EECS 545 – Machine Learning - Homework #4

Due: 11:59pm 12/1/2017

Homework Policy: Working in groups is allowed, but each member must submit their own writeup. Please write the members of your group on your solutions (maximum allowed group size is 4). Homeworks will be submitted via Gradescope (https://gradescope.com/) as pdf files (**including your code**).

1) **Robust Linear Regression (20 pts).**

Least square is not robust to outliers because errors get squared. In this problem, you will compare least-squares and robust regression. The robust regression problem is

$$\min_{\mathbf{w},b} \frac{1}{n}\sum_{i=1}^{n} \rho(y_i - \mathbf{w}^T\mathbf{x}_i - b),$$

where ρ is a *robust loss function.* Unlike least squares, there is no closed form solution for robust linear regression. We will employ the *majorize-minimize* (MM) algorithm to minimize the robust regression objective function $J(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \rho(y_i - \mathbf{w}^T\mathbf{x}_i - b)$. The idea of MM is:

- Initialize $\mathbf{w}_0, b_0$
- $t \leftarrow 0$
- Repeat
  - (*majorize*) Find a function $J_t(\mathbf{w}, b)$ such that

$$J(\mathbf{w}_t, b_t) = J_t(\mathbf{w}_t, b_t)$$
$$J(\mathbf{w}, b) \leq J_t(\mathbf{w}, b), \forall \mathbf{w}, b$$

  - *(minimize)* $(\mathbf{w}_{t+1}, b_{t+1}) \leftarrow \arg\min_{\mathbf{w},b} J_t(\mathbf{w}, b)$
  - $t \leftarrow t + 1$

  Until convergence

The key idea is to choose a surrogate function J_t, whose graph is above J and it approximates J locally. The surrogate function J_t should have some good properties so that it can be minimized efficiently. We will choose J_t that have the form

$$J_t(\mathbf{w}, b) = \sum_{i=1}^{n} \rho_t(y_i - \mathbf{w}^T\mathbf{x}_i - b)$$

where ρ_t is a majorizing function of ρ, which means that $\rho(r) \leq \rho_t(r)\, \forall r$.

We introduce the notations

$$\psi(r) \triangleq \rho'(r)$$
$$\varphi(r) \triangleq \frac{\psi(r)}{r}, r \neq 0$$
$$r_{t,i} = y_i - \mathbf{w}_t^T\mathbf{x}_i - b_t$$

then the following result provides a majorizing function for a broad class of ρ.

Lemma 1. *Assume $\rho(r)$ is symmetric and differentiable, nondecreasing for $r > 0$. $\varphi(r)$ is nonincreasing for $r > 0$, $\varphi(0) \triangleq \lim_{r\to 0} \varphi(r)$ exists and $\varphi(r)$ is continuous. Define*

$$J_t(\mathbf{w}, b) = \sum_{i=1}^{n} \rho_t(y_i - \mathbf{w}^T\mathbf{x}_i - b)$$

where

$$\rho_t(r) = \rho(r_{t,i}) - \frac{1}{2} r_{t,i}\psi(r_{t,i}) + \frac{1}{2}\frac{\psi(r_{t,i})}{r_{t,i}} r^2$$

Then J_t majorizes J

With the majorizing function, the iterative update has the form

$$(\mathbf{w}_{t+1}, b_{t+1}) = \arg\min_{\mathbf{w},b} \sum_{i=1}^{n} c_{t,i}(y_i - \mathbf{w}^T\mathbf{x}_i - b)^2$$

where

$$c_{t,i} = \frac{\psi(r_{t,i})}{r_{t,i}} = \varphi(r_{t,i})$$

Recall that the above minimization problem is exactly the same as the weighted least squares problem you solved in the previous homework, except that here the weights are updated every iteration.

You will experiment this idea on a synthetic dataset. In the dataset, most of the data come from an upward sloping line plus noise, but a fraction (thought of as outliers) come from a downward sloping line. The data is provided in `synthetic.csv`. The first column in the dataset represents x and the second column represents y. The true line (the upward sloping line) is

$$y = 10x + 5$$

Please recall that you are not allowed to use built-in functions that solve regression problems directly. It will be helpful to write a helper function to solve weighted least squares regression.

(a) For general $\rho(r)$ that satisfies the conditions in lemma 1, argue that the majorize-minimize algorithm can be understood as "iteratively reweighted least squares." Explain how the algorithm achieves robustness by considering how weights are assigned to outliers vs. inliers in comparison with the least squares loss.

