CS 536: Homework 3

Solutions

Instructions: Same as homework 1.

1 Bayesian Regression (10 pts)

Recall Bayesian estimation from homework 1. Here we estimate a parameter θ by maximizing the weighted log likelihood

$$\theta_{new} = argmax_{\hat{\theta}} P(\hat{\theta}) P(X_1, X_2, \ldots X_n | \hat{\theta})$$

The weight function $P(\hat{\theta})$ is commonly known as the *prior*.

1. Show that the Ridge regression estimate is equivalent to doing Bayesian estimation as above by assuming a Gaussian prior on each weight value, i.e. assuming $w_i \sim N(0, \frac{\sigma^2}{\lambda})$.

   Solution: The model is $y = w \cdot X + N(0, \sigma^2)$. Assuming $P(w_i) = \frac{\sqrt{\lambda}}{\sqrt{2\pi}\sigma} e^{-\frac{\lambda w_i^2}{2\sigma^2}}$ we get that the weighted likelihood is

   $$\prod_{i=1}^{n} P(w_i) \prod_{i=1}^{m} P(X_i, y_i | w_1, w_2, \ldots w_n) = \prod_{i=1}^{n} \frac{\sqrt{\lambda}}{\sqrt{2\pi}\sigma} e^{-\frac{\lambda w_i^2}{2\sigma^2}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(y_i - w \cdot X_i)^2}{2\sigma^2}}$$

   Taking log, the part that depends on the parameter is $-\sum_{i=1}^{m}((y_i - w \cdot X_i)^2) - \lambda \sum_{i=1}^{n} w_i^2$. Maximizing this is the same as computing the Ridge estimate.
2. Show that the Lasso estimate is equivalent to doing Bayesian estimation as above by assuming a Laplace prior[1] on each weight value, i.e. assuming $w_i \sim L(0, \frac{\sigma^2}{\lambda})$.

   Solution: The model is $y = w \cdot X + N(0, \sigma^2)$. Assuming $P(w_i) = \frac{\lambda}{2\sigma^2} e^{-\frac{\lambda |w_i|}{\sigma^2}}$ we get that the weighted likelihood is

   $$\prod_{i=1}^{n} P(w_i) \prod_{i=1}^{m} P(X_i, y_i | w_1, w_2, \ldots w_n) = \prod_{i=1}^{n} \frac{\lambda}{2\sigma^2} e^{-\frac{\lambda |w_i|}{\sigma^2}} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(y_i - w \cdot X_i)^2}{2\sigma^2}}$$

   Taking log, the part that depends on the parameter is $-\sum_{i=1}^{m}((y_i - w \cdot X_i)^2) - \lambda \sum_{i=1}^{n} |w_i|$. Maximizing this is the same as computing the Lasso estimate.

Here λ is the penalty used in Ridge/Lasso regression and the regression model is $y = \sum_{i=1}^{n} w_i x_i + \xi$, where $\xi \sim N(0, \sigma^2)$ is Gaussian noise.

2 Ridge vs Lasso (10 pts)

Consider regression in 2 dimensions, i.e. $y = w_1 x_1 + w_2 x_2 + \xi$, where $\xi \sim N(0, \sigma^2)$ is Gaussian noise.

1. Derive closed form expressions for the Ridge and Lasso estimates for the above 2-dimensional problem.

[1] https://en.wikipedia.org/wiki/Laplace_distribution

2. How do the estimates change as the co-variance between x_1 and x_2 changes.

   Solution: Let's first consider ridge regression. We know that the ridge estimate is $\hat{w} = (X^TX + \lambda I)^{-1}X^TY$. If we have m samples then X is the $m \times 2$ data matrix and Y is the m dimensional column vector of output values. Let λ_1 and λ_2 be the principal eigenvalues of the covariance matrix. Then in class we saw that the coefficient along the coordinate corresponding to λ_1 will be shrunk by $\frac{\lambda_1^2}{\lambda_1^2+\lambda}$. Similarly for λ_2. If the variables are uncorrelated, both the coordinates will be shrunk equally. As the correlation increases, the gap between λ_1 and λ_2 increases and hence the coordinates will be shrunk unequally. The smaller coordinate will be shrunk more aggressively. Hence, in this case, behavior of ridge is very much dependent on the correlation between x_1 and x_2.

