

# 6.867 Machine learning and neural networks

## Problem set 1 — Solutions

September 25th, 2001

### 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/mllearn/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  $\{\mathbf{x}_t\}_{t=1,\dots,n}$ , and returns the predicted output values  $\{y_t\}_{t=1,\dots,n}$
  - 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.

**Answer:** First to read the data (ignoring column four):

```
>> data = load('housing.data');
>> x = data(:, [1:3 5:13]);
>> y = data(:, 14);
```

To get the training and test errors for training set of size  $s$ , we invoke the following MATLAB command:

```
>> [trainE, testE] = testLinear(x, y, s)
```

	<b>training size</b>	<b>training error</b>	<b>test error</b>
	10	$6.27 \times 10^{-26}$	$1.05 \times 10^5$
	50	3.437	24253
Here are the errors I got:	100	4.150	1328
	200	9.538	316.1
	300	9.661	381.6
	400	22.52	41.23

*Note that for a training size of ten, the training error should have been zero. The very low, but still non-zero, error is a result of limited precision of the calculations, and is not a problem. Furthermore, with only ten training examples, the optimal regression weights are not uniquely defined. There is a four dimensional linear subspace of weight vectors that all yield zero training error. The test error above (for a training size of ten) represents an*

arbitrary choice of weights from this subspace (implicitly made by the `pinv()` function). Using different, equally optimal, weights would yield different test errors.

**Scoring:** 15 points were awarded for questions 1+2. As indicated in the clarification emails, it was OK to either use, or not use, column four.

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

**Answer:** The training error will be zero if the input vectors are linearly independent. More precisely, since we are allowing an affine term  $w_0$ , it is enough that the input vectors with an additional term always equal to one, are linearly independent. Let  $X$  be the matrix of input vectors, with additional 'one' terms,  $y$  any output vector, and  $w$  a possible weight vector. If the inputs are linearly independent,  $Xw = y$  always has a solution, and the weights  $w$  lead to zero training error.

*Note that if  $X$  is a square matrix with linearly independent rows, than it is invertible, and  $Xw = y$  has a unique solution. But even if  $X$  is not square matrix, but its rows are still linearly independent (this can only happen if there are less rows than columns, i.e. less features than training examples), then there are solutions to  $Xw = y$ , which do not determine  $w$  uniquely, but still yield zero training error (as in the case of a sample size of ten above).*

**Scoring:** 5 points

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.

**Answer:** The training error tends to increase. As more examples have to be fitted, it becomes harder to 'hit', or even come close, to all of them.

The test error tends to decrease. As we take into account more examples when training, we have more information, and can come up with a model that better resembles the true behavior. More training examples lead to better generalization.

**Scoring:** 6 points. It was not enough to describe the behavior— it was necessary to explain the reason for this behavior.

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)

**Answer:** We will use the following functions, on top of those from question two:

```
function xx = degexpand(x,deg)
```

```
function [trainE, testE] = testPoly(x,y,numtrain,deg)
```

6. Prove that such scaling of features does not change the regression predictions. That is, given training feature vectors and output values  $\{\mathbf{x}_t, y_t\}_{t=1,\dots,n}$  and a test input vector  $\mathbf{x}_{\text{test}}$ , and scaling factors  $\{\alpha_i\}_{i=1,\dots,13}$ , we would like to prove that the prediction of the test output value would be the same if we trained a linear regression on  $\{\tilde{\mathbf{x}}_t, y_t\}_{t=1,\dots,n}$ , 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).

*We provide two alternate proofs:*

**Answer:** Let us consider the original feature  $x_t$  and the corresponding scaled version  $\hat{x}_t * \alpha_i$ . We can represent the relationship between the new and old matrices of inputs using a transformation matrix with only the scaling coefficients in its diagonal (**A**):

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{A}$$

Plugging this into the optimal weight equation (where  $X, y$  refer to the training set):

$$\begin{aligned}\hat{\mathbf{w}} &= (\hat{\mathbf{X}}^T \hat{\mathbf{X}})^{-1} \hat{\mathbf{X}}^T \mathbf{y} \\ &= ((\mathbf{X}\mathbf{A})^T \mathbf{X}\mathbf{A})^{-1} (\mathbf{X}\mathbf{A})^T \mathbf{y} \\ &= (\mathbf{A}^T \mathbf{X}^T \mathbf{X} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{X}^T \mathbf{y} \\ &= \mathbf{A}^{-1} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{A}^{T-1} \mathbf{A}^T \mathbf{X}^T \mathbf{y} \\ &= \mathbf{A}^{-1} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{A}^{-1} \mathbf{w}\end{aligned}$$

