# COMPSCI 671D Spring 2019
# Homework 5

## 1 Probabilistic Interpretation of Ridge Regression

*Please try this problem yourself before looking up – that is the best way to learn.*

Consider a regression problem in which we have data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. As usual, we can stack all our features in a $n \times p$ matrix, $\mathbf{X}$, and all our labels in a $n \times 1$ vector $\mathbf{y}$. We would like to use ridge regression to make our prediction, i.e. $f(\mathbf{x}) = \mathbf{x}\beta^*$, and:

$$\beta^* = \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda||\beta||_2^2, \tag{1}$$

where $\lambda$ is a hyperparameter that controls the amount of regularization. Note that, when $\lambda = 0$ we get the sum of squares loss from the least squares problem. In this problem we are going to show that ridge regression has a probabilistic interpretation as a prior probability distribution over the feature weights.

**a)** Derive the closed form solution to (**??**) by setting the gradient of the loss function equal to 0. Now consider the following probabilistic model:

$$\Pr(\mathbf{y}|\mathbf{X}, \beta) = \mathcal{MVN}(\mathbf{X}\beta, \mathbf{I}_{n\times n})$$
$$\Pr(\beta|\lambda) = \mathcal{N}(0, \frac{1}{\lambda}\mathbf{I}),$$

where $\mathbf{I}_{n\times n}$ is the $n \times n$ identity matrix, and we will use $\mathbf{I}$ as a shorthand for the $p \times p$ identity matrix. Here, $\lambda > 0$ is a fixed and known precision parameter for the prior. Under this model, the full data has likelihood:

$$\Pr(\mathbf{y}|\mathbf{X}, \beta) = (\sqrt{2\pi})^{-n/2}|\mathbf{I}_{n\times n}|^{-1} \exp\left\{-\frac{1}{2}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{I}_{n\times n})^{-1}(\mathbf{y} - \mathbf{X}\beta)\right\}.$$

**b)** Using Bayes' Rule show that, up to a proportionality constant, the posterior pdf of $\beta|\mathbf{y}, \mathbf{X}, \lambda$ can be written as:

$$\Pr(\beta|\mathbf{y}, \mathbf{X}, \lambda) \propto \exp\left\{-\frac{1}{2}\left[\beta^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\beta - 2\beta^T\mathbf{X}\mathbf{y}\right]\right\}$$

Hint: "up to a proportionality constant" also includes "drop all terms that don't include $\beta$," because those terms do not enter into the estimation of $\beta$ in practice.

**c)** Show that, up to a proportionality constant, the pdf in b) can be expressed as:

$$\Pr(\beta|\mathbf{y},\mathbf{X},\lambda) \propto \exp\left\{-\frac{1}{2}(\beta-(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y})^T(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})(\beta-(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y})\right\}.$$

**d)** The result in part c) looks very similar to part of the pdf of a very popular distribution. What is this distribution? Give its parameters for the case in c). Show that the Maximum a-Posteriori (MaP) estimate (this is just the maximum of the posterior distribution) for $\beta$ under this distribution is also the solution to the ridge minimization problem in (**??**), i.e. show that:

$$\beta_{MaP} = \arg\max \Pr(\beta|\mathbf{y},\mathbf{X},\lambda) = \beta^*.$$

You can do this by appealing to known properties of the distribution you have found.

## 2 Statistical Learning Theory: Sub-Gaussian Bounds

For a random variable $X$ with $\mathbb{E}[X]=0$, we say that $X$ is sub-Gaussian with variance proxy $\sigma^2$ if its distribution satisfies the following:

$$\mathbb{E}[e^{sX}] \leq e^{\frac{\sigma^2 s^2}{2}}, \quad \forall\, s \in \mathbb{R}. \tag{2}$$

Most known families of distributions are sub-Gaussian, and assuming that the data's distribution is sub-Gaussian is generally considered a mild assumption, especially in comparison to assuming that the data comes from a specific distribution. While in class we have looked at distribution-free bounds, in this question we derive some useful bounds that come from assuming that the data is sub-Gaussian, a mild, yet stronger, assumption. In this problem we will show that sub-Gaussian random variables have tails that decay exponentially fast: the probability of selecting a value $t$ far away from 0 decays exponentially fast as the value $t$ gets further and further away from 0. We will also employ a method known as Chernoff's bounding method (Chernoff 1952), which is widely applied in statistical learning theory to construct many different types of bounds.

**a)** Prove that, for all real numbers $t>0$, for sub-Gaussian distribution $\mathcal{D}$:

$$\Pr_{X\sim\mathcal{D}}(X>t) \leq e^{\frac{\sigma^2 s^2}{2}-st},$$

for all $s>0$.
Hint: $e^X$ is always positive, so Markov's inequality applies in this case.

**b)** Starting from the result in part a) prove that

$$\Pr_{X\sim\mathcal{D}}(X>t) \leq e^{-\frac{t^2}{2\sigma^2}},$$

for all $t>0$.
Hint: The result in part a) is true for all $s>0$ but to get the tightest possible bound we would want $s$ that makes the bound in a) the smallest possible. This looks a lot like a minimization problem...