   Now let's consider Lasso. Let S_1 be the correlation of x_1 with y, i.e. $S_1 = \sum_{i=1}^m y_i X_{i,1}$ and $S_2 = \sum_{i=1}^m y_i X_{i,2}$. Without loss of generality, assume that $|S_1| \geq |S_2|$. There will be three cases:

   Case 1: $\lambda \geq 2|S_1|$. In this case both w_1 and w_2 will be zero. To see this, notice that the cost of the Lasso objective at $w_1 = w_2 = 0$ is equal to $\sum_{i=1}^m y_i^2$. The cost at a point away from 0, say $(w_1 = \delta_1, w_2 = \delta_2)$ is $\sum_{i=1}^m (y_i - \delta_1 X_{i,1} - \delta_2 X_{i,2})^2 + \lambda|\delta_1| + \lambda|\delta_2|$. The difference between this cost and the cost at 0 is $= \sum_{i=1}^m -(\delta_1 X_{i,1} + \delta_2 X_{i,2})(2y_i - \delta_1 x_{i,1} - \delta_2 X_{i,2}) + \lambda|\delta_1| + \lambda|\delta_2|$. This can be written as $-2\delta_1 \sum_{i=1}^m y_i X_{i,1} - 2\delta_2 \sum_{i=1}^m y_i X_{i,2} + \delta_1 \sum_{i=1}^m X_{i,1}^2 + \delta_2 \sum_{i=1}^m X_{i,2}^2 + \lambda|\delta_1| + \lambda|\delta_2|$. This difference is always greater than $-2\delta_1 S_1 - 2\delta_2 S_2 + \lambda|\delta_1| + \lambda|\delta_2|$. For a high value of λ, this difference will always be positive and hence the weights prefer to be at 0.

   Case 2: $2|S_1| > \lambda > S_{crit}$. Once λ becomes less than $2|S_1|$, w_1 will have incentive to deviate from 0 (Again, look at the above calculation). For a while, w_1 will be non-zero and w_2 will be 0, until λ reaches a critical value λ_{crit}. In order to compute the value of S_{crit}, fix w_1 and do the above cost analysis with w_2 moving from 0 to δ. The value turns out to be $S_{crit} = \sum_i 2(y_i - w_1 x_{i,1}) x_{i,2}$. In other words, this is the correlation of x_2 with the residual vector, after subtracting $w_1 x_{i,1}$.

   Case 3: $\lambda \leq S_{crit}$. In this case both w_1 and w_2 will have incentive to move away from zero.

   In all the three cases, we can compute the closed form optimal value (by setting gradients to 0) since we know exactly which variables will be non-zero. The critical value S_{crit} will shift towards zero as the correlation between x_1 and x_2 increases.

3 VC Dimension Bounds (10 pts)

Let C be a finite class of functions from $\mathcal{R}^d \mapsto \{0, 1\}$. Show that the VC dimension of C is at most $\log |C|$, where $|C|$ = the number of functions in C.

Solution: Suppose that VC dimension of C is more than $\log |C|$. This implies that there exists $m > \log |C|$ points that can be shattered by functions in C. There are more than $|C|$ possible labelings of this set of m points. However, there are only $|C|$ functions in C and hence C cannot shatter this set. Thus, we reach a contradiction.

4 VC Dimension of Majority (10 pts)

Show that if hypothesis class H has VC-dimension d, then the class $MAJ_k(H)$ has VC-dimension $O(kd \log kd)$. Here, $MAJ_k(H)$ is the class of functions achievable by taking majority votes over k functions in H. Note that we are only asking for an upper bound here, not a lower bound.

Solution: Suppose that the VC dimension of $MAJ_k(H)$ is m. Then there exists a set of m point that can be shattered in all possible ways. We also know that the number of ways to shatter a set of m points using functions in H is $C[m] \leq m^d$. This means that the number ways to shatter m points using k functions in H is at most $\binom{C[m]}{k} \leq m^{dk}$. Hence, we have that $2^m \leq m^{dk}$. Solving for m we get that $m = O(kd \log kd)$.