(b) Consider the robust loss

$$\rho(r) = \sqrt{1 + r^2} - 1$$

Implement the MM algorithm, and report the parameters of the linear function you estimated. For comparison, also report the parameters of ordinary least squares on this data.

(c) Generate a single plot that shows the data, the true line, the ordinary least squares estimate, and the robust estimate. Create a legend and use different line styles.

2) **Constructing Kernels (20 pts).**

(a) Let $\mathbf{u}, \mathbf{v}$ be vectors of dimension d. What feature map ϕ does the kernel

$$k(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle + 1)^4$$

correspond to? In other words, specify the function $\phi(\cdot)$ so that $k(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u})^\top \phi(\mathbf{v})$ for all $\mathbf{u}, \mathbf{v}$. Please show the expression for $d = 3$ and describe how to extend it to arbitrary dimension d.

(b) Let k_1, k_2 be positive-definite kernel functions over $\mathbb{R}^D \times \mathbb{R}^D$, let $a \in \mathbb{R}^+$ be a positive real number, let $f : \mathbb{R}^D \to \mathbb{R}$ be a real-valued function and let $p : \mathbb{R} \to \mathbb{R}$ be a polynomial with *positive* coefficients. For each of the functions k below, state whether it is necessarily a positive-definite kernel. If you think it is, prove it; if you think it is not, give a counterexample.

(i) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$

(ii) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) - k_2(\mathbf{x}, \mathbf{z})$

(iii) $k(\mathbf{x}, \mathbf{z}) = ak_1(\mathbf{x}, \mathbf{z})$

(iv) $k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z})k_2(\mathbf{x}, \mathbf{z})$

(v) $k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$

(vi) $k(\mathbf{x}, \mathbf{z}) = p(k_1(\mathbf{x}, \mathbf{z}))$

(vii) Prove that the Gaussian Kernel $k(\mathbf{x}, \mathbf{z}) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}\right)$ can be expressed as $\phi(\mathbf{x})^T\phi(\mathbf{z})$, where $\phi(\cdot)$ is an infinite-dimensional vector. (Hint: using power series)

3) **Kernel Ridge Regression (20 pts).**

Recall that the error function for ridge regression (linear regression with L2 regularization) is:

$$E(\mathbf{w}) = (\Phi\mathbf{w} - \mathbf{t})^T(\Phi\mathbf{w} - \mathbf{t}) + \lambda\mathbf{w}^T\mathbf{w}$$

and its closed-form solution and model are:

$$\hat{\mathbf{w}} = (\Phi^T\Phi + \lambda I)^{-1}\Phi^T\mathbf{t} \text{ and } \hat{f}(\mathbf{x}) = \hat{\mathbf{w}}^T\phi(\mathbf{x}) = \mathbf{t}^T\Phi(\Phi^T\Phi + \lambda I)^{-1}\phi(\mathbf{x})$$

Now we want to kernelize ridge regression and allow non-linear models.

(a) Use the following matrix inverse lemma to derive the closed-form solution and model for kernelized ridge regression:

$$(P + QRS)^{-1} = P^{-1} - P^{-1}Q(R^{-1} + SP^{-1}Q)^{-1}SP^{-1}$$

where P is an $n \times n$ invertible matrix, R is a $k \times k$ invertible matrix, Q is an $n \times k$ matrix and S is a $k \times n$ matrix. Make sure that your kernelized model only depends on the feature vectors $\phi(\mathbf{x})$ through inner products with other feature vectors.

(b) Apply kernelized ridge regression to the automobile mpg dataset. The training data and test data are provided in `auto_mpg_train.csv` and `auto_mpg_test.csv`, respectively. The first column is the mpg data while the other 7 columns are the features:

```
1. mpg:  continuous
```

```
2. cylinders:  multi-valued discrete
3. displacement:  continuous
4. horsepower:  continuous
5. weight:  continuous
6. acceleration:  continuous
7. model year:  multi-valued discrete
8. origin:  multi-valued discrete
```

We have normalized the feature data to the range [0,1]. Please apply the kernelized ridge regression to this dataset (use mpg as the target, the other 7 columns as features). Report the RMSE (Root Mean Square Error) of the models on the test data. Try (set $\lambda = 1$).