And so, the predicted output is:

$$\begin{aligned}\text{predicted } \hat{y}_{\text{test}} &= \hat{\mathbf{X}}_{\text{test}} \hat{\mathbf{w}} \\ &= \mathbf{X}_{\text{test}} \mathbf{A} \mathbf{A}^{-1} \mathbf{w} \\ &= \mathbf{X}_{\text{test}} \mathbf{w} \\ &= \text{predicted } y_{\text{test}}\end{aligned}$$

**Answer:** Note that since  $\tilde{\mathbf{x}}$  is a linear function of  $\mathbf{x}$ , the set of linear function of  $\mathbf{x}$  is exactly equal to the set of linear function of  $\tilde{\mathbf{x}}$ . The predictor, in both cases, is the function from this set that minimizes the training error. And so, both predictors will be the same.

**Scoring:** *5 points*

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

**Answer:** To get the training and test errors for maximum degree  $d$ , we invoke the following MATLAB command:

```
>> [trainE,testE] = testPoly(x,y,400,d)
```

	<b>degree</b>	<b>training error</b>	<b>test error</b>
	0	83.8070	102.2266
	1	22.5196	41.2285
	2	14.8128	32.8332
	3	12.9628	31.7880
Here are the errors I got:	4	10.8684	5262
	5	9.4376	5067
	6	7.2293	$4.8562 \times 10^7$
	7	6.7436	$1.5110 \times 10^6$
	8	5.9908	$3.0157 \times 10^9$
	9	5.4299	$7.8748 \times 10^{10}$
	10	4.3867	$5.2349 \times 10^{13}$

*These results were obtained using `pinv()`. Using different operations, although theoretically equivalent, might produce different results for higher degrees. In any case, using any of the suggested methods above, the errors should match the above table at least up to degree five. Beyond that, using `inv()` starts producing unreasonable results due to extremely small values in the matrix, which make it almost singular (non-invertible). If you used `inv()` and got such values, you should point this out.*

*Degree zero refers to having a constant predictor, i.e. predict the same input value for all output values. The constant value that minimizes the training error (and is thus used) is the mean training output.*

**Scoring:** 10 points were awarded for questions 5 and 7 together. Point deductions might be noted on either place– check the 'MATLAB' score on the first page.

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

**Answer:** Allowing more complex models, with more features, we can use as predictors functions that better correspond to the true behavior of the data. And so, the *approximation error* (the difference between the optimal model from our limited class, and the true behavior of the data) decreases as we increase the degree. As long as there is enough training data to support such complex models, the *generalization error* is not too bad, and the test error decreases. However, past some point we start *over-fitting* the training data and the increase in the generalization error becomes much more significant than

the continued decrease in the approximation error (which we cannot directly observe), causing the test error to rise.

Looking at the test error, the best maximum degree seems to be three.

**Scoring:** 6 points, one of which for the choice of degree

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 than or equal to the training error using a lower degree.

**Answer:** Predictors of lower maximum degree are included in the set of predictors of higher maximum degree (they correspond to predictors in which weights of higher degree features are set to zero). Since we choose the predictor from within the set that minimizes the training error, allowing more predictors, can only decrease the training error.

**Scoring:** 5 points. Discussing why this behavior makes sense, or why the error tends to decrease is not enough.

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?

**Answer:** We show for all  $m \geq n - 1$  (where  $n$  is the number of training examples), the training error is 0, by constructing weights which predict the training examples exactly. Let  $j$  be a component of the input with no repeated values. We let  $w_{i,d} = 0$  for all  $i \neq j$ , and all  $d = 1, \dots, m$ . Then we have

$$f(\mathbf{x}) = w_0 + \sum_i \sum_d w_{i,d} \mathbf{x}_i^d = w_0 + \sum_{d=1}^m w_{j,d} \mathbf{x}_j^d$$

Given  $n$  training points  $(x_1, y_1), \dots, (x_n, y_n)$  we are required to find  $w_0, w_{j,1}, \dots, w_{j,m}$  s.t.  $w_0 + \sum_{d=1}^m w_{j,d} (x_i)_j^d = y_i, \forall i = 1, \dots, n$ . That is, we want to interpolate  $n$  points with a degree  $m \geq n - 1$  polynomial, which can be done exactly as long as the points  $x_i$  are distinct.

