Name: **Student ID:**

CS 189: Introduction to Machine Learning

Homework 1 Solutions

Due: September 13, 2016 at 11:59pm

Instructions

- This homework includes both a written portion and a coding portion.
- We prefer that you typeset your answers using LaTeX. Neatly handwritten and scanned solutions will also be accepted. Make sure to start each question on a new page.
- You will be submitting **two** things to Gradescope:
  - Append a screenshot or LaTeX snippet of your code to the last page of your writeup. Submit **a PDF of your writeup** to the Homework 1 assignment on Gradescope.
  - Zip up your source code and submit that **zip file** to the Homework 1 Code assignment on Gradescope.
- You should be able to see CS 189/289A on Gradescope when you log in with your bCourses email address. Please make a Piazza post if you have any problems accessing Gradescope.
- The assignment covers concepts in probability, linear algebra, matrix calculus, and decision theory.
- **Start early. This is a long assignment. Some of the material may not have been covered in lecture; you are responsible for finding resources to understand it.**

Problem 1: Expected Value.

A target is made of 3 concentric circles of radii $1/\sqrt{3}$, 1 and $\sqrt{3}$ feet. Shots within the inner circle are given 4 points, shots within the next ring are given 3 points, and shots within the third ring are given 2 points. Shots outside the target are given 0 points.

Let X be the distance of the hit from the center (in feet), and let the probability density function of X be

$$f(x) = \begin{cases} \frac{2}{\pi(1+x^2)} & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

What is the expected value of the score of a single shot? Tip: integration is hard, use Wolfram Alpha.

Solution: The expected value is

$$\int_0^{1/\sqrt{3}} 4\frac{2}{\pi(1+x^2)}dx + \int_{1/\sqrt{3}}^{1} 3\frac{2}{\pi(1+x^2)}dx + \int_1^{\sqrt{3}} 2\frac{2}{\pi(1+x^2)}dx$$
$$= \frac{2}{\pi}\left[4\left(\tan^{-1}\frac{1}{\sqrt{3}} - \tan^{-1} 0\right) + 3\left(\tan^{-1} 1 - \tan^{-1}\frac{1}{\sqrt{3}}\right) + 2\left(\tan^{-1}\sqrt{3} - \tan^{-1} 1\right)\right]$$
$$= \boxed{\frac{13}{6}}$$

Problem 2: MLE.

Assume that the random variable X has the exponential distribution

$$f(x;\theta) = \theta e^{-\theta x} \qquad x \geq 0, \theta > 0$$

where θ is the parameter of the distribution. Show how to use the method of maximum likelihood to estimate θ from n observations of X: $x_1, \ldots, x_n$.

Solution: We'll solve the general case for the MLE of an exponential distribution:

$$\begin{aligned}
\mathcal{L}(x_1, x_2, ..., x_n; \theta) &= \prod_{i=1}^{n} f(x_i;\theta) \\
&= \prod_{i=1}^{n} \theta e^{-\theta x_i} \\
&= \theta^n \exp\left(-\theta \sum_{i=1}^{n} x_i\right)
\end{aligned}$$

Finding the log-likelihood:

$$\ell(x_1, x_2, ..., x_n; \theta) = n \log\theta - \theta \sum_{i=1}^{n} x_i$$

Taking the derivative with respect to θ:

$$\begin{aligned}
\frac{\partial \ell}{\partial \theta} &= \frac{n}{\theta} - \sum_{i=1}^{n} x_i = 0 \\
\hat{\theta} &= \frac{n}{\sum_{i=1}^{n} x_i}
\end{aligned}$$

Definition. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix. We say that A is **positive definite** if $\forall x \in \mathbb{R}^n \mid x \neq \vec{0},\ x^\top Ax > 0$. Similarly, we say that A is **positive semidefinite** if $\forall x \in \mathbb{R}^n,\ x^\top Ax \geq 0$.

Problem 3: Positive Definiteness.

Let $x = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top \in \mathbb{R}^n$, and let $A \in \mathbb{R}^{n \times n}$ be the square matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

(a) Give an explicit formula for $x^\top Ax$. Write your answer as a sum involving the elements of A and x.