**c)** Let $X_1, \dots, X_n$ be $n$ independent sub-Gaussian random variables all with the same variance proxy, $\sigma^2$. Using the tools employed in parts a) and b) prove that, for all $t > 0$:

$$\Pr_{X \sim \mathcal{D}} \left( \frac{1}{n} \sum_{i=1}^{n} X_i > t \right) \leq e^{\frac{-nt^2}{2\sigma^2}}.$$

Hint: First show that $\mathbb{E}[e^{s \sum_{i=1}^{n} X_i}] \leq e^{\frac{\sigma^2 s^2}{2}}$, then use a Chernoff bound like you did in parts a) and b).
Hint: Try setting $s = \frac{1}{n}$ in your final bound.

## 3 K-Means as an Optimization Problem

For a dataset $D = \{(\mathbf{x}_i)\}_{i=1}^{n}$, $\mathbf{x}_i \in \mathbb{R}^p$, the K-means algorithm wants to find $k$ centers of cluster such that the sum of distances between the cluster center and samples in that cluster are minimized. The algorithm operates iteratively by first assigning each observation to its nearest centroid and then by updating the centroids with the average value of the observations assigned to them.

**a)** Write down the mathematical form of the objective function for K-means using Euclidean distance as the distance function. You are welcome to use lecture notes to help you.

**b)** Let $L(\tilde{\mathbf{x}}^{(t)})$ be the objective function for K-means, where $\tilde{\mathbf{x}}^{(t)}$ are the centroids of the K clusters at iteration $t$ of the algorithm. Prove that K-Means will converge. You can use the fact that an optimization algorithm will converge when its objective function is both monotonic and bounded.

**c)** Construct a toy dataset in which K-Means does not always converge to the globally optimal solution.
Hint: You can do this with only 3 or 4 data points and proper initialization of cluster centers.

**d)** Why does K-Means fail to find the globally optimal solution? Given this result, does the Gaussian Mixture Model (GMM) always find the globally optimal solution? You can use known facts about both algorithms and their loss functions to answer this question.

## 4 EM Algorithm for Topic Modeling

In this question we will try to design an algorithm for discovering the abstract topics that occur in a set of documents. In the problem, we have a set of $M$ abstract documents in the universe, $\mathcal{D}$, which are mixtures of latent topics. We also have a dictionary of words, $\mathcal{W}$, with size $N$. Intuitively, $\mathcal{W}$ is the list of all unique words that occur at least once within $\mathcal{D}$. Using these two sets we can denote the number of times word $w_j$ occurs in document $d_i$ as $n(w_j, d_i)$. It follows that we can represent a document $d_i$ as a vector of size $N$, each entry of which corresponds to how many times word $w_j$ appears in it. The topics $z$ are latent variables in a topic set $\mathcal{Z}$ sized $K$. Intuitively, if a document is about certain topic it will include some particular words with higher probability. For example, "music", "show" and "film" appear frequently in documents about Arts while "school", "student" and "teachers" usually occur if the document is about Education. Meanwhile, a document can be a mixture of several topics, e.g., a document can be about arts education for high school students. Inspired by the intuition, a reasonable approach to model the generative process is as follows.

$$\Pr(d_i, z_k, w_j) = \Pr(d_i)\Pr(z_k|d_i)\Pr(w_j|z_k)$$

which means the topics only depend on the documents and the words only depend on topics. For computational convenience, let $\Pr(d_i)$ be a uniform distribution and $\Pr(z_k|d_i)$ and $\Pr(w_j|z_k)$ be Multinomial distributions. i.e.

$$\Pr(d_i) = \frac{1}{M}$$

$$\Pr(z_k|d_i) = \alpha_{ik}, \; \sum_{k=1}^{K} \alpha_{ik} = 1$$

$$\Pr(w_j|z_k) = \beta_{kj}, \; \sum_{j=1}^{N} \beta_{kj} = 1.$$

We are interested in learning $\alpha_{ik}$ and $\beta_{kj}$ because they are substantively interesting: $\alpha_{ik}$ represents the proportion of topic $k$ that makes up document $i$, and $\beta_{kj}$ the probability of seeing word $j$ under topic $k$. Therefore, a single word $w_j$ in a document $d_i$ is generated in this way: (1) Choose a topic $z_k$ from the topic distribution $Pr(z_k|d_i)$, the probability of choosing topic $k$ is $\alpha_{ik}$; (2) Choose a word from the vocabulary distribution $p(w_j|z_k)$ in this topic, the probability is $\beta_{kj}$. We want to learn these parameters by maximizing the likelihood of the data using the EM algorithm.

**a)** For our set of $N$ documents and $W$ word of choices, write down the log-likelihood of the model above, i.e. $log(p(D = d_i, W = w_j|\alpha, \beta))$.

Remember that in the EM (Expectation-Maximization) algorithm, we can figure out the parameters and the hidden variables by iteratively computing (1) the distribution of latent variables using the old parameters and (2) find new parameters that maximize the likelihood using the old latent variable distribution. Now try to:

**b)** E-step: Derive the distribution of latent variable $p(z_k|w_j, d_i, \alpha^{old}, \beta^{old})$ by fixing the old parameters.