5 Ridge in RKHS (10 pts)

Given a regression problem $y = \sum_{i=1}^{n} w_i x_i + \xi$, where $\xi \sim N(0, \sigma^2)$, show how one would run ridge regression efficiently in a different feature space $\phi : \mathcal{R}^n \mapsto H$. Here H is a reproducing kernel Hilbert space with a kernel function $K : \mathcal{R}^n \times \mathcal{R}^n \mapsto \mathcal{R}$. Your algorithm should run in time polynomial in n and the size of training data and the time required to compute the kernel function on a pair of points. The run time should have no dependence on the dimensionality of the Hilbert space.

Hint: You might find the following equality from linear algebra useful:

$$(P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = P B^T (B P B^T + R)^{-1}.$$

Solution: Let $\boldsymbol{X}$ be the $m \times d$ matrix where each row corresponds to a data point and let $\boldsymbol{Y}$ be the column vector of the output values, i.e. y_i's. Then we know that the ridge estimate can be written as $w = (\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I}_d)^{-1} \boldsymbol{X}^T \boldsymbol{Y}$. Using the linear algebraic inequality mentioned above, this can be re-written as $w = \boldsymbol{X}^T (\boldsymbol{X}\boldsymbol{X}^T + \lambda \boldsymbol{I}_m)^{-1} \boldsymbol{Y}$. If we are in the kernel space, this would become $w = \boldsymbol{\phi(X)}^T (\boldsymbol{\phi(X)}\boldsymbol{\phi(X)}^T + \lambda \boldsymbol{I}_m)^{-1} \boldsymbol{Y} = \boldsymbol{\phi(X)}^T (\boldsymbol{K} + \lambda \boldsymbol{I}_m)^{-1} \boldsymbol{Y} = \sum_{i=1}^{m} \alpha_i \boldsymbol{\phi(X_i)}$. Here $\alpha_i = ((\boldsymbol{K} + \lambda \boldsymbol{I}_m)^{-1} Y)_i$ and $\boldsymbol{K}$ is the kernel matrix. The value of a new point X can be computed as $w \cdot X = \sum_i \alpha_i K(X, X_i)$.

6 L_0 Regression (10 pts)

Consider a regression problem $y = \sum_{i=1}^{n} w_i x_i + \xi$, where $\xi \sim N(0, \sigma^2)$. Given $(\vec{X}_1, y_1), (\vec{X}_2, y_2), \ldots (\vec{X}_m, y_m)$, let $\boldsymbol{X}$ be the $m \times n$ data matrix where $\boldsymbol{X}_{i,j} = \vec{X}_{i,j}$ and let $\boldsymbol{Y} = [y_1, y_2, \ldots, y_m]^T$. Define

$$\hat{w} = min_w \frac{1}{2m} \sum_{i=1}^{m} (y_i - \vec{X}_i \cdot w)^2 + \lambda \|w\|_0$$

Show that if $\frac{1}{m} \boldsymbol{X^T X} = \boldsymbol{I}$ then

$$\hat{w}_j = \begin{cases} \frac{1}{m} v_j^T \mathbf{Y} & \text{if} \quad \frac{1}{m} v_j^T \mathbf{Y} > \sqrt{2\lambda} \\ 0 & \text{if} \quad \frac{1}{m} v_j^T \mathbf{Y} \in [-\sqrt{2\lambda}, \sqrt{2\lambda}] \\ \frac{1}{m} v_j^T \mathbf{Y} & \text{if} \quad \frac{1}{m} v_j^T \mathbf{Y} < -\sqrt{2\lambda} \end{cases}$$

for all $j = 1 \ldots n$. Here v_j is the jth column of $\boldsymbol{X}$ and $\|w\|_0 =$ number of non-zeros in w.