(b) Show that if A is positive definite, then the entries on the diagonal of A are positive (that is, $a_{ii} > 0$ for all $1 \leq i \leq n$).

Solution:

(a)

$$x^\top Ax = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

(b) Let $i \in [1, n]$, and let e_i be the i^{th} standard basis vector (that is, the vector of all zeros except for a single 1 in the i^{th} position). Since A is positive definite, we have $e_i^\top Ae_i = a_{ii} > 0$.

Problem 4: Short Proofs.

A is symmetric in all parts.

(a) Let A be a positive semidefinite matrix. Show that $A + \gamma I$ is positive definite for any $\gamma > 0$.

(b) Let A be a positive definite matrix. Prove that all eigenvalues of A are greater than zero.

(c) Let A be a positive definite matrix. Prove that A is invertible. (Hint: Use the previous part.)

(d) Let A be a positive definite matrix. Prove that there exist n linearly independent vectors $x_1, x_2, ..., x_n$ such that $A_{ij} = x_i^\top x_j$. (Hint: Use the spectral theorem and what you proved in (b) to find a matrix B such that $A = B^\top B$.)

Solution:

(a) Let $x \neq 0$. Then

$$\begin{aligned} x^\top (A + \gamma I)x &= x^\top Ax + x^\top \gamma I x \\ &= x^\top Ax + \gamma \|x\|^2 \\ &> 0 \end{aligned}$$

because $x^\top Ax \geq 0$ (since A is positive semidefinite) and $\|x\|^2 > 0$ (because $x \neq 0$). Hence $A + \gamma I$ is positive definite.

(b) We know $x^\top Ax > 0$ for any x. Consider v to be any eigenvector with $Av = \lambda v$. Then $v^\top Av = \lambda v^\top v > 0$. Since $v^\top v > 0$ (by definition, v is non-zero), we must have $\lambda > 0$.

(c) Since A is positive definite, all eigenvalues are positive. But then if A is not invertible, 0 is an eigenvalue, which is a contradiction. Thus A must be invertible.

(d) Because A is symmetric positive definite, we diagonalize to obtain $A = P^\top DP$ with orthogonal P and diagonal matrix D with eigenvalues on the diagonal. Since all eigenvalues are positive, we can define $E = D^{1/2}$. Then $A = P^\top EEP = P^\top E^\top EP = (EP)^\top EP$. We thus define $x_1, x_2, ..., x_n$ as the columns of the matrix EP, which are linearly independent since P is orthogonal. As we desired, $A_{ij} = x_i^\top x_j$.

Problem 5: Derivatives and Norm Inequalities.

Derive the expression for following questions. Do not write the answers directly.

(a) Let $\mathbf{x}, \mathbf{a} \in \mathbb{R}^n$. Compute $\frac{\partial(\mathbf{x}^T\mathbf{a})}{\partial \mathbf{x}}$.

(b) Let $\mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{x} \in \mathbb{R}^n$. Compute $\frac{\partial(\mathbf{x}^T\mathbf{A}\mathbf{x})}{\partial \mathbf{x}}$.

(c) Let $\mathbf{A}, \mathbf{X} \in \mathbb{R}^{n\times n}$. Compute $\frac{\partial \text{Trace}(\mathbf{X}\mathbf{A})}{\partial \mathbf{X}}$.

(d) Let $\mathbf{x} \in \mathbb{R}^n$. Prove that $\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{n}\|\mathbf{x}\|_2$. (Hint: The Cauchy-Schwarz inequality may come in handy.)

(e) Write down a simple expression for $g(x) = \sup_{\|z\|_1 \leq 1} x^T z$. Hint: first prove an upper bound on $g(x)$, then propose a choice of z that achieves the bound.

