CAP 6610 Machine Learning, Spring 2022

Homework 3

1. (20%) *Monotonicity of loss and regularizer as the regularization parameter changes.* In regularized empirical risk minimization, we choose the parameter $\boldsymbol{\theta} \in \mathbb{R}^m$ to minimize the regularized empirical risk, $L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$, where $L(\boldsymbol{\theta})$ is the empirical risk, $r(\boldsymbol{\theta})$ is the regularizer, and $\lambda > 0$ is the regularization hyper-parameter. (The exact form of the functions L and r is irrelevant in this problem.) The hyper-parameter $\lambda > 0$ is used to trade off the two objectives, $L(\boldsymbol{\theta})$ and $r(\boldsymbol{\theta})$. Intuition suggests that as λ increases, $r(\boldsymbol{\theta})$ decreases while $L(\boldsymbol{\theta})$ increases. In this exercise we verify that this is the case.

   Suppose $0 < \lambda < \tilde{\lambda}$. Let $\boldsymbol{\theta}^\star$ minimize $L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$ and $\widetilde{\boldsymbol{\theta}}^\star$ minimize $L(\boldsymbol{\theta}) + \tilde{\lambda} r(\boldsymbol{\theta})$.

   (a) Show that $r(\boldsymbol{\theta}^\star) \geq r(\widetilde{\boldsymbol{\theta}}^\star)$. In other words, increasing λ will never make our regularization error larger.

   (b) Show that $L(\boldsymbol{\theta}^\star) \leq L(\widetilde{\boldsymbol{\theta}}^\star)$. In other words, increasing λ will never decrease our training error.

   Hint. Use the fact that $\boldsymbol{\theta}^\star$ is the minimizer of $L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$ and similarly for $\widetilde{\boldsymbol{\theta}}^\star$. This means that for any $\boldsymbol{\theta}$, we have $L(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}^\star) + \lambda r(\boldsymbol{\theta}^\star)$, and similarly for $\widetilde{\boldsymbol{\theta}}^\star$.

   Solution.

   (a) Since $\boldsymbol{\theta}^\star$ is the minimizer of $L(\boldsymbol{\theta})+\lambda r(\boldsymbol{\theta})$, we have for any $\boldsymbol{\theta}$, we have $L(\boldsymbol{\theta})+\lambda r(\boldsymbol{\theta}) \geq L(\boldsymbol{\theta}^\star)+\lambda r(\boldsymbol{\theta}^\star)$, and similarly for $\widetilde{\boldsymbol{\theta}}^\star$. Therefore

   $$L(\widetilde{\boldsymbol{\theta}}^\star) + \lambda r(\widetilde{\boldsymbol{\theta}}^\star) \geq L(\boldsymbol{\theta}^\star) + \lambda r(\boldsymbol{\theta}^\star) \tag{1}$$

   $$L(\boldsymbol{\theta}^\star) + \tilde{\lambda} r(\boldsymbol{\theta}^\star) \geq L(\widetilde{\boldsymbol{\theta}}^\star) + \tilde{\lambda} r(\widetilde{\boldsymbol{\theta}}^\star) \tag{2}$$

   Adding the two inequalities and rearrange, we get

   $$(\tilde{\lambda} - \lambda) r(\boldsymbol{\theta}^\star) \geq (\tilde{\lambda} - \lambda) r(\widetilde{\boldsymbol{\theta}}^\star).$$

   Since $\lambda < \tilde{\lambda}$, this means $r(\boldsymbol{\theta}^\star) \geq r(\widetilde{\boldsymbol{\theta}}^\star)$

   (b) Multiply (1) by $\tilde{\lambda}$ and (2) by λ gives

   $$\tilde{\lambda} L(\widetilde{\boldsymbol{\theta}}^\star) + \tilde{\lambda}\lambda r(\widetilde{\boldsymbol{\theta}}^\star) \geq \tilde{\lambda} L(\boldsymbol{\theta}^\star) + \tilde{\lambda}\lambda r(\boldsymbol{\theta}^\star) \tag{3}$$

   $$\lambda L(\boldsymbol{\theta}^\star) + \tilde{\lambda}\lambda r(\boldsymbol{\theta}^\star) \geq \lambda L(\widetilde{\boldsymbol{\theta}}^\star) + \tilde{\lambda}\lambda r(\widetilde{\boldsymbol{\theta}}^\star) \tag{4}$$