Solution: The objective can be written in matrix form as $\frac{1}{2m}(\boldsymbol{Y} - \boldsymbol{X}w)^T (\boldsymbol{Y} - \boldsymbol{X}w) + \lambda \|w\|_0$. Expanding, we get $\frac{1}{2m}[\boldsymbol{Y}^T \boldsymbol{Y} - 2w^T \boldsymbol{X}^T \boldsymbol{Y} + w^T \boldsymbol{X}^T \boldsymbol{X} w] + \lambda \|w\|_0$. We can ignore the first term as it does not depend on w. Hence, our goal is to minimize $\frac{1}{2m} w^T \boldsymbol{X}^T \boldsymbol{X} w - \frac{1}{m} w^T \boldsymbol{X}^T \boldsymbol{Y} + \lambda \|w\|_0$. Using the fact that $\frac{1}{m} \boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{I}$ and letting v_j be the jth column of $\boldsymbol{X}$, we can reduce the problem to minimizing

$$\sum_{j=1}^{n} \frac{w_j^2}{2} - \frac{1}{m} v_j^T \boldsymbol{Y} w_j + \lambda I(w_j \neq 0)$$

Here $I()$ is the indicator function that takes value 1 if $w_j \neq 0$ and 0 otherwise. The above objective is the sum of n independent 1-dimensional problems that can be optimized separately. Let's consider the 1-dimensional problem for a given w_j. When $w_j = 0$, the objective value if 0. When $w_j \neq 0$, we can compute the optimal value by setting the gradient to 0. The optimal value will turn out to be $\frac{1}{m} v_j^T \boldsymbol{Y}$ and the objective value will be $-\frac{1}{2}(\frac{1}{m} v_j^T \boldsymbol{Y})^2 + \lambda$. Hence, whenever this value is less than 0, w_j will be $\frac{1}{m} v_j^T \boldsymbol{Y}$. Otherwise, w_j will be 0.

7 Infinite VC Dimension (10pts)

Define a function $h_w(x) = sgn(sin(wx))$ for $w \in \mathbb{R}$. Let $H = \{h_w(x) : w \in \mathbb{R}\}$. Show that the VC dimension of H is infinite.

Solution: It is enough to show that for any integer $m > 0$, there exists a set of m points that can be shattered. Let the points be $\{x_1 = \frac{1}{2}, x_2 = \frac{1}{2^2}, \ldots, x_m = \frac{1}{2^m}\}$. Then, for any labeling $(y_1, y_2, \ldots, y_m) \in \{-1, +1\}^m$, the hypothesis $sgn(sin(ax))$ will shatter the given labeling by setting $a = \pi(1 + \sum_{i=1}^{m} 2^{i-1}(1 - y_i))$.

8 Matchmaking via Lasso (30pts)

You have just launched a new dating app. In order to gain an edge among your competitors, you would like to get a better understanding of your customer base. Towards that goal you want to group your users into different types, say based on their personality[2]. Based on a user's information and interaction history, you have represented each user as a high dimensional vector. Your goal now is to partition these vectors into k groups. Formally,

You are given a set S of n vectors in $\mathbb{R}^d$. You want to output k sets $S_1, S_2, \ldots S_k$ where $S_i \subseteq S$ and $S_i \cap S_j = \phi, \forall i \neq j$ and $\cup_{i=1}^{k} S_i = S$. In order to make this problem well defined, we need to make a reasonable assumption about the sets S_i[3]. Here is one such assumption

1. **Low Rank Assumption:** For each $i = 1 \ldots k$, let A_i be the $|S_i| \times d$ matrix where each row corresponds to the vector of one of the users in the set S_i. Then, A_i has rank $\frac{d}{k}$.

Show how you would use Lasso to get the correct partitioning under the above assumption. You can assume that all sets are of the same size and $rank(\sum_{i=1}^{k} A_i) = d$. Furthermore, for each set S_i, any subset of $\frac{d}{k}$ points in S_i are linearly independent.

[Hint: The low rank assumption implies that points within a set lie in a low dimensional space.]

Solution: The idea is to solve for each point X_i, the following problem

$$min_w \|X_i - S_{-i}w\|^2 + \lambda\|w\|_1$$

here S_{-i} is the set of all data points excluding X_i. We want to show that under the assumptions above, there is a value of λ such that the optimal w for X_i will only have non-zero values for other points in X_i's cluster. An easier argument to prove is that there exists a value V such that minimizing $\|X_i - S_{-i}w\|^2$ subject to $\|w\|_1 \leq V$ will have the desired effect.