Solution:

(a) Let $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$ and $\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix}$.

$$\mathbf{x}^T\mathbf{a} = \sum_{i=1}^{n} x_i a_i$$

Taking partial derivative wrt a component, we get

$$\frac{\partial\left(\mathbf{x}^T\mathbf{a}\right)}{\partial x_k} = a_k$$

Placing all partial derivatives into a single vector, we get

$$\frac{\partial\left(\mathbf{x}^T\mathbf{a}\right)}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial(\mathbf{x}^T\mathbf{a})}{\partial x_1} \\ \frac{\partial(\mathbf{x}^T\mathbf{a})}{\partial x_2} \\ \dots \\ \frac{\partial(\mathbf{x}^T\mathbf{a})}{\partial x_n} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix} = \mathbf{a}$$

(b) Let $\mathbf{A} = \left[a_{ij}\right]_{n\times n}$. We can write

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_{i,j=1}^{n} a_{ij} x_i x_j$$

Taking partial derivative wrt a component, we get

$$\begin{aligned}
\frac{\partial\left(\mathbf{x}^T\mathbf{A}\mathbf{x}\right)}{\partial x_k} &= \frac{\partial}{\partial x_k}\left(\sum_{i,j=1}^{n} a_{ij}x_ix_j\right) \\
&= \frac{\partial}{\partial x_k}\left(x_1\sum_{j=1}^{n} a_{1j}x_j + x_2\sum_{j=1}^{n} a_{2j}x_j + \cdots + x_k\sum_{j=1}^{n} a_{kj}x_j + \cdots + x_n\sum_{j=1}^{n} a_{nj}x_j\right) \\
&\text{(Use product rule of differentiation, i.e. } (fg)' = f'g + fg'\text{), on each term)} \\
&= x_1a_{1k} + x_2a_{2k} + \cdots + x_ka_{kk} + \sum_{j=1}^{n} a_{kj}x_j + \cdots + x_na_{nk} \\
&= (x_1a_{1k} + x_2a_{2k} + \cdots + x_ka_{kk} + \cdots + x_na_{nk}) + \left(\sum_{j=1}^{n} a_{kj}x_j\right) \\
&= \left(\sum_{i=1}^{n} a_{ik}x_i\right) + \left(\sum_{j=1}^{n} a_{kj}x_j\right) \\
&= \left(k^{th} \text{ column of } \mathbf{A}\right)^T \mathbf{x} + \left(k^{th} \text{ row of } \mathbf{A}\right)^T \mathbf{x}
\end{aligned}$$

Placing all partial derivatives into a single vector, we get

$$\frac{\partial\left(\mathbf{x}^T\mathbf{A}\mathbf{x}\right)}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}$$

(c) Let $\mathbf{A} = [a_{ij}]_{n\times n}$ and $\mathbf{X} = [x_{ij}]_{n\times n}$. We can write

$$\text{Trace}(\mathbf{XA}) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}a_{ji}$$

Taking partial derivative wrt a component, we get

$$\begin{aligned}
\frac{\partial \text{Trace}(\mathbf{XA})}{\partial x_{ij}} &= \frac{\partial}{\partial x_{ij}}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}a_{ji}\right) \\
&= a_{ji}
\end{aligned}$$

Placing all partial derivatives into the matrix, we get

$$\frac{\partial \text{Trace}(\mathbf{XA})}{\partial \mathbf{X}} = [a_{ji}]_{n\times n} = \mathbf{A}^T$$

(d) First let's prove the left-hand side inequality as follows:

$$\begin{aligned}
\|\mathbf{x}\|_2 &= \sqrt{\sum_{i=1}^{n} x_i^2} \\
&= \sqrt{x_1^2 + x_2^2 \cdots + x_n^2} \\
&\text{(adding positive terms)} \\
&\leq \sqrt{x_1^2 + x_2^2 \cdots + x_n^2 + 2\Bigg(\sum_{1 \leq i < j \leq n} |x_i||x_j|\Bigg)} \\
&= \sqrt{(|x_1| + |x_2| + \cdots + |x_n|)^2} \\
&= |x_1| + |x_2| + \cdots + |x_n| \\
&= \|\mathbf{x}\|_1 \\
\implies \|\mathbf{x}\|_2 &\leq \|\mathbf{x}\|_1
\end{aligned}$$