**Scoring:** 3 points

## 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  $\mu$  and variance  $\sigma^2$ , 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  $\theta$  (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  $\theta$  when we have already observed the data.

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

**Answer:**

$$\begin{aligned} L(x_1, x_2, \dots, x_n; \mu, \sigma) &= \prod_{i=1}^n L(x_i; \mu, \sigma) \\ &= \left( \frac{1}{\sqrt{2\pi\sigma^2}} \right)^n e^{-\frac{\sum_{i=1}^n (x_i - \mu)^2}{2\sigma^2}} \end{aligned}$$

**Scoring:** 2 points

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

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

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:

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



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

**Answer:**

$$\frac{\partial \log L(x_1, \dots, x_n; \mu, \sigma)}{\partial \mu} = -\frac{1}{2\sigma^2} \sum_i (2\mu - 2x_i) = \frac{1}{\sigma^2} \sum_i x_i - \frac{n}{\sigma^2} \mu,$$

and equating (2) to zero we get, after multiplying by  $\sigma^2$ :

$$\hat{\mu} = \frac{1}{n} \sum_i x_i$$

**Scoring:** 4 points

*To be entirely rigorous, one should also make sure this is a maximum and not a minimum, e.g. by looking at the second derivative, and that the likelihood is bounded.*

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

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  $\mathbf{E}_{X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma)} [\hat{\mu}(X_1, \dots, X_n)] = \mu$  for all  $\mu, \sigma$ . The expectation here is over the possible choices of the random samples assuming they came from a Gaussian with mean  $\mu$  and variance  $\sigma^2$ .

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

**Answer:**

$$\begin{aligned} \mathbf{E}_{X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma)} [\hat{\mu}(X_1, \dots, X_n)] &= \mathbf{E} \left[ \frac{1}{n} \sum_i X_i \right] \\ &= \frac{1}{n} \mathbf{E} \left[ \sum_i X_i \right] = \frac{1}{n} \sum_i \mathbf{E}[X_i] \\ &= \frac{1}{n} \sum_i \mu = \mu \end{aligned}$$

thus,  $\hat{\mu}$  is an unbiased estimator of  $\mu$ .

**Scoring:** 4 points

We now proceed to calculate the maximum likelihood estimator for  $\sigma$ :

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

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  $\sigma$ . Note that in taking this derivative, we assume that  $\mu$  is set to its maximum likelihood value. However, we already know the value of  $\mu$  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  $\sigma^2$ , or its square root  $\sigma$ ?

**Answer:** No, it does not matter.  $\sigma^2$  is a monotonic function of  $\sigma$ , with strictly positive derivative, for  $\sigma > 0$  (which is the relevant range). Although the derivatives with respect to  $\sigma$  and to  $\sigma^2$  will be different, they will zero in the same places.

**Scoring:** 1 point

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

**Answer:** Taking the derivative of the likelihood with respect to  $\sigma^2$  (we could have also taken the derivative with respect to  $\sigma$ ):

$$\begin{aligned} \frac{\partial \log L}{\partial \sigma^2} &= -\frac{n}{2} \frac{1}{2\pi\sigma^2} 2\pi - \frac{1}{2} \left( \sum_{i=1}^n (x_i - \mu)^2 \right) \frac{\partial}{\partial \sigma^2} \sigma^{-2} \\ &= -\frac{n}{2\sigma^2} + \frac{(\sigma^2)^{-2}}{2} \left( \sum_{i=1}^n (x_i - \mu)^2 \right) \end{aligned}$$

Setting this to 0, we get:

$$\frac{1}{\hat{\sigma}^4} \sum_{i=1}^n (x_i - \mu)^2 = \frac{n}{\hat{\sigma}^2}$$

And since we must consider also the maximum with respect to  $\mu$ , we can plug in  $\hat{\mu}$  which we previously calculated, and get:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2$$

**Scoring:** 4 points

6. We would now like to show that  $\hat{\sigma}^2$  is not an unbiased estimator of  $\sigma^2$ . Calculate  $\mathbf{E}_{X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma)} [\hat{\sigma}^2(X_1, \dots, X_n)]$  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.