   Adding the two inequalities and rearrange, we get

   $$(\tilde{\lambda} - \lambda) L(\widetilde{\boldsymbol{\theta}}^\star) \geq (\tilde{\lambda} - \lambda) L(\boldsymbol{\theta}^\star).$$

   Again since $\lambda < \tilde{\lambda}$, this means $L(\boldsymbol{\theta}^\star) \leq L(\widetilde{\boldsymbol{\theta}}^\star)$

2. (30%) *MAP interpretation of regularized empirical loss minimization.* We have seen that some (unregularized) empirical loss minimization problems can be interpreted as maximum likelihood estimation (MLE) if we choose certain parametric form for the conditional probability $p(y|\boldsymbol{x};\boldsymbol{\theta})$. Assuming the data samples are i.i.d., MLE of $p(y|\boldsymbol{x};\boldsymbol{\theta})$ is equivalent to

   $$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \sum_{i=1}^{n} -\log p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}).$$

After some trivial transformations, we can recover some supervised learning models such as least squares regression and logistic classification.

Some statisticians, who call themselves Bayesians, believe that we should treat $\boldsymbol{\theta}$ as random as well, and impose probability distributions on them. In this case, the probability that we really care about is $p(\boldsymbol{\theta}|Y, \boldsymbol{X})$, the conditional probability of $\boldsymbol{\theta}$ given data $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ and $Y = \{y_1, \dots, y_n\}$. According to Bayes rule,

$$p(\boldsymbol{\theta}|Y, \boldsymbol{X}) = \frac{p(Y|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{X})}{p(Y|\boldsymbol{X})}.$$

Furthermore, it is common to assume that $\boldsymbol{\theta}$ is independent of $\boldsymbol{X}$ and $(\boldsymbol{x}_i, y_i)$ are i.i.d. conditioned on $\boldsymbol{\theta}$, leading to

$$p(\boldsymbol{\theta}|Y, \boldsymbol{X}) = \frac{p(\boldsymbol{\theta}) \prod_{i=1}^n p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})}{p(Y|\boldsymbol{X})}.$$

Here, $p(\boldsymbol{\theta})$ is called the prior (*a priori* in Latin), $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ is called the likelihood, and $p(\boldsymbol{\theta}|Y, \boldsymbol{X})$ is called the posterior (*a posteriori* in Latin).

Depending on the definition of the prior and the likelihood, the denominator $p(Y|\boldsymbol{X})$ may be very hard to evaluate. Instead, we can try to find a point estimate $\boldsymbol{\theta}$ that maximizes the posterior probability, which is called maximum *a posteriori* (MAP), since the denominator does not depend on $\boldsymbol{\theta}$ and can be omitted in maximization. This is equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \sum_{i=1}^n -\log p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}).$$

For each of the following cases, give an explicit MAP formulation for estimating $\boldsymbol{\theta}$. Find their relationship to the corresponding regularized empirical loss minimization problems. Specifically, give an exact expression for the regularization parameter λ in terms of the prior and likelihood distributions.

(a) $p(y|\boldsymbol{x}, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\phi}^\top \boldsymbol{\theta}, \sigma^2)$ and $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, \sigma_0^2 \boldsymbol{I})$;

(b) $p(y|\boldsymbol{x}, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\phi}^\top \boldsymbol{\theta}, \sigma^2)$ and $p(\boldsymbol{\theta})$ follows a multivariate Laplacian distribution:

$$p(\boldsymbol{\theta}) = \prod_{j=1}^m \frac{1}{2a} \exp\left(-\frac{|\theta_j|}{a}\right);$$

(c) $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \Pr[yu \geq 0]$ where $y = \pm 1$, $p(u|\boldsymbol{x}, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\phi}^\top \boldsymbol{\theta}, \sigma^2)$ and $p(\boldsymbol{\theta}) \sim \mathcal{N}(0, \sigma_0^2 \boldsymbol{I})$;

(d) $p(y|\boldsymbol{x}, \boldsymbol{\theta}) = 1/(1 + \exp(-y\boldsymbol{\phi}^\top \boldsymbol{\theta}))$ where $y = \pm 1$ and $p(\boldsymbol{\theta})$ follows a multivariate Laplacian distribution as in (b).

Solution. We fix the regularized empirical loss minimization formulation to be

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n} \sum_{i=1}^n \ell_i(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta}). \tag{5}$$

Sometimes the $1/n$ factor is dropped, but that basically corresponds to a regularization parameter $n\lambda$. Either way is acceptable.