Let's now prove the right-hand side inequality as follows:

$$\begin{aligned}
& \|\mathbf{x}\|_1 = |x_1| + |x_2| + \cdots + |x_n| \\
\implies & \|\mathbf{x}\|_1 = \underbrace{(|x_1|, |x_2|, \ldots, |x_n|)^T}_{\text{call this vector } \mathbf{x}'} \bullet (1, 1, \ldots 1) \\
\implies & \|\mathbf{x}\|_1 = \mathbf{x}'^T \bullet \mathbf{1} \\
& \qquad \text{(Using Cauchy}-\text{Schwarz inequality on the right)} \\
\implies & \|\mathbf{x}\|_1 \leq \|\mathbf{x}'\|_2 \|\mathbf{1}\|_2 \\
& \qquad \text{Note: } \|\mathbf{x}'\|_2 = \|\mathbf{x}\|_2 \text{ and } \|\mathbf{1}\|_2 = \sqrt{n} \\
\implies & \|\mathbf{x}\|_1 \leq \sqrt{n}\|\mathbf{x}\|_2
\end{aligned}$$

Thus, we have shown

$$\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{n}\|\mathbf{x}\|_2$$

(e) $g(z) = \sum_i x_i z_i \leq \sum_i |x_i||z_i| \leq \sum_i |z_i| \max_j |x_j| = \|z\|_1 \max_j |x_j| \leq \max_j |x_j|$. Let $j_\star = \arg\max_j |x_j|$. Let $z = \mathrm{sgn}(x_{j_\star}) e_{j_\star}$, then z achieves the supremum. So, $g(x) = \max_j |x_j|$.

Problem 6: Gaussian classification.

Let $P(x \mid \omega_i) \sim \mathcal{N}(\mu_i, \sigma^2)$ for a two-category, one-dimensional classification problem with $P(\omega_1) = P(\omega_2) = 1/2$. Here, the classes are ω_1 and ω_2. For this problem, we have $\mu_2 \geq \mu_1$.

(a) Find the optimal Bayes decision boundary (i.e., find x such that $P(\omega_1 \mid x) = P(\omega_2 \mid x)$). What is the corresponding decision rule?

(b) Show that the Bayes error associated with this decision rule is

$$P_e = \frac{1}{\sqrt{2\pi}} \int_a^\infty e^{-z^2/2} dz$$

where $a = \dfrac{\mu_2 - \mu_1}{2\sigma}$. The Bayes error is the probability of misclassification:

$$P_e = P((\text{misclassified as } \omega_1) \mid \omega_2) P(\omega_2) + P((\text{misclassified as } \omega_2) \mid \omega_1) P(\omega_1).$$

Solution:

(a) $P(\omega_1 \mid x) = P(\omega_2 \mid x) \rightarrow P(x \mid \omega_1)P(\omega_1) = P(x \mid \omega_2)P(\omega_2) \rightarrow P(x \mid \omega_1) = P(x \mid \omega_2) \rightarrow \mathcal{N}(\mu_1, \sigma^2) = \mathcal{N}(\mu_2, \sigma^2) \rightarrow (x - \mu_1)^2 = (x - \mu_2)^2 \rightarrow x = \dfrac{\mu_1 + \mu_2}{2}$. The decision rule is to select ω_1 if $x < \dfrac{\mu_1 + \mu_2}{2}$, and ω_2 otherwise.

(b)

$$P_e = \frac{1}{2} \int_{-\infty}^{(\mu_1+\mu_2)/2} \mathcal{N}(\mu_2, \sigma^2)\ du + \frac{1}{2} \int_{(\mu_1+\mu_2)/2}^{\infty} \mathcal{N}(\mu_1, \sigma^2)\ du$$

We normalize each of these to obtain:

$$P_e = \frac{1}{2} P(\mathcal{N}(0,1) \leq \frac{\mu_1 - \mu_2}{2\sigma}) + \frac{1}{2} P(\mathcal{N}(0,1) \geq \frac{\mu_2 - \mu_1}{2\sigma})$$

$$= P(\mathcal{N}(0,1) \geq \frac{\mu_2 - \mu_1}{2\sigma})$$

Finally, we plug in the PDF of the standard normal to observe that $P_e = P(\mathcal{N}(0,1) \geq \dfrac{\mu_2 - \mu_1}{2\sigma}) = \dfrac{1}{\sqrt{2\pi}} \displaystyle\int_a^\infty e^{-z^2/2} dz$, where $a = \dfrac{\mu_2 - \mu_1}{2\sigma}$.

