Homework 4, due Wednesday April 15

COMS 4771 Spring 2015

Problem 1 (Anatomy of ordinary least squares)**.** Let $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ be a random vector in $\mathbb{R}^p$, and let Y be a real-valued random variable. Let $\mathbb{E}$ denote the expectation operator with respect to the joint distribution of $(\boldsymbol{X}, Y)$. (Remember, we treat vectors in $\mathbb{R}^p$ as $p \times 1$ column vectors, so $\mathbb{E}(\boldsymbol{X}\boldsymbol{X}^\top)$ is a $p \times p$ matrix.) Also, let $\boldsymbol{X}_{(-i)} = (X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_p)$ be the random vector in $\mathbb{R}^{p-1}$ that is the same as $\boldsymbol{X}$ except with the i-th random variable X_i omitted.

(a) Determine a linear function of $\boldsymbol{X}$—call it $\hat{Y} := \langle \boldsymbol{w}, \boldsymbol{X} \rangle$—for which

$$\mathbb{E}\Big[(Y - \hat{Y})^2\Big]$$

is as small as possible. You may assume that $\mathbb{E}(\boldsymbol{X}\boldsymbol{X}^\top)$ is invertible.

Hint: determine the vector $\boldsymbol{w} \in \mathbb{R}^p$ using the same method used to derive ordinary least squares.

(b) For the $\hat{Y}$ from part (a), what is $\mathbb{E}((Y - \hat{Y})\boldsymbol{X})$? Show the derivation of your answer.

(c) For each $i \in \{1, 2, \ldots, p\}$, define

$$Z_i := X_i - \mathbb{E}(X_i \boldsymbol{X}_{(-i)})^\top \mathbb{E}(\boldsymbol{X}_{(-i)} \boldsymbol{X}_{(-i)}^\top)^{-1} \boldsymbol{X}_{(-i)},$$

which is the residual of the best approximation of X_i as a linear function of $\boldsymbol{X}_{(-i)}$. (We assume that $\mathbb{E}(\boldsymbol{X}_{(-i)} \boldsymbol{X}_{(-i)}^\top)$ is invertible.)

What is $\mathbb{E}(Z_i \boldsymbol{X}_{(-i)})$? Show the derivation of your answer.

Hint: this is similar to part (b).

(d) Show that, for each $i \in \{1, 2, \ldots, p\}$,

$$\mathbb{E}(Z_i^2) w_i = \mathbb{E}(Y Z_i), \qquad (\star)$$

where $\boldsymbol{w} = (w_1, w_2, \ldots, w_p)$ is the vector determining $\hat{Y}$ from part (a).

Hint: First, show that $\mathbb{E}(Z_i^2) = \mathbb{E}(X_i Z_i)$ (using the result from part (c)). Then, show that $\mathbb{E}(\langle \boldsymbol{w}, \boldsymbol{X} \rangle Z_i) = w_i \mathbb{E}(X_i Z_i)$ (again, using the result from part (c)), and $\mathbb{E}((Y - \hat{Y}) Z_i) = 0$ (using the result from part (b)). Use the fact that $Y Z_i = \langle \boldsymbol{w}, \boldsymbol{X} \rangle Z_i + (Y - \hat{Y}) Z_i$ and linearity of expectation to conclude $(\star)$.

Postscript: We think of Z_i as what's leftover from X_i after subtracting out its best linear prediction in terms of $\boldsymbol{X}_{(-i)}$. Eq. $(\star)$ shows that if you are just interested in obtaining w_i (which generally depends on the distribution of all of $\boldsymbol{X}$), then it can be obtained just from the distribution of Z_i (and Y, of course); the information one could get from the rest of $\boldsymbol{X}$ is not necessary.

Problem 2 (Image quantization using k-means)**.** Download the "reshaped" OCR image data set `mnist_patches.mat` from Courseworks, and load it into MATLAB. This is exactly the same as the original data set, except (i) labels are omitted, (ii) each data point has been reshaped to be a 28×28 matrix (e.g., the i-th training image is `data(:,:,i)`), and (iii) all (non-overlapping) 4×4 patches (as vectors in $\mathbb{R}^{16}$) from all training images have been extracted into a matrix `patches` (and the same was done for test images into `testpatches`).

(In this problem, you won't actually have to use thie original training `data`, just the training `patches`, the `testdata`, and the `testpatches`.)