(i) Polynomial kernel $k(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle + 1)^2$

(ii) Polynomial kernel $k(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle + 1)^4$

(iii) Polynomial kernel $k(\mathbf{u}, \mathbf{v}) = (\langle \mathbf{u}, \mathbf{v} \rangle + 1)^8$

(iv) Gaussian kernel $k(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma^2}\right)$ (set $\sigma = 1$)

4) **Subgradient Methods for Soft-Margin SVM and Kernelized SVM (30 pts).**

In this problem, you will implement the subgradient and stochastic subgradient methods for solving the soft-margin SVM problem. As discussed in HW3, the soft-margin SVM problem is

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max\left(0, 1 - t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)\right) \tag{1}$$

(a) The error function in the minimization problem for solving soft-margin SVM is

$$E(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max\left(0, 1 - t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)\right)$$

Find its derivatives: $\nabla_{\mathbf{w}} E(\mathbf{w}, b)$ and $\frac{\partial}{\partial b} E(\mathbf{w}, b)$. Where the derivative is undefined, use a subderivative. For the hinge loss $\ell = \max(0, r)$, a valid subderivative of $d\ell/dr$ at the point $r = 0$ can be any value in the range $[\lim_{r\to 0^-} \frac{d\ell}{dr}, \lim_{r\to 0^+} \frac{d\ell}{dr}] = [0, 1]$.

(b) Implement the soft-margin SVM using batch gradient descent. Here is the pseudo code:

Algorithm 1: SVM Batch Gradient Descent

```
$\mathbf{w}^* \leftarrow \mathbf{0}$ ;
$b^* \leftarrow 0$ ;
for j=1 to NumIterations do
    $\mathbf{w}_{grad} \leftarrow \nabla_{\mathbf{w}} E(\mathbf{w}^*, b^*)$ ;
    $b_{grad} \leftarrow \frac{\partial}{\partial b} E(\mathbf{w}^*, b^*)$ ;
    $\mathbf{w}^* \leftarrow \mathbf{w}^* - \alpha(j)\ \mathbf{w}_{grad}$ ;
    $b^* \leftarrow b^* - \alpha(j)\ b_{grad}$ ;
end
return $\mathbf{w}^*$
```

The learning rate for the j-th iteration is defined as:

$$\alpha(j) = \frac{\eta_0}{1 + j \cdot \eta_0}$$

Set η_0 to 0.001 and the slack cost C to 3. Show the iteration-versus-accuracy (training accuracy) plot. The training and test data/labels are provided in `digits_training_data.csv`, `digits_training_labels.csv`, `digits_test_data.csv` and `digits_test_labels.csv`. This dataset is a subset of MNIST dataset and it only contains digits 7 and 9. Same as in previous homework assignments, each row in the training data and test data files is a vectorized digit image.

(c) Let

$$E^{(i)}(\mathbf{w}, b) = \frac{1}{2N}\|\mathbf{w}\|^2 + C \max\left(0, 1 - t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)\right)$$

then

$$E(\mathbf{w}, b) = \sum_{i=1}^{N} E^{(i)}(\mathbf{w}, b)$$

Find the derivatives $\nabla_{\mathbf{w}} E^{(i)}(\mathbf{w}, b)$ and $\frac{\partial}{\partial b} E^{(i)}(\mathbf{w}, b)$. Once again, use a subderivative if the derivative is undefined.

(d) Implement the soft-margin SVM using stochastic gradient descent. Here is the pseudo-code:

Algorithm 2: SVM Stochastic Gradient Descent

```
$\mathbf{w}^* \leftarrow \mathbf{0}$ ;
$b^* \leftarrow 0$ ;
for j=1 to NumIterations do
    for i = Random Permutation of 1 to N do
        $\mathbf{w}_{grad} \leftarrow \nabla_{\mathbf{w}} E^{(i)}(\mathbf{w}^*, b^*)$ ;
        $b_{grad} \leftarrow \frac{\partial}{\partial b} E^{(i)}(\mathbf{w}^*, b^*)$ ;
        $\mathbf{w}^* \leftarrow \mathbf{w}^* - \alpha(j)\ \mathbf{w}_{grad}$ ;
        $b^* \leftarrow b^* - \alpha(j)\ b_{grad}$ ;
    end
end
return $\mathbf{w}^*$
```

Use the same $\alpha(\cdot)$, η_0 and C in (b). Be sure to use a new random permutation of the indices of the inner loop for each iteration of the outer loop. Show the iteration-versus-accuracy (outer iteration and training accuracy) curve **in the same plot** as that for batch gradient descent. The training and test data/labels are provided in `digits_training_data.csv`, `digits_training_labels.csv`, `digits_test_data.csv` and `digits_test_labels.csv`.

