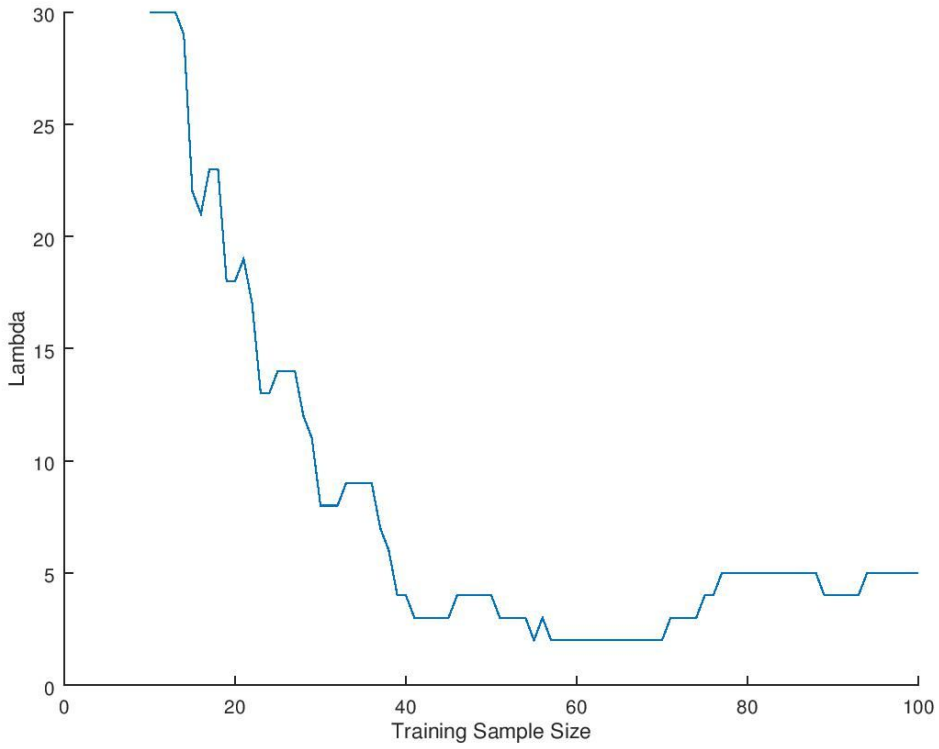
Assignment 3

Question 1. Implement the ridge-regression algorithm. **no need to submit your code**. Run the algorithm on the dataset `regdata.mat` provided on the course web page. Run the regression using $\lambda \in \{0, 1, 2, \ldots, 30\}$ on the training set X, Y provided in the data file. Try training-set sizes between 10 and 100. For each training set size that you try, find the value of λ that obtains the smallest mean-squared-error (the average squared loss) on the test set provided in the data file.

(a) (10pts) Submit a plot of the value of λ that minimizes the mean squared error on the test set as a function of the training set size m.



(b) (10pts) What trend to you expect in the plot based on what we learned in class? Explain.

Lambda is the parameter representing the trade off between the norm of W and the loss function. The higher lambda is the more we are encouraged to pick W with smaller norm.
We would expect that the more samples we have the less the norm should be taken into consideration and a smaller lambda should be used. A wrong predictor, meaning a predictor that gives wrong weight to features, will eventually "pay" on larger training set. This is because in high probability there will be one or more training samples that will "punish" him. Therefore we don't really need to consider the norm.
In the other hand, if we have a smaller training set, we might pick a predictor that supplied good results to the training set, but from the wrong intentions (wrong features) - the classic overfitting. Punish a predictor for its norm will encourage him to take into consideration only the right features.

(c) (5pts) Did you get this trend in the plot you submitted? If there are any differences, explain why they could occur.

Yes, this graph is decreasing, not monotonically but obviously decreasing.
After about 40 samples the required lambda is low(2-5). The results supports the theory we learned in class - the more samples we used a smaller lambda was needed to get smaller error.
There isn't any difference beside the noise in the end, it might have happened because all the lambdas in the region 2-5 gave small error rate.