**Answer:** Because the variance of any random variable  $R$  is given by  $\text{var}(R) = E[R^2] - (E[R])^2$ , the expected value of the square of a Gaussian random variable  $X_i$  with mean  $\mu$  and variance  $\sigma^2$  is  $E[X_i^2] = \text{var}(X_i) + (E[X_i])^2 = \sigma^2 + \mu^2$ .

$$\begin{aligned}
E_{X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma)}[\hat{\sigma}^2(X_1, \dots, X_n)] &= E\left[\frac{1}{n} \sum_{i=1}^n \left(X_i - \frac{\sum_{j=1}^n X_j}{n}\right)^2\right] \\
&= \frac{1}{n} \sum_{i=1}^n E\left[\left(X_i - \frac{\sum_{j=1}^n X_j}{n}\right)^2\right] \\
&= \frac{1}{n} \sum_{i=1}^n E\left[\left(X_i - \frac{\sum_{j=1}^n X_j}{n}\right)\left(X_i - \frac{\sum_{j=1}^n X_j}{n}\right)\right] \\
&= \frac{1}{n} \sum_{i=1}^n E\left[X_i^2 - \frac{2}{n} X_i \sum_{j=1}^n X_j + \frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n X_j X_k\right] \\
&= \frac{1}{n} \sum_{i=1}^n E[X_i^2] - \frac{2}{n^2} \sum_{i=1}^n \sum_{j=1}^n E[X_i X_j] + \frac{1}{n^3} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n E[X_j X_k]
\end{aligned}$$

Consider the two summations  $\sum_{i=1}^n \sum_{j=1}^n E[X_i X_j]$  and  $\sum_{i=j}^n \sum_{k=1}^n E[X_j X_k]$ . Of the  $n^2$  terms in each of these summations,  $n$  of them satisfy  $i = j$  or  $j = k$ , so these terms are of the form  $E[X_i^2]$ . By linearity of expectation, these terms contribute  $nE[X_i^2]$  to the sum. The remaining  $n^2 - n$  terms are of the form  $E[X_i X_j]$  or  $E[X_j X_k]$  for  $i \neq j$  or  $j \neq k$ . Because the  $X_i$  are independent samples, it follows from linearity of expectation that these terms contribute  $(n^2 - n)E[X_i]E[X_j]$  to the summation.

$$\begin{aligned}
\sum_{i=1}^n \sum_{j=1}^n E[X_i X_j] &= \sum_{j=1}^n \sum_{k=1}^n E[X_j X_k] \\
&= nE[X_i^2] + (n^2 - n)E[X_i][X_j] \\
&= n(\sigma^2 + \mu^2) + (n^2 - n)\mu\mu = n\sigma^2 + n\mu^2 + n^2\mu^2 - n\mu^2 \\
&= n\sigma^2 + n^2\mu^2
\end{aligned}$$

$$\begin{aligned}
& E_{X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma)}[\hat{\sigma}^2(X_1, \dots, X_n)] \\
&= \frac{1}{n} \sum_{i=1}^n (\sigma^2 + \mu^2) - \frac{2}{n^2} (n\sigma^2 + n^2\mu^2) + \frac{1}{n^3} \sum_{i=1}^n (n\sigma^2 + n^2\mu^2) \\
&= \frac{1}{n} (n\sigma^2 + n\mu^2) - 2\frac{\sigma^2}{n} - 2\mu^2 + \frac{1}{n^3} (n^2\sigma^2 + n^3\mu^2) \\
&= \sigma^2 + \mu^2 - 2\frac{\sigma^2}{n} - 2\mu^2 + \frac{\sigma^2}{n} + \mu^2 \\
&= \sigma^2 - \frac{\sigma^2}{n} = \frac{n-1}{n}\sigma^2
\end{aligned}$$

Since the expected value of  $\hat{\sigma}^2(X_1, \dots, X_n)$  is not equal to the actual variance  $\sigma^2$ ,  $\hat{\sigma}^2$  is not an unbiased estimator. In fact, the maximum likelihood estimator tends to underestimate the variance. This is not surprising: consider the case of only a single sample: we will never detect any variance. If there are multiple samples, we will detect variance, but since our estimate for the mean will tend to be shifted from the true mean in the direction of our samples, we will tend to underestimate the variance.

**Scoring:** 4 points

7. Suggest an unbiased estimator  $\tilde{\sigma}^2(x_1, \dots, x_n)$  for  $\sigma^2$ , 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.

