CSCI 5521 Spring 2017 Homework #4

Craig Ching
#1452647
chin0007@umn.edu

May 2, 2017

1 Problem 1

1.1 (a) Professor HighLowHigh claims: $\mathbf{v}^t = \mathbf{x}^t$ for all $t = 1, \ldots, N$. Is the claim correct?

Given:

$$\begin{aligned} \mathbf{z}^t &= W^T \mathbf{x}^t \\ \mathbf{v}^t &= W \mathbf{z}^t \end{aligned} \tag{1}$$

The matrix W is the projection matrix and $W \in \mathbb{R}^{D \times d}, d < D$, the first equation is the projection from the original space to a new subspace and the second equation is the projection from the subspace back to the original space. We can write this transformation combined as:

$$\mathbf{v}^t = WW^T \mathbf{x}^t \tag{2}$$

So the claim that $\mathbf{x}^t = \mathbf{v}^t$ can only be true if $WW^T = I$, the identity matrix. In order for this to be true, W has to be an orthogonal, square matrix, in other words, $W \in \mathbb{R}^{D \times D}$ and then $W^T = W^{-1}$. But the problem states that $W \in \mathbb{R}^{D \times d}, d < D$, so W can't be orthogonal since it isn't square and $WW^T \neq I$, therefore:

$$\mathbf{v}^t \neq \mathbf{x}^t \tag{3}$$

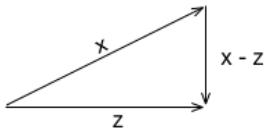
1.2 (b) Professor HighLowHigh also claims: $\sum_{t=1}^{N} ||\mathbf{x}^t||_2^2 - \sum_{t=1}^{N} ||\mathbf{v}^t||_2^2 = \sum_{t=1}^{N} ||\mathbf{x}^t - \mathbf{v}^t||_2^2$. Is the claim correct?

I believe this claim is correct. PCA is an orthogonal transformation of the data potentially to a lower dimension, though not necessarily so. If the W matrix is formed from all D covariance matrix eigenvalues and all the eigenvalues are distinct, then the transformation WW^T is the identity matrix and:

$$\mathbf{x}^t = \mathbf{v}^{\mathbf{t}} \tag{4}$$

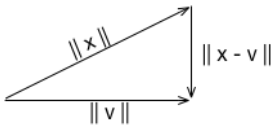
and the claim is trivially true. It is interesting to note that as $d = D, d = D-1, ...d = D-D$, the matrix WW^T starts as the identity matrix $\mathbb{I}$ and each subsequent projection to a lower dimensional space reduces the approximation of the matrix WW^T to $\mathbb{I}$.

If $d < D$, then a transformation to a lower dimensional subspace is implicit in the transformation WW^T and some information is lost. The transformation $\mathbf{z}^t = W^T \mathbf{x}^{\mathbf{t}}$ results in an orthogonal projection to the lower dimensional subspace. This is usually depicted in the projection from $\mathbb{R}^2 \mapsto \mathbb{R}$ thus:



The resulting transformation back to the original space, $\mathbf{v}^t = W\mathbf{z}^t$, loses no more information from the lower dimensional subspace and, hence, $||\mathbf{z}^t|| = ||\mathbf{v}^t||$.

Given the previous information, that no information is lost in translating from the lower dimensional space to the original space, we can depict the figure above thus:



So if we can prove that $\mathbf{v}^t \perp (\mathbf{x}^t - \mathbf{v}^t)$, then by Pythagoras' theorem, we can prove the claim true. The proof relies on the fact that for any two vectors that are orthogonal, their inner product is 0. Therefore, we want to prove:

$$(\mathbf{x}^t - \mathbf{v}^t) \cdot \mathbf{v}^t = 0 \tag{5}$$

Given $\mathbf{v}^t = WW^T\mathbf{x}^t$, we replace $\mathbf{v}^t$ with $WW^T\mathbf{x}^t$:

$$\begin{aligned}
(\mathbf{x}^t - WW^T\mathbf{x}^t)(WW^T\mathbf{x}^t) &= 0 \\
\mathbf{x}^tWW^T\mathbf{x}^t - (WW^T\mathbf{x}^t)(WW^T\mathbf{x}^t) &= 0 \\
\mathbf{x}^tWW^T\mathbf{x}^t - (\mathbf{x}^tWW^T)(WW^T\mathbf{x}^t) &= 0 \\
\mathbf{x}^tWW^T\mathbf{x}^t - (\mathbf{x}^t(WW^TWW^T)\mathbf{x}^t) &= 0 \\
\mathbf{x}^tWW^T\mathbf{x}^t - (\mathbf{x}^t(W(W^TW)W^T)\mathbf{x}^t) &= 0 \\
\mathbf{x}^tWW^T\mathbf{x}^t - (\mathbf{x}^t(W(\mathbb{I})W^T)\mathbf{x}^t) &= 0 \\
\mathbf{x}^tWW^T\mathbf{x}^t - (\mathbf{x}^t(WW^T)\mathbf{x}^t) &= 0 \\
\mathbf{x}^tWW^T\mathbf{x}^t - \mathbf{x}^tWW^T\mathbf{x}^t &= 0 \\
0 &= 0
\end{aligned} \tag{6}$$

Therefore:

$$(\mathbf{x}^t - \mathbf{v}^t) \cdot \mathbf{v}^t = 0 \tag{7}$$

The key to this proof is that $W^TW = \mathbb{I}$. This is true because the matrix W is assumed to contain the eigenvectors of $\mathbf{\Sigma}$ that are associated with unique eigenvalues. Therefore the columns of W are orthonormal and so $W^TW = \mathbb{I}$.

Because $\mathbf{v}^t \perp (\mathbf{x}^t - \mathbf{v}^t)$, then the following holds by Pythagoras' theorem:

$$||\mathbf{x}^t||_2^2 = ||\mathbf{v}^t||_2^2 + ||\mathbf{x} - \mathbf{v}||_2^2 \tag{8}$$

Rearranging which gives:

$$||\mathbf{x}^t||_2^2 - ||\mathbf{v}^t||_2^2 = ||\mathbf{x} - \mathbf{v}||_2^2 \tag{9}$$

This is the solution for any vector $\mathbf{x}^t$, but holds for:

$$\sum_{t=1}^{N} ||\mathbf{x}^t||_2^2 - \sum_{t=1}^{N} ||\mathbf{v}^t||_2^2 = \sum_{t=1}^{N} ||\mathbf{x}^t - \mathbf{v}^t||_2^2 \tag{10}$$

Since if this is true for any projection, it is true for all projections and the sums are trivial. Thus, the claim is proved true.

2 Problem 2

2.1 (a) Show that the stochastic gradient descent update for $v_{i,h}$ is of the form $v_{i,h}^{new} = v_{i,h}^{old} + \Delta v_{i,h}$ with the update $\Delta v_{i,h} = \eta \Delta_i^t z_h^t$, where $\Delta_i^t = -g'(a_i^t)\frac{\partial L(r_i^t, y_i^t)}{\partial y_i^t}$.

The gradient descent update rule for $\Delta v_{i,h}$ has the form:

$$\Delta v_{i,h} = -\eta \frac{\partial E}{\partial v_{i,h}} \tag{11}$$

Noting that we only need to update one weight, $v_{i,h}$, and that we are using stochastic gradient descent, we can remove the summations over k and N respectively, and rewrite the loss function that we need to evaluate as:

$$\begin{aligned} E(W, V|Z) &= L(r_i^t, y_i^t) \\ &= L(r_i^t, g(a_i^t)) \\ &= L(r_i^t, g(\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0})) \end{aligned} \tag{12}$$