(d) (15pts) In this data set, the label y of each example x was generated by setting $y = \langle w, x \rangle + \eta$, where w is a fixed vector which is the same for all examples in the data set, and η is a standard Gaussian random variable, $\eta \sim N(0, \sigma)$ for some $\sigma > 0$. η is drawn independently for each example in the data set. What is the Bayes-optimal predictor for this problem with respect to the squared loss? And how about the absolute loss? Prove your claims.

<u>squared loss:</u>

$h^*(x) = E[Y|X = x]$ (We learn in class that Bayes-optimal predictor for squared loss is the **expected value)**

$= E(< w, x > + \eta | X = x) = E(< w, x > | X = x) + E(\eta)$

Since <w,x> is constant and $\eta \sim N(\mu = 0, \sigma) \Rightarrow E[\eta] = 0$

So: $h^*(x) = < w, x >$ for this problem with respect to the squared loss.

<u>absolute loss:</u>

$h^*(x) = MEDIAN[Y|X = x]$ (We learn in class that Bayes-optimal predictor for absolute loss is the **MEDIAN value).**

$= M(< w, x > + \eta)$ (The operator M is homogeneous and keeps shifting) $= M(< w, x >) + M(\eta)$

Since <w,x> is constant and $\eta \sim N(\mu = 0, \sigma) \Rightarrow M[\eta] = \mu = 0$, by the properties of Normal distribution.

So: $h^*(x) = < w, x >$ for this problem with respect to the absolute loss.

Question 2. (10pts) We saw in class that the LASSO regression algorithm minimizes the following objective function:

$$\text{Minimize}_{w\in\mathbb{R}^d}\lambda\|w\|_1+\sum_{i=1}^{m}(\langle w,x_i\rangle-y)^2.$$

Write a quadratic program that solves this minimization program.

Answers 2. We will show how to solve LASSO regression using quadratic program.
First we will recall the settings of quadratic program:
$Minimize_{z\in R^n}\frac{1}{2}z^T * H * z + \ < u,z > \qquad subject\ to \qquad Az \geq v$
We will define:
$A_{2d+2m_x\,2d+m} = [\ Id_{d_x d}\ 0_{d_x m}\ -Id_{d_x d}\ ;\ -Id_{d_x d}\ 0_{d_x m}\ Id_{d_x d}\ ;\ X_{m_x d}\ -Id_{m_x m}\ 0_{m_x d}\ ;\ -X_{m_x d}\ Id_{m_x m}\ 0_{m_x d}\ ]$
v= $[\ 0_{2d_x 1}\ ;\ Y_{m_x 1}\ ;\ -Y_{m_x 1}\ ]$

The way A and v are defined forces z to look like:
$z = [w_{d_x 1};\ (X * w - Y)_{m_x 1}; |w|_{d_x 1}]$

Explanation:
$w_{d_x 1}$ - d "free" coordinates , the w we are looking for in LASSO.
$(X * W - Y)_{m_x 1}$ - notice that foreach coordinate i $d < i \leq d + m$ we force
$z_i \geq x_{i-d} * w - y_{i-d} \ \wedge\ z_i \leq x_{i-d} * w - y_{i-d} \rightarrow z_i = x_{i-d} * w - y_{i-d}$.
$|w|_{d_x 1}$ - for the last d coordinates we forced $z_{d+m+i} \geq z_i \wedge z_{d+m+i} \geq - z_{d+m+i}$ => $z_{d+m+i} \geq |z_i|$.
Since this is a minimization problem the vector z that achieve the minimum has to apply equality.

