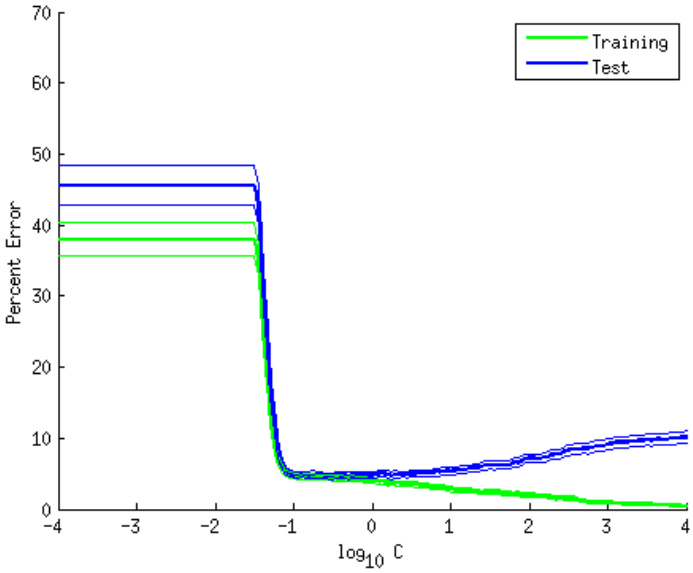

Figure 3: Semilog plot of test error and training error for various values of C, averaged over 50 trials, with standard error bands.