6.867 Machine learning and neural networks

Problem set 1

Due September 20, in class

What and how to turn in?

Turn in short written answers to the questions explicitly stated, and when requested to explain or prove. Do **not** turn in answers when requested to "think", "consider", "try" or "experiment" (except when specifically instructed). You may turn in answers to questions marked "optional"— they will be read and corrected, but a grade will not be recorded for them.

Turn in all MATLAB code explicitly requested, or that you used to calculate requested values. It should be clear exactly what command was used to get the answer to each question.

To help the graders (including yourself...), please be neat, answer the questions briefly, and in the order they are stated. Staple each "Problem" separately, and be sure to write your name on the top of every page.

Problem 1: regression

Reference: Lecture two; Chapter five.

Here we will be using a regression method to predict housing prices in suburbs of Boston. You'll find the data in the file "housing.data". Information about the data, including the column interpretation can be found in the file "housing.names". These files, like many other data files in the course, are taken from the UCI Machine Learning Repository `http://www.ics.uci.edu/ mlearn/MLSummary.html`. They are provided also on the course web page, and in the course Athena locker, `/mit/6.867`.

We will predict the median house value (the 14th, and last, column of the data) based on the other columns.

1. First, we will use a linear regression model to predict the house values, using squared-error as the criterion to minimize. In other words $y = f(\mathbf{x}; \hat{\mathbf{w}}) = \hat{w}_0 + \sum_{i=1}^{13} \hat{w}_i x_i$,

where $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{t=1}^{n} (\mathbf{y_t} - f(\mathbf{x_t}; \mathbf{w}))^2$; here y_t are the house values, x_t are input vectors, and n is the number of training examples.

Write the following MATLAB functions (these should be simple functions to code in MATLAB):

- A function that takes as input weights $\mathbf{w}$ and a set of input vectors $\{\mathbf{x}_t\}_{t=1,\ldots,n}$, and returns the predicted output values $\{y_t\}_{t=1,\ldots,n}$
- A function that takes as input training input vectors and output values, and return the optimal weight vector $\hat{\mathbf{w}}$.
- A function that takes as input a training set of input vectors and output values, and a test set input vectors, and output values, and returns the mean training error (i.e. average squared-error over all training samples) and mean test error.

2. To test our linear regression model, we will use part of the data set as a training set, and the rest as a test set. For each training set size, use the first lines of the data file as a training set, and the remaining lines as a test set. Write a MATLAB function that takes as input the complete data set, and the desired training set size, and returns the mean training and test errors.

   Turn in the mean squared training and test errors for each of the following training set sizes: 10, 50, 100, 200, 300, 400.

   (Quick validation: For a sample size of 100, I got a mean training error of 4.15 and a mean test error of 1328)

3. What condition must hold for the training input vectors so that the training error will be zero for any set of output values?

4. Do the training and test errors tend to increase or decrease as the training set size increases? Why? Try some other training set sizes to see that this is only a tendency, and sometimes the change is in the different direction.

5. We will now move on to polynomial regression. We will predict the house values using a function of the form:

$$f(\mathbf{x}; \mathbf{w}) = w_0 + \sum_{i=1}^{13} \sum_{d=1}^{m} w_{i,d} x_i^d$$

   Where again, the weights $\mathbf{w}$ are chosen so as to minimize the mean squared error of the training set. Think about why we also include all lower order polynomial terms up to the highest order rather than just the highest ones [do not turn in an answer].

   Note that we only use features which are powers of a single input feature. We do so mostly in order to simplify the problem. In most cases, it is more beneficial to use features which are products of different input features, and perhaps also their powers.

Think of why such features are usually more powerful [you do not have to turn in an answer].

Write a version of your MATLAB function from section 2 that takes as input also a maximal degree m and returns the training and test error under such a polynomial regression model.

NOTE: When the degree is high, some of the features will have extremely high values, while others will have very low values. This causes severe numeric precision problems with matrix inversion, and yields wrong answers. To overcome this problem, you will have to appropriately scale each feature x_i^d included in the regression model, to bring all features to roughly the same magnitude. Be sure to use the same scaling for the training and test sets. For example, divide each feature by the maximum absolute value of the feature, among all training and test examples. (MATLAB matrix and vector operations can be very useful for doing such scaling operations easily)