(a) The log-likelihood of $p(y|\boldsymbol{x}, \boldsymbol{\theta})$ is

$$\log p(y|\boldsymbol{x}, \boldsymbol{\theta}) = -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y - \boldsymbol{\phi}^\top \boldsymbol{\theta})^2.$$

The log-likelihood of $p(\boldsymbol{\theta})$ is

$$\log p(\boldsymbol{\theta}) = -\frac{m}{2} \log(2\pi\sigma_0^2) - \frac{1}{2\sigma_0^2} \|\boldsymbol{\theta}\|^2.$$

Therefore, the MAP formulation is equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\phi}_i^\top\boldsymbol{\theta})^2 + \frac{\sigma^2}{n\sigma_0^2}\|\boldsymbol{\theta}\|^2.$$

This is ridge regression with $\lambda = \sigma^2/(n\sigma_0^2)$.

(b) The log-likelihood of $p(y|\boldsymbol{x},\boldsymbol{\theta})$ is the same as part (a). The log-likelihood of $p(\boldsymbol{\theta})$ is

$$\log p(\boldsymbol{\theta}) = -m\log(2a) - \frac{1}{a}\|\boldsymbol{\theta}\|_1.$$

Therefore, the MAP formulation is equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\phi}_i^\top\boldsymbol{\theta})^2 + \frac{\sigma^2}{na}\|\boldsymbol{\theta}\|_1.$$

This is lasso regression with $\lambda = \sigma^2/(na)$.

(c) The log-likelihood of $p(y|\boldsymbol{x},\boldsymbol{\theta})$ is

$$\begin{aligned}\log p(y|\boldsymbol{x},\boldsymbol{\theta}) &= \log\int_0^{+\infty}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(u-y\boldsymbol{\phi}^\top\boldsymbol{\theta})^2}{2}\right)du\\ &= \log\int_{-\infty}^{y\boldsymbol{\phi}^\top\boldsymbol{\theta}}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{u^2}{2}\right)du\\ &= \log\varPhi(y\boldsymbol{\phi}^\top\boldsymbol{\theta}),\end{aligned}$$

where $\varPhi(\cdot)$ is the cumulative distribution function (CDF) of $\mathcal{N}(0,1)$. The log-likelihood of $p(\boldsymbol{\theta})$ is the same as in part (a). The MAP formulation is equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}-\log\varPhi(y_i\boldsymbol{\phi}_i^\top\boldsymbol{\theta}) + \frac{1}{n\sigma_0^2}\|\boldsymbol{\theta}\|^2.$$

This is L_2 regularized probit regression with $\lambda = 1/(n\sigma_0^2)$.

(d) The log-likelihood of $p(y|\boldsymbol{x},\boldsymbol{\theta})$ is

$$\log p(y|\boldsymbol{x},\boldsymbol{\theta}) = -\log(1+\exp(-y\boldsymbol{\phi}^\top\boldsymbol{\theta})).$$

The log-likelihood of $p(\boldsymbol{\theta})$ is the same as in part (b). The MAP formulation is equivalent to

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i\boldsymbol{\phi}_i^\top\boldsymbol{\theta})) + \frac{1}{na}\|\boldsymbol{\theta}\|^2.$$

This is L_1 regularized logistic regression with $\lambda = 1/(na)$.

3. (20%) *Proximal operator for the group lasso regularizer.* In this exercise we derive the proximal operator for the group lasso regularizer. We will be using the notion of subgradient to make the derivation more solid.

Recall that the proximal operator of the function $\rho\sum_{j=1}^{m}\|\boldsymbol{\Theta}_j\|$ at a particular point $\widetilde{\boldsymbol{\Theta}}$, where $\boldsymbol{\Theta}_j$ is the jth row of $\boldsymbol{\theta}$, is the solution to the following optimization problem

$$\underset{\boldsymbol{\Theta}}{\text{minimize}} \quad \rho\sum_{j=1}^{m}\|\boldsymbol{\Theta}_j\| + \frac{1}{2}\|\boldsymbol{\Theta} - \widetilde{\boldsymbol{\Theta}}\|^2.$$

Notice that the problem is separable over $\boldsymbol{\Theta}_j$, so it suffices to consider the simpler problem

$$\underset{\boldsymbol{\Theta}_j}{\text{minimize}} \quad \rho\|\boldsymbol{\Theta}_j\| + \frac{1}{2}\|\boldsymbol{\Theta}_j - \widetilde{\boldsymbol{\Theta}}_j\|^2. \tag{6}$$

(a) Give an expression for the subdifferential of the function $\|\boldsymbol{\Theta}_j\|$. Recall that subdifferential is the set of subgradients, so you need to list all possible subgradients. *Hint.* The function is only non-differentiable when $\boldsymbol{\Theta}_j = 0$; when the function is differentiable, the only element in the subdifferential is the gradient; otherwise it is some convex set.