We will now show how to define H and u:
$H_{2d+m_x\,2d+m} = if\ (d < i,j \leq d + m) : H_{i,j} = 2,\ else : 0$
$u_{2d+m_x 1} = [0_{d+m_x 1}; \lambda_{d_x 1}]$

We will notice now:
$\frac{1}{2}z^T * H * z = \frac{1}{2}[w_{d_x 1};\ (X * w - Y)_{m_x 1}; |w|_{d_x 1}]^T * H * [w_{d_x 1};\ (X * w - Y)_{m_x 1}; |w|_{d_x 1}] =$
$[0_{d_x 1};\ (X * w - Y)_{m_x 1}; 0_{\ d_x 1}]^T * [w_{d_x 1};\ (X * W - Y)_{m_x 1}; |w|_{d_x 1}] = z = (X * W - Y)^2$

$< u,z > = < [0_{d+m_x 1}; \lambda_{d_x 1}], [w_{d_x 1};\ (X * w - Y)_{m_x 1}; |w|_{d_x 1}] > \ = |w|$

Therefore we managed to transform the LASSO problem to quadratic program problem. Given z, the output of the quadratic program solver, the first d coordinates of z are the vector w we are looking for in LASSO.

Question 3. Let $\mathcal{X}$ be the set of all the undirected graphs over n vertices and let $\mathcal{Y} = \{0, 1\}$.

For a graph $x \in \mathcal{X}$, define the mapping $g : \mathcal{X} \to \mathbb{N}^n$, where coordinate i in the vector $g(x)$ is the number of i-cliques in the graph x.

Let $\mathcal{H}$ be a hypothesis class which includes all the functions $h : \mathcal{X} \to \mathcal{Y}$ that have the form: $h(x) \equiv \mathbb{I}[g(x) = v]$, for some vector $v \in \mathbb{N}^n$.

Suppose that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$, and suppose that in this distribution, the label of a graph is completely determined by the numbers of i-cliques that the graph has for the various i.

(a) (10pts) Use the PAC-learning upper bounds that we showed in class to show that the sample complexity of learning $\mathcal{H}$ is $O(n^2 \log(n))$.

The PAC learning upper bound claims that any ERM algorithm with training sample size $m \geq \frac{2*log(|H|)+2*log(2/\delta)}{\varepsilon^2}$ gets an error of at most ε, with a probability of at least $1-\delta$ over the random training samples($\varepsilon, \delta \in (0,1)$). Given ε, δ, we find out that the sample size needed is at least $\frac{2}{\varepsilon^2} * log(|H|) + \frac{2*log(2/\delta)}{\varepsilon^2}$. Since $\frac{2}{\varepsilon^2}$, $\frac{2*log(2/\delta)}{\varepsilon^2}$ are constant values, it terms of big O the sample complexity depends on $log(|H|)$. We will now show an upper bound for the size of H. If we could find the number of different outputs of $g(x)$, $x \in X$, we could find the number of different predictors. The i coordinate in g(x) is the number of i-clique in the graph x. The number of i-clique possible is at most $\binom{n}{i}$. Since i is between 1 to n, the coordinate with the highest value is for i= $\frac{n}{2}$, for this i there are at most $\frac{n!}{(\frac{n}{2})!*(\frac{n}{2})!} \leq n^{\frac{n}{2}}$ .Because there are n coordinates and each coordinate is a natural number which isn't greater than $n^{\frac{n}{2}}$, the number of different $g(x)$ is at most $(n^{\frac{n}{2}})^n$,
hence |H|= $(n^{\frac{n}{2}})^n = (n^{\frac{n^2}{2}})$
This means, the sample complexity is O(log(|H|))= $O(log(n^{\frac{n^2}{2}})) = O(n^2 * log(n^{\frac{n}{2}})) = O(n^2 * log(n))$, as required.

(b) (10pts) Adam and Ronnie want to learn a classifier that is guaranteed a low error with a probability of at least 95%. Ronnie used a training set of size m drawn from $\mathcal{D}$ to learn a classifier from $\mathcal{H}$. How many examples should Adam feed his classifier, so that his classifier has an error guarantee which is half the error guarantee of Ronnie's? Explain.