6. Prove that such scaling of features does not change the regression predictions. That is, given training feature vectors and output values $\{\mathbf{x}_t, y_t\}_{t=1,\ldots,n}$ and a test input vector $\mathbf{x}_{\text{test}}$, and scaling factors $\{\alpha_i\}_{i=1,\ldots,13}$, we would like to prove that the prediction of the test output value would be the same if we trained a linear regression on $\{\tilde{\mathbf{x}}_t, y_t\}_{t=1,\ldots,n}$, where $\tilde{x}_{t,i} = \alpha_i x_{t,i}$, and predicted on $\tilde{\mathbf{x}}_{\text{test}}$, where $\tilde{x}_{\text{test},i} = \alpha_i x_{\text{test},i}$. (It is enough to prove this for the linear model (maximum degree one), and this is what we require you to prove. The result extends to scaling each "extended" feature x_i^d (which is what we actually do), since this is just linear regression using these "extended" features).

7. For a training set size of 400, turn in the mean squared training and test errors for maximal degrees of zero through ten.

   (Quick validation: for maximal degree two, I got a training error of 14.5 and a test error of 32.8).

8. Explain the qualitative behavior of the test error as a function of the polynomial degree. Which degree seems to be the best choice?

9. Prove (in two sentences) that the training error is monotonically decreasing with the maximal degree m. That is, that the training error using a higher degree and the same training set, is necessarily less then or equal to the training error using a lower degree.

10. We claim that if there is at least one feature (component of the input vector $\mathbf{x}$) with no repeated values in the training set, then the training error will approach zero as the polynomial degree increases. Why is this true?

Problem 2: estimation

Reference: Lecture 1-2; Recitations; Chapter four.

In this problem, we will derive maximum likelihood, and MAP, estimators for parameters of Gaussian distributions.

Recall that a univariate Gaussian (or normal) random variable, with mean μ and variance σ^2, is given by the following probability density function:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Maximum Likelihood Estimation

The *likelihood* $L(\text{data}; \theta)$ is the probability (or probability density, for continuous distributions) of the data given the model parameters θ (here the model is a Gaussian). Note that we could have written $P(\text{data}|\theta)$ for the likelihood. The likelihood notation is used to emphasize that $L(\text{data}; \theta)$ is viewed as a function of the parameters θ when we have already observed the data.

1. Write down the likelihood $L(x_1, \ldots, x_n; \mu, \sigma)$ of a sample drawn independently from a normal distribution with (unknown) mean and variance.

   The maximum likelihood estimator $\hat{\mu}(x_1, \ldots, x_n)$, is the value of μ that maximizes the likelihood L:

   $$\hat{\mu}(x_1, \ldots, x_n) = \arg\max_{\mu} \max_{\sigma} L(x_1, \ldots, x_n; \mu, \sigma)$$

   Instead of searching for the maximum of L, we will search for the maximum of $\log L$. This is fine since the logarithm is a monotonically increasing function. To find the maximum, we would like to solve the equation:

   $$\frac{\partial \log L(x_1, \ldots, x_n; \mu, \sigma)}{\partial \mu} = 0$$

2. Calculate $\frac{\partial \log L(x_1, \ldots, x_n; \mu, \sigma)}{\partial \mu}$ and solve the above equation, in order to find the maximum likelihood estimator $\hat{\mu}$. Show that the solution does not depend on σ.

   In general, we might have needed to find the values of σ which maximize L together with μ. This is luckily unnecessary, since as you showed, $\arg\max_{\mu} L(\mu, \sigma)$ is independent of σ.

   Note that $\hat{\mu}$ is a function of the sampled values, and thus $\hat{\mu}$ can itself be viewed as a random variable. An estimator such as $\hat{\mu}$ is said to be *unbiased* if the expected value of this random variable is equal to the "true" value being estimated, that is if $\mathbf{E}_{X_1, \ldots, X_n \sim \mathcal{N}(\mu, \sigma)} [\hat{\mu}(X_1, \ldots, X_n)] = \mu$ for all μ, σ. The expectation here is over the possible choices of the random samples assuming they came from a Gaussian with mean μ and variance σ^2.

3. Calculate $\mathbf{E}_{X_1,\dots,X_n \sim \mathcal{N}(\mu,\sigma)}\left[\hat{\mu}(X_1,\dots,X_n)\right]$. Is $\hat{\mu}$ unbiased? Hint: the expectation of a sum is equal to the sum of the expectations.