(b) A necessary and sufficient condition for optimality is that 0 is an element of the subdifferential. Use this condition and the expression of the subdifferential to derive the solution of (6).

Solution.

(a) The norm function is differentiable when the argument is nonzero; consider a k-vector $\boldsymbol{x}$, the partial derivative of $\|\boldsymbol{x}\| = (x_1^2 + \cdots + x_k^2)^{1/2}$ w.r.t. x_i is

$$\frac{\partial\|\boldsymbol{x}\|}{\partial x_i} = (1/2)(x_1^2 + \cdots + x_k^2)^{-1/2} \cdot 2x_i,$$

so the gradient (which is the only element in the subdifferential) is $\boldsymbol{x}/\|\boldsymbol{x}\|$. Furthermore, we can write the norm function as

$$\|\boldsymbol{x}\| = \max_{\|\boldsymbol{a}\|=1} \boldsymbol{a}^\top \boldsymbol{x},$$

due to the Cauchy-Schwartz inequality. The gradient of $\boldsymbol{a}^\top\boldsymbol{x}$ is $\boldsymbol{a}$, and according to subgradient calculus, the subdifferential is the convex hull of all such $\boldsymbol{a}$'s that gives the maximum value. When $\boldsymbol{x} \neq 0$, the maximum is only attained when $\boldsymbol{a} = \boldsymbol{x}/\|\boldsymbol{x}\|$, which confirms our previous result. When $\boldsymbol{x} = 0$, the max is attained for any $\boldsymbol{a}$ with $\|\boldsymbol{a}\| = 1$, therefore the subdifferential is their convex hull $\{\boldsymbol{g} \mid \|\boldsymbol{g}\| \leq 1\}$. To summarize, and using $\boldsymbol{\Theta}_j$ as the variable, we have

$$\partial\|\boldsymbol{\Theta}_j\| = \begin{cases} \{\boldsymbol{\Theta}_j/\|\boldsymbol{\Theta}_j\|\}, & \boldsymbol{\Theta}_j \neq 0, \\ \{\boldsymbol{g} \mid \|\boldsymbol{g}\| \leq 1\}, & \boldsymbol{\Theta}_j = 0. \end{cases}$$

(b) The gradient for the proximal term $(1/2)\|\boldsymbol{\Theta}_j - \tilde{\boldsymbol{\Theta}}_j\|^2$ is simply $\boldsymbol{\Theta}_j - \tilde{\boldsymbol{\Theta}}_j$. If $\boldsymbol{\Theta}_j$ is optimal for (6), then

$$0 \in \partial\rho\|\boldsymbol{\Theta}_j\| + \boldsymbol{\Theta}_j - \tilde{\boldsymbol{\Theta}}_j,$$

or equivalently

$$\frac{1}{\rho}(\tilde{\boldsymbol{\Theta}}_j - \boldsymbol{\Theta}_j) \in \partial\|\boldsymbol{\Theta}_j\|.$$

According to our derivation in part (a), there are two cases:

i. If $\boldsymbol{\Theta}_j = 0$, then it implies $\|(1/\rho)\tilde{\boldsymbol{\Theta}}_j\| \leq 1$, i.e., $\|\tilde{\boldsymbol{\Theta}}_j\| \leq \rho$;

ii. If $\boldsymbol{\Theta}_j \neq 0$, then we must have $\boldsymbol{\Theta}_j = \alpha(\tilde{\boldsymbol{\Theta}}_j - \boldsymbol{\Theta}_j)$ for some scalar $\alpha > 0$ and $\|(1/\rho)(\tilde{\boldsymbol{\Theta}}_j - \boldsymbol{\Theta}_j)\| = 1$. The first condition simplifies to $\boldsymbol{\Theta}_j = \beta\tilde{\boldsymbol{\Theta}}_j$ for some $\beta > 0$, and plugging it into the second condition, we get

$$\left\| \frac{1-\beta}{\rho}\tilde{\boldsymbol{\Theta}}_j \right\| = 1,$$

so $\beta = 1 - \rho/\|\tilde{\boldsymbol{\Theta}}_j\|$. Notice that $\beta > 0$ only if $\|\boldsymbol{\Theta}_j\| > \rho$, which complements all the other cases.