According Alpaydin, the chain rule we need to evaluate is:

$$\frac{\partial E}{\partial v_{i,h}} = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial v_{i,h}} \tag{13}$$

But, since we have an activation function, $g(a_i^t)$, we will account for that with:

$$\frac{\partial E}{\partial v_{i,h}} = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial a_i^t} \frac{\partial a_i^t}{\partial v_{i,h}} \tag{14}$$

Applying the chain rule:

$$\begin{aligned} \frac{\partial E}{\partial y_i^t} &= \frac{\partial L(r_i^t, y_i^t)}{\partial y_i^t} \\ \frac{\partial y_i^t}{\partial a_i^t} &= g^{'}(a_i^t) \\ \frac{\partial a_i^t}{\partial v_{i,h}} &= \frac{\partial}{\partial v_{i,h}} (\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0}) = z_h^t \end{aligned} \tag{15}$$

Then the update rule is learning rate (η) $\times$ error (Δ_i^t) $\times$ input (z_h^t):

$$\Delta v_{i,h} = \eta \Delta_i^t z_h^t \tag{16}$$

Where the error Δ_i^t is:

$$\Delta_i^t = -g^{'}(a_i^t) \frac{\partial L(r_i^t, y_i^t)}{\partial y_i^t} \tag{17}$$

2.2 (b) Show that the stochastic gradient descent update for $w_{h,j}$ is of the form $w_{h,j}^{new} = w_{h,j}^{old} + \Delta w_{h,j}$ with the update $\Delta w_{h,j} = \eta \Delta_h^t x_j^t$, where $\Delta_h^t = g^{'}(a_h^t) \sum_{i=1}^{k} \Delta_i^t v_{i,h}$

According to Alpaydin, the update rule for $w_{h,j}$ is:

$$\frac{\partial E}{\partial w_{h,j}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial z_h} \frac{\partial z_h}{\partial w_{h,j}} \tag{18}$$

But, this time we have two activation functions, $g(a_i^t)$ and $g(a_h^t)$, for which we need to account:

$$\frac{\partial E}{\partial w_{h,j}} = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial a_i^t} \frac{a_i^t}{z_h} \frac{\partial z_h}{\partial a_h^t} \frac{\partial a_h^t}{\partial w_{h,j}} \tag{19}$$

The error from the hidden layer to the output layer is propagated to this layer. We see this in the terms:

$$\Delta_i^t = \frac{\partial E}{\partial y_i^t} \frac{\partial y_i^t}{\partial a_i^t} \tag{20}$$

We also need the partial derivative of a_i^t with respect to z_h:

$$\frac{\partial}{z_h} (\sum_{h=1}^{H} v_{i,h} z_h^t + v_{i0}) = v_{i,h} \tag{21}$$

If there are k output nodes, there are also k weights ($v_{i,h}$) and the error for each of them has to be accounted for. We account for that with:

$$\sum_{i=1}^{k} \Delta_i^t v_{i,h} \tag{22}$$

And finishing off rest of the terms in the chain rule:

$$\begin{aligned} \frac{\partial z_h}{\partial a_h^t} &= g^{'}(a_h^t) \\ \frac{\partial a_h^t}{\partial w_{h,j}} &= \frac{\partial}{w_{h,j}}(\sum_{j=1}^{d} w_{h,j} x_j^t + w_0) = x_j^t \end{aligned} \tag{23}$$

Then the update rule is learning rate (η) $\times$ error (Δ_h^t) $\times$ input (x_j^t):

$$\Delta w_{h,j} = \eta \Delta_h^t x_j^t \tag{24}$$

Where the error Δ_h^t is:

$$\Delta_h^t = g^{'}(a_h^t) \sum_{i=1}^{k} \Delta_i^t v_{i,h} \tag{25}$$

3 Problem 3

3.1 Description

MyFLDA2 is an implementation of Fisher's Linear Discriminant Analysis for two classes. This implementation is run against the scikit-learn standard dataset Boston housing prices with changes as described below. When the test file, **q3.py**, is run, a threshold is found for MyFLDA2 to predict from a given unseen observation from the data that minimizes training set error. This can take a bit to run, but there is ample feedback so that a user will know this is happening. The algorithm exhaustively runs 5-fold cross-validation against every observation in the dataset and one more for the average of the class means, and chooses a threshold for the model based on which threshold gives the lowest mean training set error. This model with that threshold set is then run against the Boston50 and Boston75 datasets a final time to report the errors. A sample run of this is documented below in the **Results** section.