Proof: Let S_i be the cluster of X_i and A_i be the corresponding matrix. The assumption that every set of $\frac{d}{k}$ points in A_i are linearly independent implies that these points are in *general position*, i.e. any set of $\frac{d}{k}$ points in A_i span a $\frac{d}{k}$ dimensional subspace. The rank assumption implies that these k subspaces are independent, i.e. any linear combination of points from $\cup_{j \neq i} A_j$ cannot lie in the subspace of S_i. Now, given a point X_i belonging to S_i, we know that there exists a way to represent X_i using $\frac{d}{k}$ other points in S_i. The w_{sparse} be one such combination that has the minimum value of $\|w\|_1 = V^*$. Then we claim that for $V = V^*$, no other combination having non-zero values on points from other clusters can exist.

Consider another combination w that has $\|w\|_1 \leq V^*$. Let $w = w_{sparse} + h + h_{-i}$. Here h is a vector having non-zero values for points in S_i and h_{-i} is a vector having non zeros values for points in $\cup_{j \neq i} S_j$. We know that $X_i = S_{-i}w_{sparse}$. Hence, we must also have $X_i = S_{-i}w = X_i + S_{-i}(h + h_i)$. This implies that $S_{-i}(h + h_{-i}) = 0$. Hence, we get that there is a linear combination of points not from S_i that gives a non-zero vector in S_i. From independence, this cannot happen and hence $S_{-i}h_{-i} = 0$. This means that $w_{sparse} + h$ is also a feasible solution with L_1 norm less than V^* leading to contradiction on the fact that w_{sparse} achieves the minimum L_1 norm of V^*.

9 Coordinate Gradient Descent (20pts)

Recall that Lasso solves the following loss minimization problem $min_w \frac{1}{m} \sum_{i=1}^{m} (y_i - \vec{X}_i \cdot w)^2 + \lambda\|w\|_1$. Given a vector w, suppose we want to locally improve it by just changing one coordinate,

[2]That way you can suggest better matches. For instance, you can recommend people of similar personality type more often to a person.

[3]These sets represent "similar" people and hence must have special structure.

say coordinate j. To do this we will minimize $\frac{1}{m}\sum_{i=1}^{m}(y_i - \vec{X_i}\cdot\hat{w})^2 + \lambda\|\hat{w}\|_1$ over all vectors $\hat{w}$ such that $\hat{w}_i = w_i, \forall i \neq j$. So this is a one-dimensional problem.

1. Derive a close form expression for the update of $\hat{w}_j$.

Solution: For a given j, define the residual value $S_j = \frac{1}{m}\sum_{i=1}^{m} X_{i,j}(y_i - \hat{y}_i)$, where $\hat{y}_i = \sum_{j'\neq j} w_{j'}X_{i,j'}$. Then the update for w_j will be

$$\hat{w}_j = \begin{cases} S_j - \lambda & \text{if} \quad S_j > \lambda \\ 0 & \text{if} \quad S_j \in [-\lambda, \lambda] \\ S_j + \lambda & \text{if} \quad S_j < -\lambda \end{cases}$$

Using the above observation, a practical algorithm for Lasso is the following:

- Set $w_0 = \vec{0}$.
- for $j = 1 \ldots n$
  - Update the current weight vector to a new one by optimizing over j.

The above for loop is repeated until the weight vector converges or the total loss does not decrease significantly from one step to the next.

2. Implement your solution to Problem 8 using the above algorithm as a subroutine to solve Lasso.
   Solution: When you implement the solution from problem 8, one expects to get a graph with many *good* edges and not many *false* edges, i.e. edges between points belonging to different clusters. Given such a graph, the goal is then to partition it into k clusters based on edge information. A popular algorithm for this is called *spectral clustering*. See this: https://en.wikipedia.org/wiki/Spectral_clustering.
3. Download the CIFAR-10 dataset from here: http://www.cs.toronto.edu/ kriz/cifar.html
4. The dataset has 60,000 images belonging to 10 different classes. 50,000 images are for training and 10,000 for testing. Each image is a vector of length 3072. Use your implementation above to partition the test set into 10 categories. Report your error. Error is defined as $\frac{\text{\# pairs of images that have the same class label but belong to different sets in your solution}}{\text{total \# pairs of images}}$.

Note: Use the training data or a subset of it to tune the value of λ. Submit your code. Your algorithm for Problem 8 is probably designed to work when *all* the sets lie in a low dimensional space. You will have to modify it in a reasonable way to run it on this dataset.