Lets define ε the error Ronnie gets with here training sample m. From PAC upper bound in probability of 0.95, $m \geq \frac{2(log(|H|)*log(\frac{2}{0.05})}{\varepsilon^2}$ => $\varepsilon \geq \sqrt{\frac{2(log(|H|)*log(40)}{m}}$. Adam wants his error to be half the error Ronne has. Therefore Adam wants to get with probability of 0.95 an error of $\frac{\varepsilon}{2}$. From PAC this means m', the sample complexity of Adam, is:
$m' \geq \frac{2(log(|H|)*log(\frac{2}{0.05})}{(\frac{\varepsilon}{2})^2} = 4 * \frac{2(log(|H|)*log(\frac{2}{0.05})}{(\varepsilon)^2}$. Since epsilon is at least $\sqrt{\frac{2(log(|H|)*log(40)}{m}}$, we get:

$4 * \frac{2(log(|H|)*log(\frac{2}{0.05})}{(\varepsilon)^2} \leq 4 * \frac{2(log(|H|)*log(\frac{2}{0.05})}{\frac{2(log(|H|)*log(40)}{m}} = 4 * m$. Therefore if $m' \geq 4m$ in probability of 0.95 Adam's error guaranteed to be half the error guaranteed for Ronnie.

Question 4. Consider the following optimization objective for a regression problem, where $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathbb{R}$:

$$\text{Minimize}_{w \in \mathbb{R}^d} \lambda \|w\|_2^4 + \sum_{i=1}^{m} \exp^{|\langle w, x_i \rangle - y_i|}$$

(a) (7pts) Does the Representer Theorem apply for this objective function? Prove your claim.

Answers 4.a. We will proof that this objective function, the Representer Theorem applies. Given a feature map $\psi : R^{\lambda} \to R^n$, the representer theorem claims that if the objective can be represented as:
$Minimize\, f(< w, \psi(x_1) >, ... < w, \psi(x_m) >) + R(||w||)\ s.t\ f : R^m \to R\ and\ R : R_+ \to R\ monotonic\ non\ decreasing$,
Then the Representer Theorem applies and the optimal solution to the objective can be written as:
$w = \sum_{i=0}^{m} \alpha(i)\psi(x_i)\ where\ \alpha \in R^m.$
Therefore, we need to show that there exists f and R that satisfy the conditions and
$Minimize_{w \in R^m} \lambda ||w||_2{}^4 + \sum_{i=1}^{m} exp(| < w, \psi(x_i) > - y_i|) = Minimize_{w \in R^m} f(< w, \psi(x_1) > ..... < w, \psi(x_m) >) + R(||w||)$.
Lets define $f(t_1, ..., t_n) = \sum_{i=1}^{m} exp(|t_i - y_i|),\ R(t) = \lambda * t^4$. we will now tonic that:
$Minimize_{w \in R^m} f(< w, \psi(x_1) > ..... < w, \psi(x_m) >) + R(||w||) = Minimize_{w \in R^m} \lambda ||w||_2{}^4 + \sum_{i=1}^{m} exp(| < w, \psi(x_i) > - y_i|)$.
We just need to stasify the conditions $f : R^m \to R\ and\ R : R_+ \to R\ monotonic\ non\ decreasing$. f is a function that gets m real numbers and returns a single real number, hence $f : R^m \to R$. R is a function that for every input t returns a real number. We just left to show that R is *monotonic non decreasing On* R_+.
Let $x, y > 0, x \le y$ we will show that $f(x) \le f(y)$:
$f(x) = \lambda x^4,\ \lambda \ge 0\ \Rightarrow\ \lambda x^4 \le \lambda y^4\ = f(y)\ \Rightarrow\ f(x) \le f(y)$.

Answers 4.b.
From the section above, we know that the following problem applies the Representer Theorem. Therefore, the optimal solution to the objective can be written as $w = \sum_{i=1}^{m} \alpha_i \psi(x_i),\ \forall i,\ \alpha_i \in R$. Let w be the optimal solution for the problem, therefore $\lambda ||w||_2{}^4 + \sum_{i=1}^{m} exp(| < w, \psi(x_i) > - y_i|)$ is minimal for w over all $w' \in R^n$.

$$\lambda || \sum_{i=1}^{m} \alpha_i \psi(x_i)||_2{}^4 + \sum_{i=1}^{m} exp(| < \sum_{j=1}^{m} \alpha_j \psi(x_j), \psi(x_i) > - y_i|) \le \lambda ||w'||_2{}^4 + \sum_{i=1}^{m} exp(| < w, \psi(x_i) > - y_i|)$$