**Answer:** Consider the estimator:

$$\begin{aligned}
\tilde{\sigma}^2 &\doteq \frac{1}{n-1} \sum_i (x_i - \hat{\mu})^2 \\
&= \frac{n}{n-1} \hat{\sigma}^2
\end{aligned}$$

To verify that it is unbiased, we use the expectation of  $\hat{\sigma}^2$  that we derived above:

$$\mathbf{E}[\tilde{\sigma}^2] = \mathbf{E}\left[\frac{n}{n-1}\hat{\sigma}^2\right] = \frac{n}{n-1}\frac{n-1}{n}\sigma^2 = \sigma^2$$

*The unbiased estimator  $\tilde{\sigma}^2$  is frequently used instead of the maximum likelihood estimator. In fact, the default behavior of the MATLAB function `VAR(X)` is to return the unbiased estimator. The maximum likelihood estimator, which is also the variance of the sample, can be calculated using `VAR(X, 1)`.*

**Scoring:** 3 points

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

**Answer:** We can use the calculations in section five, but instead of taking the maximum over  $\mu$ , we just use the known, constant  $\mu$ . We get:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

In calculating the expected value of the estimator, note that we get exactly the definition of a variance of a random variable:

$$\begin{aligned} \mathbf{E}_{X_1, X_2, \dots, X_n \sim \mathcal{N}(\mu, \sigma)} [\hat{\sigma}^2(X_1, X_2, \dots, X_n)] &= \frac{1}{n} \sum_{i=1}^n \mathbf{E}_{X_i \sim \mathcal{N}(\mu, \sigma)} [(X_i - \mu)^2] \\ &= \frac{1}{n} \sum_{i=1}^n \text{Var}[\mathcal{N}(\mu, \sigma)] \\ &= \frac{1}{n} \sum_{i=1}^n \sigma^2 \\ &= \sigma^2 \end{aligned}$$

And the estimator is unbiased.

**Scoring:** 4 points

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:  $\check{\mu}(x_1, \dots, x_n) = x_1$ . Show that this is an unbiased estimator of  $\mu$ .

**Answer:**  $E_{X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma)} [\check{\mu}(X_1, \dots, X_n)] = E[X_1] = \mu$

**Scoring:** 2 points

One reason that  $\check{\mu}$  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  $\mu$ .

An estimator  $\hat{\theta}$  is (mean squared) *consistent* if it converges to  $\theta$  in the following sense:  $\mathbf{E}_{X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma)} [(\hat{\theta}(X_1, \dots, X_n) - \theta)^2] \rightarrow 0$  as  $n \rightarrow \infty$ . 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  $\theta$ .

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

**Answer:**

$$\mathbf{E} [(\hat{\mu} - \mu)^2] = \mathbf{E} \left[ \left( \frac{\sum_i X_i}{n} - \mu \right)^2 \right] = \frac{1}{n^2} \mathbf{E} \left[ \left( \sum_i X_i - \mathbf{E} \left[ \sum_i X_i \right] \right)^2 \right]$$

noticing that this is the definition of the variance:

$$\begin{aligned} &= \frac{1}{n^2} \mathbf{Var} \left[ \sum_i X_i \right] \\ &= \frac{1}{n^2} n \mathbf{Var} [X_i] = \frac{1}{n^2} n \sigma^2 \\ &= \frac{\sigma^2}{n} \xrightarrow{n \rightarrow \infty} 0 \end{aligned}$$

**Scoring:** *Optional question– no score recorded*

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

**Answer:** They are both consistent estimators. It is easy to see that as  $n$  becomes large, the two estimators converge to each other (for this reason, for large  $n$ , the maximum likelihood estimator is almost unbiased). A calculation similar to the above calculation for  $\hat{\mu}$ , though somewhat more involved, shows that they are both consistent.

**Scoring:** *No score recorded*

### 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  $\theta$ . 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 defined as the value of the parameters  $\theta$  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  $\theta$  can be ignored in the maximization.

**Answer:** By the definition of conditional probability, we have  $p(\theta|\text{data}) = \frac{p(\theta,\text{data})}{p(\text{data})}$ . Conditioning on the value of  $\theta$ , we obtain  $p(\theta, \text{data}) = p(\text{data}|\theta)p(\theta)$ . This yields the following.

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

Because the probability  $p(\text{data})$  is independent of  $\theta$ , it acts as a constant with respect to the maximization over  $\theta$ , and so the value of  $\theta$  that maximizes  $\frac{p(\text{data}|\theta)p(\theta)}{p(\text{data})}$  is the same value that maximizes  $p(\theta)p(\text{data}|\theta)$ .

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

**Scoring:** 2 points

Consider samples  $x_1, \dots, x_n$  from a Gaussian random variable with known variance  $\sigma^2$  and unknown mean  $\mu$ . 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.