**c)** M-step: Find the $\alpha^{new}$ and $\beta^{new}$ that optimize the log likelihood using $p(z_k|w_j, d_i, \alpha^{old}, \beta^{old})$ as the distribution of $z$.

You would iterate these until convergence to get the final parameters in practice. Just so you know, the model you have just been working with in this problem is a very famous and very popular model. :)

## 5 What Happens to My Gradient

In this problem you will get the chance to construct a neural network with architecture $784 - H - H - 1$, where $H$ is the number of hidden nodes you choose. This neural network can be used for binary classification, such 0, 1 digits classification for MNIST. The model can be represented as

$$f(\mathbf{x}; \theta) : \mathbb{R}^{784} \rightarrow [0, 1]$$

where $\theta = \{\mathbf{W}_1 \in \mathbb{R}^{784\times H}, \mathbf{b}_1 \in \mathbb{R}^H, \mathbf{W}_2 \in \mathbb{R}^{H\times H}, \mathbf{b}_2 \in \mathbb{R}^H, \mathbf{w}_3 \in \mathbb{R}^H, b_3 \in \mathbb{R}, \}$. Explicitly, $f(\mathbf{x})$ is

$$\mathbf{h}_1 = \sigma\left(\mathbf{W}_1^\top \mathbf{x} + \mathbf{b}_1\right)$$

$$\mathbf{h}_2 = \sigma\left(\mathbf{W}_2^\top \mathbf{h}_1 + \mathbf{b}_2\right)$$

$$f(\mathbf{x}) = \sigma\left(\mathbf{w}_3^\top \mathbf{h}_2 + b_3\right)$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$, and it is an element-wise operator, which means if $\mathbf{x} = (x^1, x^2, ..., x^d) \in \mathbb{R}^d$, we have $\sigma(\mathbf{x}) = (\sigma(x^1), \sigma(x^2), ..., \sigma(x^d))$. The loss function for this model (or the negative log-likelihood) is the usual one:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N}(y_i \log f(\mathbf{x}_i) + (1-y_i)\log(1-f(\mathbf{x}_i)))$$

**a)** Derive the gradients $\nabla_{\mathbf{w}_3}\mathcal{L}(\theta)$, $\nabla_{b_3}\mathcal{L}(\theta)$, $\nabla_{\mathbf{W}_2}\mathcal{L}(\theta)$, $\nabla_{\mathbf{b}_2}\mathcal{L}(\theta)$, $\nabla_{\mathbf{W}_1}\mathcal{L}(\theta)$, $\nabla_{\mathbf{b}_1}\mathcal{L}(\theta)$ by backpropagation.
Hints: The problem would be much easier if you define $\mathbf{z}_l = \mathbf{W}_l^T \mathbf{h}_{l-1} + \mathbf{b}_l$, and derive the relationship between $\nabla_{\mathbf{z}_l}\mathcal{L}(\theta)$ and $\nabla_{\mathbf{z}_{l-1}}\mathcal{L}(\theta)$. Also, you can give the final result in a recursive way as long as it is correct and consistent.

**b)** Use the preprocessed MNIST data provided in the attachment. MNIST is a dataset of images of handwritten digits that also contain labels for the number they correspond to. We want to train a neural network to learn to recognize these handwritten digits. Use gradient descent with the gradients you have derived in part a) to train a neural network on the data. Report the accuracy for the network on the train and test data. Since the sample size is large, we can approximate the true gradient with stochastic gradient for computational convenience:

$$\nabla\mathcal{L}(\theta) \approx \frac{1}{B}\sum_{j=1}^{B}\nabla\mathcal{L}(\theta)_j$$

which means at each step, instead of calculating the gradients with all samples in the dataset, we randomly pick a mini-batch of samples and calculate gradient using these samples. In implementing the neural network, it would be beneficial to build and train the model in a general way. That is, build a NN with $d$ hidden layers such that we can easily modify $d$.

**c)** In our current model, we have 2 hidden layers. What is the magnitude of $\nabla_{\mathbf{W}_1}\mathcal{L}(\theta)$? Now try to modify the the number of hidden layers $d$, from 1 to 4. What happens to the magnitute of $\nabla_{\mathbf{W}_1}\mathcal{L}(\theta)$ when we increase $d$? In this question, you don't need to train the whole neural network, just initialize it with random weights, e.g. $\mathcal{N}(0,1)$, and calculate the gradient.

**d)** Suppose that all weights are finite and the weight matrices $\mathbf{W}_2, \mathbf{W}_3, ..., \mathbf{W}_d$ are diagonalizable matrices in which the absolute value of the eigenvalues are upper-bounded by $4-\epsilon$. Prove that

$$\lim_{d\to\infty}\nabla_{\mathbf{W}_1}\mathcal{L}(\theta) = \mathbf{0}$$

Hints: Try to use the the relationship between $\nabla_{\mathbf{z}_l}\mathcal{L}(\theta)$ and $\nabla_{\mathbf{z}_{l-1}}\mathcal{L}(\theta)$ and try to apply a common inequality.