Problem 7: Regularized Least Squares.

In this question we'll revisit regularized least squares. Let $x_1, \dots, x_n \in \mathbf{R}^d$, $y_1, \dots, y_n \in \mathbf{R}$ be the training dataset. Let $X \in \mathbf{R}^{(n,d)}$ be the corresponding data matrix. The ℓ_2-regularized least square estimate for w is the solution to the following optimization problem:

$$\underset{w}{\text{minimize}} \, \frac{1}{2}\|Xw - y\|_2^2 + \frac{\lambda}{2}\|w\|_2^2 \tag{1}$$

Here $\lambda > 0$ is the regularization parameter.

(a) Compute the gradient of the objective function in (1) with respect to w.

(b) Set the gradient to zero to get a closed-form solution for w.

(c) Recall that any vector $w \in \mathbf{R}^d$ can be written as $w = w_n + X^T\alpha$ for some w_n in the nullspace of X (i.e. $Xw_n = 0$) and some $\alpha \in \mathbf{R}^n$. Furthermore, recall that w_n is perpendicular to $X^T\alpha$ for any α. Using this decomposition of w, show that the first term in the objective function of (1) depends only on α, and does not depend on w_n.

(d) Prove that the second term of (1) *does* depend on w_n, but is minimized (over w_n) when $w_n = 0$. Hint: remember that w_n is orthogonal to $X^T\alpha$.

(e) Conclude that $w_\star = X^T\alpha_\star$ for some $\alpha_\star$, and rewrite (1) as an optimization problem over α.

(f) Write down a simple, closed-form solution for $\alpha_\star$. Try to make this as simple as possible.

(g) Compare (f) and (b); computationally, when might you want to find $\alpha_\star$ instead of $w_\star$?

Solution:

(a) $\nabla_w \frac{1}{2}\|Xw - y\|_2^2 + \frac{\lambda}{2}\|w\|_2^2 = X^T(Xw - y) + \lambda w$.

(b) $w_\star = (X^TX + \lambda I)^{-1}X^Ty$.

(c) $\frac{1}{2}\|Xw - y\|_2^2 = \frac{1}{2}\|X(w_n + X^T\alpha) - y\|_2^2 = \frac{1}{2}\|XX^T\alpha - y\|_2^2$.

(d) The Pythagorean theorem implies $\|w_n + A^T\alpha\|_2^2 = \|A^T\alpha\|_2^2 + \|w_n\|_2^2$, which is minimized (over w_n) by $w_n = 0$.

(e) The objective function is $L = \frac{1}{2}\|XX^T\alpha - y\|_2^2 + \frac{\lambda}{2}\|X^T\alpha\|_2^2$.

(f)

$$
\begin{aligned}
L &= \frac{1}{2}(XX^T\alpha - y)^T(XX^T\alpha - y) + \frac{\lambda}{2}(X^T\alpha)^T(X^T\alpha) \\
&= \frac{1}{2}(\alpha^T XX^T XX^T\alpha - y^T XX^T\alpha - \alpha^T XX^T y + y^T y) + \frac{\lambda}{2}(\alpha^T XX^T\alpha) \\
&= \frac{1}{2}(\alpha^T XX^T XX^T\alpha - 2\alpha^T XX^T y + y^T y) + \frac{\lambda}{2}(\alpha^T XX^T\alpha) \\
&\text{since } y^T XX^T\alpha \text{ is a scalar that we can freely transpose.} \\
\nabla_\alpha L = 0 &= \frac{1}{2}(2XX^T XX^T\alpha_\star - 2XX^T y) + \frac{\lambda}{2}(2XX^T\alpha_\star) \text{ using identities in the Matrix Cookbook} \\
&= XX^T(XX^T\alpha_\star - y + \lambda\alpha_\star) \\
&\iff XX^T(XX^T + \lambda I)\alpha_\star = XX^T y \\
&\iff XX^T(XX^T + \lambda I)(\alpha_\star + h) = XX^T y \\
&\iff \alpha_\star = (XX^T + \lambda I)^{-1}y + h
\end{aligned}
$$

where h is any element of the nullspace of XX^T. The last two steps follow from the general fact that $Ax = Ab \implies x + h = b$, where $h \in \text{Null}(A)$. If XX^T is invertible, then $\alpha_\star = (XX^T + \lambda I)^{-1}y$ is the unique minimizer of the loss (the null space is just $\{0\}$).