3.2 Boston50 Housing Dataset

The standard Boston housing prices dataset is used. This dataset consists of 506 samples and 13 features. It is typically used for regression, but we modify it for use in classification contexts. The target data is split such that the lowest 50% of the data is labeled -1 and the highest 50% is labeled 1. No other modifications are made to the dataset.

3.3 Boston75 Housing Dataset

The standard Boston housing prices dataset is used. This dataset consists of 506 samples and 13 features. It is typically used for regression, but we modify it for use in classification contexts. The target data is split such that the lowest 75% of the data is labeled -1 and the highest 25% is labeled 1. No other modifications are made to the dataset.

3.4 Results

method: MyFLDA2
dataset: Boston50

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	mean	std dev
0.1275	0.1089	0.1980	0.1584	0.1386	0.1463	0.0304

method: MyFLDA2
dataset: Boston75

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	mean	std dev
0.0588	0.1188	0.1287	0.1089	0.1188	0.1068	0.0248

method: LogisticRegression
dataset: Boston50

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	mean	std dev
0.0784	0.2079	0.1188	0.1683	0.1089	0.1365	0.0460

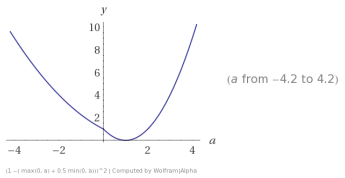
method: LogisticRegression
dataset: Boston75

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	mean	std dev
0.0784	0.1287	0.1188	0.0891	0.0693	0.0969	0.0230

4 Extra Credit

4.1 Is $L_{sq}^{(tlu)}(\mathbf{w})$ a convex function of the parameter w?

No, $L_{sq}^{(tlu)}$ is **not** a convex function. If we assume our function is in $\mathbb{R}^2$ and $y = 1$ and look at the plot of the function $(1 - (max(0, a) + 0.5min(0, a)))^2$:



We see a little "kink" at $(0, 1)$. Given this definition of convexity:

$$f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2) \tag{26}$$

If we let:

$$\begin{aligned} x_1 &= -0.04 \\ x_2 &= 0.04 \\ w &= 1.0 \\ t &= 0.9 \end{aligned} \tag{27}$$

We will show that this function is not convex. Noticing that $\mathbf{w}$ is a scalar and set to 1, we ignore it. The function we need to evaluate in our definition of convexity, given by $f(x)$ is:

$$f(x) = (1 - (max(0, x) + 0.5min(0, x)))^2 \tag{28}$$

$$\begin{aligned} t\mathbf{x}_1 + (1-t)\mathbf{x}_2 &= (0.9)(-0.04) + (1-0.9)0.04 = -0.032 \\ f(-0.032) = (1 - (max(0, -0.032) + 0.5min(0, -0.032)))^2 &= \mathbf{1.032256} \\ f(\mathbf{x}_1) = (1 - (max(0, -0.04) + 0.5min(0, -0.04)))^2 &= 1.0404 \\ f(\mathbf{x}_2) = (1 - (max(0, 0.04) + 0.5min(0, 0.04)))^2 &= 0.9216 \\ tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2) = (0.9)(1.0404) + (1-0.9)(0.9216) &= \mathbf{1.02852} \\ \mathbf{1.032256} \leq \mathbf{1.02852} &== False \end{aligned} \tag{29}$$

Therefore we conclude that the function is **not** convex.