   We now proceed to calculate the maximum likelihood estimator for σ:

   $$\hat{\sigma}(x_1,\dots,x_n) = \arg\max_{\sigma}\max_{\mu} L(x_1,\dots,x_n;\mu,\sigma)$$

   We do so in a similar way, by taking the derivative of $\max_\mu \log L(x_1,\dots,x_n;\mu,\sigma)$, with respect to σ. Note that in taking this derivative, we assume that μ is set to its maximum likelihood value. However, we already know the value of μ that maximizes $L(\mu,\sigma)$ and so can just plug it in.

4. Does it matter if we take the derivative with respect to the variance σ^2, or its square root σ?

5. Calculate $\hat{\sigma}(x_1,\dots,x_n)$.

6. We would now like to show that $\hat{\sigma}^2$ is not an unbiased estimator of σ^2. Calculate $\mathbf{E}_{X_1,\dots,X_n \sim \mathcal{N}(\mu,\sigma)}\left[\hat{\sigma}^2(X_1,\dots,X_n)\right]$ to do so. Hint: note that $X_1,\dots,X_n$ are independent, and use the fact that the expectation of a product of independent random variables is the product of the expectations.

7. Suggest an unbiased estimator $\tilde{\sigma}^2(x_1,\dots,x_n)$ for σ^2, based on the the maximum likelihood estimator above, and show that $\tilde{\sigma}^2$ is in fact unbiased. Hint: scale the maximum likelihood estimator so that it will be unbiased.

8. Consider a sample $x_1,\dots,x_n$ drawn from a Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$, where the true mean μ is known, but the variance is not. What is the maximum likelihood estimator for the variance in this case ? Is it unbiased ?

   We now return to the case in which neither the mean nor the variance are known.

9. An estimator being unbiased does not necessarily make it good. For example, consider the following estimator for the mean of a Gaussian random variable: $\breve{\mu}(x_1,\dots,x_n) = x_1$. Show that this is an unbiased estimator of μ.

   One reason that $\breve{\mu}$ is not a very good estimator, is that no matter how many samples we have, it will not improve. It will never converge to the true value of μ.

   An estimator $\hat{\theta}$ is (mean squared) *consistent* if it converges to θ in the following sense: $\mathbf{E}_{X_1,\dots,X_n \sim \mathcal{N}(\mu,\sigma)}\left[(\hat{\theta}(X_1,\dots,X_n) - \theta)^2\right] \to 0$ as $n \to \infty$. In other words, the more data points we get, the less likely it is that the estimate $\hat{\theta}(X_1,\dots,X_n)$ deviates much from θ.

10. (*optional*) Show that $\hat{\mu}$ (the maximum likelihood estimate of the mean) is a consistent estimator of μ.

11. Do you think $\hat{\sigma}^2$ is a consistent estimator of σ^2? What about $\tilde{\sigma}^2$? (no proof required)

Maximum A-Posteriori (MAP) Estimation

So far we discussed maximum likelihood estimation. Sometimes we have information or beliefs about likely values of the parameters before actually having seen the data. It turns out that we can incorporate such information relatively easily provided that the information is expressed in terms of a probability distribution (density) $P(\theta)$ over the parameters θ. This density assigns high values to those parameters that we believe are likely *a priori*.

Now that we have a prior distribution $P(\theta)$, in addition to the distribution $P(\text{data}|\theta)$, we can talk about the joint distribution $P(\theta, \text{data})$ and more interestingly, about the conditional distribution $P(\theta|\text{data})$. The maximum a-posteriori (MAP) estimator is deffined as the value of the parameters θ that maximizes this conditional distribution:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta|\text{data})$$

12. Start from this definition and show that the MAP estimator is given by a maximization of the product of the prior belief and the likelihood:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta) p(\text{data}|\theta)$$

Hint: use Bayes' law, or the definition of conditional probability, and note that factors that are independent of θ can be ignored in the maximization.

Consider samples $x_1, \ldots, x_n$ from a Gaussian random variable with known variance σ^2 and unknown mean μ. We further assume a prior distribution (also Gaussian) over the mean, $\mu \sim \mathcal{N}(m, s^2)$, with fixed mean m and variance s^2.

13. Calculate the MAP estimate $\hat{\mu}_{MAP}$. Hint: as we did before, set the derivative of the logarithm to zero.

14. (*optional*) Show that as the number of samples increase, the prior knowledge becomes insignificant. That is, all MAP estimates assuming as a prior on μ any Gaussian distribution with non-zero variance, will converge to each other. What is the common estimator that all such MAP estimators converge to ? (Further note: This actually holds with rather mild assumptions about the prior— it need not be Gaussian).

15. (*optional*) What does the MAP estimator converge to if we increase the prior variance s^2?