To summarize,

$$\mathrm{Prox}_{\rho\|\cdot\|}(\tilde{\boldsymbol{\Theta}}_j) = \begin{cases} 0, & \|\tilde{\boldsymbol{\Theta}}_j\| \leq \rho, \\ \left(1 - \rho/\|\tilde{\boldsymbol{\Theta}}_j\|\right)\tilde{\boldsymbol{\Theta}}_j, & \|\tilde{\boldsymbol{\Theta}}_j\| > \rho. \end{cases}$$

This can be simplified into one single expression

$$\mathrm{Prox}_{\rho\|\cdot\|}(\tilde{\boldsymbol{\Theta}}_j) = \left(1 - \rho/\|\tilde{\boldsymbol{\Theta}}_j\|\right)_+ \tilde{\boldsymbol{\Theta}}_j.$$

The proximal operator for the group-lasso regularization is this operation applied to each row of $\tilde{\boldsymbol{\Theta}}$.

4. (30%) *Hand-written digits classification.* The MNIST data set is a famous data set for multi-class classification, which can be downloaded here `http://yann.lecun.com/exdb/mnist/`. In this problem you will design a SGD algorithm for multi-class support vector machine with group-sparse regularization that solves the following optimization problem

$$\underset{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_k}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}\max_{c}(\boldsymbol{x}_i^\top\boldsymbol{\theta}_c - \boldsymbol{x}_i^\top\boldsymbol{\theta}_{y_i} + 1_{y_i\neq c}) + \lambda\sum_{j=1}^{m}\sqrt{\sum_{c=1}^{k}\theta_{jc}^2}.$$

Here we simply assume that the features are the image pixels themselves (we even ignore the constant 1 here).

(a) Derive the stochastic proximal subgradient algorithm for solving it. For simplicity, you can assume that there is only one term that reaches the maximum value in $\max_c(\boldsymbol{\phi}^\top\boldsymbol{\theta}_c - \boldsymbol{\phi}_i^\top\boldsymbol{\theta}_{y_i} + 1_{y_i\neq c})$ throughout the iterations. At iteration t, you can simply denote the step size as $\gamma^{(t)}$.

(b) Implement the algorithm in your favorite programming language.

(c) Run the algorithm with $\lambda = 10, 1, 0.1, 0.01$ and diminishing step size $\gamma^{(t)} = 1/t$, and run the algorithm for 10^6 iterations. At every 1000 iteration, evaluate the prediction accuracy on the test set and plot the progress on a figure.

(d) For the solution of each λ value, show a black and white figure for the pixels that are being used to make the predictions. Is it true that a large λ leads to a more sparse solution?

Solution.

(a) Denote $\boldsymbol{\Theta} = [\ \boldsymbol{\theta}_1\ \boldsymbol{\theta}_2\ \cdots\ \boldsymbol{\theta}_k\ ]$ and

$$\ell_i(\boldsymbol{\Theta}) = \max_{c}(\boldsymbol{x}_i^\top\boldsymbol{\theta}_c - \boldsymbol{x}_i^\top\boldsymbol{\theta}_{y_i} + 1_{y_i\neq c}).$$

Suppose at a particular point $\boldsymbol{\Theta}^{(t)}$ only one of them attains the maximum, then the function is differentiable. If maximum is attained at $y_i = \arg\max_c$, then

$$\nabla\ell_i(\boldsymbol{\Theta}^{(t)}) = 0;$$

otherwise, denote $\widehat{y_i} = \arg\max_c$, and

$$\nabla_{\boldsymbol{\theta}_c}\ell_i(\boldsymbol{\Theta}^{(t)}) = \begin{cases} \boldsymbol{x}_i & c = \widehat{y_i} \\ -\boldsymbol{x}_i & c = y_i \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, the gradient descent step updates at most two columns of the $\boldsymbol{\Theta}$ matrix: the column that corresponds to the correct label is added with a little bit of $\boldsymbol{x}_i$, and the column corresponding to the predicted label is subtracted with a little bit of $\boldsymbol{x}_i$; all the other columns stay the same.