In this problem, you will cluster the image patches from the training data in order to *quantize* the images. That is, you'll use clustering to find a set of *representative* image patches so that you can represent each image in a more compact ("quantized") form.

(a) Note that the number of patches is quite large, so using several iterations of Lloyd's algorithm may be a burden since each iteration requires two passes through the data. For this problem, we'll use a different algorithm that just requires a single pass through the data.

Implement the following "streaming" algorithm for k-means clustering.

input: stream of points from $\mathbb{R}^d$, positive integer $k \in \mathbb{N}$.
  Let $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k \in \mathbb{R}^d$ be the first k points from the stream.
  Let $n_1 = n_2 = \cdots = n_k = 1$.
  for each new point $\boldsymbol{x} \in \mathbb{R}^d$ from the stream **do**
    Let $j := \arg\min_{j' \in \{1,2,\ldots,k\}} \|\boldsymbol{x} - \boldsymbol{c}_{j'}\|_2^2$.
    Let $n_j := n_j + 1$.
    Let $\boldsymbol{c}_j := \left(1 - \frac{1}{n_j}\right)\boldsymbol{c}_j + \frac{1}{n_j}\boldsymbol{x}$.
  end for
  return $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k$.

Though in principal it is possible to simply take as input a (possibly) endless stream of points, your function can just take as input a matrix of data points (as well as k).

```
function centers = streaming_kmeans(patches,k)
```

Use this algorithm to obtain k representatives for the training image patches, for $k \in \{8, 16, 32, 64\}$. (The order of the patches has already been randomly permuted.)

You can quantize the training or test images using the function `quantize_image` (available on Courseworks): for example,

```
qimage = quantize_image(testdata(:,:,24),centers);
```

(Here, `centers` should be a $k \times 16$ matrix containing the k representatives you get from the streaming k-means algorithm, arranged as rows in this matrix.) The result is a 7×7 matrix that simply points to which representative is used to represent each $4\times$ patch of the image.

To see how this quantized image looks, you have to transform this back to a 28×28, using the function `decode_qimage` (also on Courseworks):

```
image = decode_qimage(qimage,centers);
```

You can view this using `imagesc` side-by-side with the original images:

```
colormap(1-gray);
image24 = decode_qimage(quantize_image(testdata(:,:,24),centers),centers);
imagesc([testdata(:,:,24), image24]);
```

Include these side-by-side comparisons for test images 24, 100, and 5000 (for $k \in \{8, 64\}$) in your report.

(b) Determine (as a function of k) the total number of bits b_k required to represent the both the k cluster representatives as well as the quantization of all 10000 test images. You may assume that each cluster representative is represented as a vector of 16 double-precision floating point numbers (and each double is 64 bits). Explain your derivation.

(c) Let $\text{cost}_k := \sum_{\boldsymbol{x} \in \text{test patches}} \min_{j \in [k]} \|\boldsymbol{x} - \boldsymbol{c}_j\|_2^2$ for the k cluster representatives $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k$ you obtained in part (a). Plot cost_k against $\log_2(k)$ for $k \in \{8, 16, 32, 64\}$.

Problem 3 (Loss functions)**.** Recall $\ell_{\text{sq}}(z) = (1 - z)^2$, $\ell_{\log}(z) = \ln(1 + \exp(-z))$, and $\ell_{\exp}(z) = \exp(-z)$. Let Y be a random variable taking values in $\{-1, +1\}$, and let $\eta := \Pr[Y = +1]$ (assume it is neither 0 nor 1).

(a) Show that $\mathbb{E}[\ell_{\text{sq}}(Y\hat{y})]$ is minimized by $\hat{y} = 2\eta - 1$.

(b) Show that $\mathbb{E}[\ell_{\log}(Y\hat{y})]$ is minimized by $\hat{y} = \ln\left(\frac{\eta}{1-\eta}\right)$.

(c) Show that $\mathbb{E}[\ell_{\exp}(Y\hat{y})]$ is minimized by $\hat{y} = \frac{1}{2}\ln\left(\frac{\eta}{1-\eta}\right)$.

You may assume that each of the loss functions ℓ_{sq}, $\ell_{\log}$, and $\ell_{\exp}$ is a convex function of $\hat{y}$.

for those lose functions. the bigger the z, the large the lsq(z). Thus to minimize E(Lsq(Yy)) is to maximize z. (improve accuracy)