Since w is optimal, it can be represented as $w = \sum_{i=1}^{m} \alpha_i \psi(x_i),\ \forall i,\ \alpha_i \in R$:

$$\lambda || \sum_{i=1}^{m} \alpha_i \psi(x_i)||_2{}^4 + \sum_{i=1}^{m} exp(| < \sum_{j=1}^{m} \alpha_j \psi(x_j), \psi(x_i) > - y_i|) = \lambda < \sum_{i=1}^{m} \alpha_i \psi(x_i), \sum_{i=1}^{m} \alpha_i \psi(x_i) >^2 + \sum_{i=1}^{m} exp(|(\sum_{j=1}^{m} \alpha_j < \psi(x_j), \psi(x_i) >) - y_i|) =$$

$$\lambda(\sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i\ \alpha_j < \psi(x_i), \psi(x_j) >)^2 + \sum_{i=1}^{m} exp(|(\sum_{j=1}^{m} \alpha_j < \psi(x_j), \psi(x_i) >) - y_i|) = \lambda(\sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i\ \alpha_j K(x_i, x_j))^2 + \sum_{i=1}^{m} exp(|(\sum_{j=1}^{m} \alpha_j K(x_j, x_i)) - y_i|)$$

Therefore, we want to find a vector $a \in R^m$ that minimize the expression:
$\lambda(\sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i\ \alpha_j K(x_i, x_j))^2 + \sum_{i=1}^{m} exp(|(\sum_{j=1}^{m} \alpha_j K(x_j, x_i)) - y_i|)$.
We will notice that all vectors in this expression are ‘m’ length.

Question 5. (10pts) Consider the following loss function for regression:

$$\ell(y, y') = (\ln(y) - \ln(y'))^2.$$

Calculate the Bayes-optimal predictor for this loss, $\operatorname{argmin}_{h:\mathcal{X}\to\mathcal{Y}} \ell(h, \mathcal{D})$. Prove your claim using the definition of a Bayes-optimal predictor.

Answers 5. **Claim:** $h^*(x) = e^{ln[Y|X=x]}$

Proof (for discrete distributions)**:**

$E_{(X,Y)\sim D}[(ln(h(X)) - ln(Y))^2 = E_X[E_Y[(ln(h(X)) - ln(Y))^2|X = x]]$

$\Rightarrow$ *Bayes − optimal minimizes* $E_Y[(ln(h(X)) - ln(Y))^2|X = x]$ *for each* $x \in X$.

Let $v_1, v_2, \ldots$ *be the possible values of* Y, *and Let* $p_i = P[Y = v_i| X = x]$.

Then:

$$h^*(x) = argmin \sum_{i=1}^{\infty} p_i(ln(v_i) - ln(b))^2, \quad of\ b.$$

Observe $f(b) = \sum_{i=1}^{\infty} p_i(ln(v_i) - ln(b))^2$.

In order to find the **minimum** of the function we will have to derivate:

$f'(b) = \sum_{i=1}^{\infty} 2p_i(ln(v_i) - ln(b)) \cdot \frac{1}{b} = \frac{1}{b} \cdot 2 \cdot \sum_{i=1}^{\infty} p_i(ln(v_i) - ln(b))$

$f'(b) = 0 \Rightarrow \frac{2}{b} \sum_{i=1}^{\infty} p_i(ln(v_i) - ln(b)) = 0 \Rightarrow \sum_{i=1}^{\infty} p_i(ln(v_i) - ln(b)) = 0$

$\Rightarrow \sum_{i=1}^{\infty} p_i \cdot ln(v_i) = ln(b) \cdot \sum_{i=1}^{\infty} p_i$, (*notice that* $\sum_{i=1}^{\infty} p_i = 1$) $\Rightarrow \sum_{i=1}^{\infty} p_i \cdot ln(v_i) = ln(b)$

$\Rightarrow b = e^{\sum_{i=1}^{\infty} p_i(ln(v_i))}$

So: $h^*(x) = e^{E[ln(Y)|X=x]}$