Note: the last two steps utilize our intuition about the null space of A, which is that it represents the "ambiguity" or "flexibility" in the solutions to the linear system of equations $Ax = b$, or equivalently the degree of "tolerance" in inputs to a linear operation A.

(g) Computing $\alpha_\star$ consists of computing XX^T ($O(n^2 d)$) and inverting the resulting $n \times n$ matrix ($O(n^3)$). Computing $w_\star$ consists of computing $X^T X$ ($O(nd^2)$) and inverting the resulting $d \times d$ matrix ($O(d^3)$). So you might want to find $\alpha_\star$ when $n << d$.

Problem 8: Least Squares Classification. In this problem we will implement a least squares classifier for the MNIST data set. The task is to classify handwritten images of numbers between 0 to 9.

We highly recommend you use the anaconda build of python. First you will need to install some packages and get some data.

```
bash code/get_data.sh
pip install python-mnist
pip install sklearn
pip install scipy
pip install numpy
```

Look in `hw1.py` for the skeleton code. You are **NOT** allowed to use any of the prebuilt classifiers in `sklearn`. Feel free to use any method from `numpy` or `scipy`.

a) In this problem we will choose a linear classifier to minimize the least squares objective:

$$W^* = \text{argmin}_{W \in R^{d \times k}} \sum_{i=0}^{n} \|W^T x_i - y_i\|_2^2 + \lambda \|W\|_F^2$$

We adopt the notation where we have n data points and each data point lives in d-dimensional space. k denotes the number of classes. Note that $\|W\|_F$ corresponds to the Frobenius norm of W, i.e. $\|\text{vec}(W)\|_2^2$.

Derive a closed form for $W_\star$.

Solution: First rewrite objective in matrix form (and multiply by $\frac{1}{2}$ for convinience

$$W^* = \text{argmin}_{W \in R^{d \times k}} \frac{1}{2} \|XW - Y\|_F^2 + \lambda \|W\|_F^2$$

Note $\|W\|_F^2 = tr(W^T W)$, and take derivative and set to 0:

$$\frac{1}{2} tr(2X^T XW) - tr(2X^T Y) + \lambda tr(2W)$$

$$tr(X^T XW) - tr(X^T Y) + \lambda tr(W) = 0$$

Linearity of trace

$$tr(X^T XW - X^T Y + \lambda W) = 0$$

Trace of 0 matrix is 0, so solve for that.

$$W^* = (X^T X + \lambda I)^{-1} X^T Y$$

b) As as first step we need to choose the vectors $y_i \in \mathbf{R}^k$ by converting the original labels (which are in $\{0, \ldots, 9\}$) to vectors.

We will use the one-hot encoding of the labels, i.e. the original label $j \in \{0, \ldots, 9\}$ is mapped to the standard basis vector e_j.

Fill in the function, `one_hot`, that takes a number in $0, \ldots, 9$ and returns the encoded vector.

c) Please implement the functions `train` and `predict` to achieve a test accuracy of 0.85 (that is your classifier should classify 85% of the examples correctly).

The solution to this part should be **very** simple. We have provided the diffstat for the staff solution (this includes all imports). Our solution takes 7 lines of code.

```
hw1.py |   14 +++++++-------
 1 file changed, 7 insertions(+), 7 deletions(-)
```

d) What is the algorithmic run time for computing `train`? Write your answer in $\mathcal{O}$ notation, (In terms of k, d, and n)

Solution:

$$O(d^3 + nd^2)$$

e) What could you do speed up training when $n << d$?

Solution: Use trick from problem 7, and solve for $X^T(XX^T + \lambda I)^{-1}Y$ instead