(e) What can you conclude about the convergence rate of stochastic gradient descent versus batch gradient descent? How did you make this conclusion?

(f) Show the Lagrangian function for minimization problem (1) and derive the dual problem. Your result should have only dual variables in the objective function as well as the constraints. How can you kernelize the soft-margin SVM based on this dual problem?

(g) Apply the soft-margin SVM (with RBF kernel) to handwritten digit classification. The training and test data/labels are provided in `digits_training_data.csv`, `digits_training_labels.csv`, `digits_test_data.csv` and `digits_test_labels.csv`. Report the training and test accuracy, and show 5 of the misclassified test images (if fewer than 5, show all; label them with your predictions). You can use `svm.SVC()` of scikit-learn (or `fitcsvm()` of Matlab) implementation of the kernelized SVM in this question. You are free to select the parameters (for RBF kernel and regularization), and please report your parameters.

5) **Principal Components Analysis (20 pts).**

In Principal Components Analysis (PCA), we project the data $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N | \mathbf{x}_i \in \mathbb{R}^D\}$ into K ($K < D$) orthogonal directions, which maximizes the following projection variance:

$$\max_{\substack{A \\ A^T A = I_k}} \sum_{k=1}^{K} \mathbf{a}_k^T S \mathbf{a}_k \tag{2}$$

where $S = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \in \mathbb{R}^{D\times D}$ is the data covariance matrix, transformation matrix $A = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_K \end{bmatrix} \in \mathbb{R}^{D\times K}$ and $\mathbf{a}_k^T \mathbf{x}_i$ is the projection of the i-th data point into the k-th direction. Suppose S has the eigenvalue decomposition $S = U\Lambda U^T$ where $U = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_D \end{bmatrix} \in \mathbb{R}^{D\times D}$ and $U^T U = I_D$; diagonal matrix $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_D)$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$. Denote $\mathbf{w}_k = U^T \mathbf{a}_k$ and $W = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \cdots & \mathbf{w}_K \end{bmatrix} \in \mathbb{R}^{D\times K}$, then we get the following optimization problem from (1):

$$\max_{\substack{W \\ W^T W = I_k}} \sum_{k=1}^{K} \mathbf{w}_k^T \Lambda \mathbf{w}_k \tag{3}$$

(a) We denote

$$h_j = \sum_{k=1}^{K} \|\mathbf{w}_k^{(j)}\|^2$$

i.e., the square of L_2 norm of the j-th row vector in W. Prove that $0 \leq h_j \leq 1$ and $\sum_{j=1}^{D} h_j = K$.

(b) Prove that

$$\max_{\substack{W \\ W^T W = I_k}} \sum_{k=1}^{K} \mathbf{w}_k^T \Lambda \mathbf{w}_k = \max_{\substack{h_j \\ W^T W = I_k}} \sum_{j=1}^{D} h_j \lambda_j \tag{4}$$

(c) What are the optimal h_j in (4)? Show that $\mathbf{a}_k = \mathbf{u}_k$ ($k = 1, \cdots, K$) is a solution of (4).

(d) Is the solution of (4) unique? Give a necessary and sufficient condition for the subspace spanned by the solution $\{\mathbf{a}_1^*, \mathbf{a}_2^*, \cdots, \mathbf{a}_K^*\}$ to be unique.

(e) We can construct the solution $\mathbf{a}_k = \mathbf{u}_k$ $(k = 1, \cdots, K)$ in an iterative way. We notice that $\mathbf{a}_1 = \mathbf{u}_1$ is a solution for

$$\max_{\substack{\mathbf{a}_1 \\ \mathbf{a}_1^T \mathbf{a}_1 = 1}} \mathbf{a}_1^T S \mathbf{a}_1$$

which is a special case (K=1) in (4).
Show that if $\mathbf{a}_k$ $(k = 2, \cdots, K)$ is orthogonal to $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_{k-1}$ (i.e. $\mathbf{a}_k^T \mathbf{u}_i = 0$ where $i = 1, ..., k-1$), then $\mathbf{a}_k = \mathbf{u}_k$ is a solution of

$$\max_{\substack{\mathbf{a}_k \\ \mathbf{a}_k^T \mathbf{a}_k = 1 \\ \mathbf{a}^T \mathbf{u}_i = 0, i=1,...,k-1}} \mathbf{a}_k^T S \mathbf{a}_k$$