**Answer:** The prior distribution over the mean is  $p(\mu) = (2\pi s^2)^{-1/2} e^{-\frac{(\mu-m)^2}{2s^2}}$ . Since the samples  $x_i$  are taken to be independent, we have:

$$p(\text{data}|\mu) = e^{-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2}$$

And combining them:

$$\log(p(\mu)p(\text{data}|\mu)) = -\frac{1}{2} \log(2\pi s^2) - \frac{(\mu - m)^2}{2s^2} - \frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$$

Taking the derivative with respect to  $\mu$ :

$$\begin{aligned} \frac{\partial \log(p(\mu)p(\text{data}|\mu))}{\partial \mu} &= 0 - \frac{2(\mu - m)(1)}{2s^2} - 0 - \frac{1}{2\sigma^2} \sum_{i=1}^n (2)(x_i - \mu)(-1) \\ &= -\frac{\mu - m}{s^2} + \frac{1}{\sigma^2} \sum_{i=1}^n (x_i - \mu) = -\frac{\mu}{s^2} + \frac{m}{s^2} + \frac{1}{\sigma^2} \sum_{i=1}^n x_i - \frac{n\mu}{\sigma^2} \end{aligned}$$

Setting this derivative to zero in order to find the maximum:

$$\begin{aligned}
 0 &= -\frac{\hat{\mu}_{MAP}}{s^2} + \frac{m}{s^2} + \frac{1}{\sigma^2} \sum_{i=1}^n x_i - \frac{n\hat{\mu}_{MAP}}{\sigma^2} \\
 \hat{\mu}_{MAP} \left( \frac{n}{\sigma^2} + \frac{1}{s^2} \right) &= \frac{1}{\sigma^2} \sum_{i=1}^n x_i + \frac{m}{s^2} \\
 \hat{\mu}_{MAP} &= \frac{\frac{1}{\sigma^2} \sum_{i=1}^n x_i + \frac{m}{s^2}}{\frac{n}{\sigma^2} + \frac{1}{s^2}} = \frac{s^2 \sum_{i=1}^n x_i + m\sigma^2}{ns^2 + \sigma^2} \\
 &= \left( \frac{ns^2}{ns^2 + \sigma^2} \right) \frac{\sum_i x_i}{n} + \left( \frac{\sigma^2}{ns^2 + \sigma^2} \right) m
 \end{aligned}$$

**Scoring:** 5 points

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  $\mu$  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).

**Answer:** Notice that as  $n$  increases, while  $s, m$  and  $\sigma$  remain constant, we have  $\frac{\sigma^2}{ns^2 + \sigma^2} \rightarrow 0$  while  $\frac{ns^2}{ns^2 + \sigma^2} \rightarrow 1$ , yielding  $\hat{\mu}_{MAP} \rightarrow \frac{\sum_i x_i}{n} = \hat{\mu}$ . That is, when there are many samples, the prior knowledge becomes less and less relevant, and all MAP estimators (for any prior) converge to the maximum likelihood estimator  $\hat{\mu}$ . This is not surprising: as we have more data, it outweighs our prior speculations.

This also tells us that these MAP estimators are consistent, since we already know that  $\hat{\mu}$  is consistent. However, they are, of course, biased— they are biased towards our prior guess  $m$ .

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

**Answer:** As the prior variance  $s^2$  increases, even for a small number of samples  $n$ , we have  $\frac{\sigma^2}{ns^2 + \sigma^2} \rightarrow 0$  while  $\frac{ns^2}{ns^2 + \sigma^2} \rightarrow 1$ , again yielding  $\hat{\mu}_{MAP} \rightarrow \frac{\sum_i x_i}{n} = \hat{\mu}$ . That is, when we have an uninformative, almost uniform, prior, we are left only with our data to base our estimation on.

On the other hand, if the prior variance is very small,  $s^2 \rightarrow 0$ , then we have  $\frac{\sigma^2}{ns^2 + \sigma^2} \rightarrow 1$  while  $\frac{ns^2}{ns^2 + \sigma^2} \rightarrow 0$ , yielding  $\hat{\mu}_{MAP} \rightarrow m$ . That is, if our prior is very concentrated, we essentially already know the answer, and can ignore the data.

Thanks to Rui Fan, Jonathan Herzog, Ray Jones, Damon Mosk-Aoyama, Luis Perez-Breva and Gregory Shakhnarovich for making available their typeset solutions.