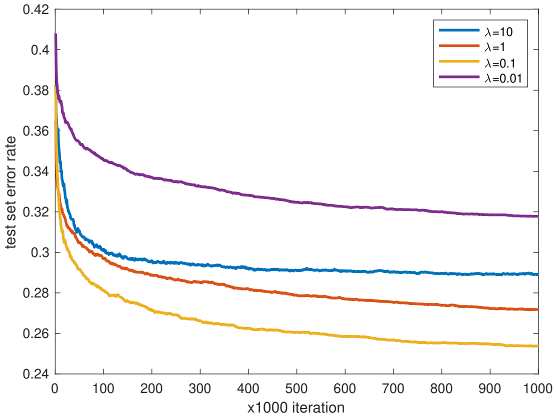
Then we apply row-wise block soft-thresholding on $\boldsymbol{\Theta}$. The norm of each row of $\boldsymbol{\Theta}$ is calculated; if it is less than $\gamma^{(t)}\lambda$, the entire row is set to zero, otherwise the magnitude of the row decreases by the amount of $\gamma^{(t)}\lambda$.

The stochastic proximal subgradient algorithm for L_1-norm regularized multiclass SVM is elaborated here. The cth column of $\boldsymbol{\Theta}$ is denoted $\boldsymbol{\theta}_c$, and the jth row is denoted $\boldsymbol{\Theta}_j$.

```
initialize $\boldsymbol{\Theta}^{(0)}$
for $t = 0, 1, \ldots$ do
    uniformly sample $i$ from $\{1, \ldots, n\}$
    calculate $\widehat{y_i} = \arg\max_c (\boldsymbol{x}_i^\top \boldsymbol{\theta}_c^{(t)} - \boldsymbol{x}_i^\top \boldsymbol{\theta}_{y_i}^{(t)} + 1_{y_i \neq c})$
    $\boldsymbol{\theta}_{y_i}^{(t+1)} \leftarrow \boldsymbol{\theta}_{y_i}^{(t)} + \gamma^{(t)} \boldsymbol{x}_i$
    $\boldsymbol{\theta}_{\widehat{y_i}}^{(t+1)} \leftarrow \boldsymbol{\theta}_{\widehat{y_i}}^{(t)} - \gamma^{(t)} \boldsymbol{x}_i$
    for $j = 1$ to $m$ do
        $\nu \leftarrow \|\boldsymbol{\Theta}_j^{(t+1)}\|$
        if $\nu \leq \gamma^{(t)} \lambda$ then
            $\boldsymbol{\Theta}_j^{(t+1)} \leftarrow 0$
        else
            $\boldsymbol{\Theta}_j^{(t+1)} \leftarrow \left(1 - \frac{\gamma^{(t)}\lambda}{\nu}\right) \boldsymbol{\Theta}_j^{(t+1)}$
        end if
    end for
end for
```

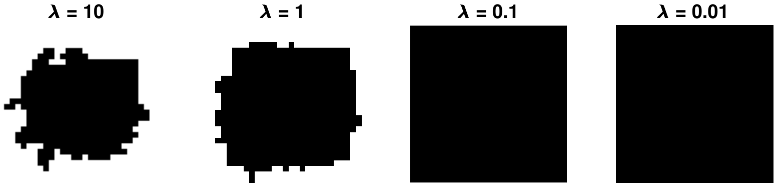
(b) A simple MATLAB code is given.

(c) Here is the convergence plot for four different λ values using the same initialization.



The testing error rate goes down as the algorithm progresses, which is a good sign that the method works. Moreover, we do see the pattern that as we reduce the value of λ, the testing error rate decreases at first and then goes back up.

(d) For each λ we obtain a $\boldsymbol{\Theta}$ solution, and each row of $\boldsymbol{\Theta}$ corresponds to one pixel. In the following figure, each pixel is either black or white, indicating whether the corresponding row in $\boldsymbol{\Theta}$ is zero (white) or not (black). Indeed, we see that a larger λ leads to fewer pixels being used for making the prediction; in fact for $\lambda = 0.1$ and $\lambda = 0.01$ all pixels are being used. When we do select pixels to be ignored for making the predictions, they all lie outside the center circle, which do make sense since we seldom write outside that region.


λ = 10
λ = 1
λ = 0.1
λ = 0.01