CS189: Introduction to Machine Learning

Homework 2

Due: September 27th, 2016, 12:00 noon, NOT MIDNIGHT

Problem 1: Visualizing Eigenvectors of Gaussian Covariance Matrix

We have two one dimensional random variables $X_1 \sim \mathcal{N}(4, 4)$ and $X_2 \sim 0.5X_1 + \mathcal{N}(3, 9)$, where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. In software, draw $N = 100$ random samples of X_1 and of X_2.

(a) Compute the mean of the sampled data.

(b) Compute the covariance matrix of the sampled data.

(c) Compute the eigenvectors and eigenvalues of this covariance matrix.

(d) On a two dimensional grid with a horizontal axis for X_1 ranging from $[-15, 15]$ and a vertical axis for X_2 ranging from $[-15, 15]$, plot the following:

   i) All $N = 100$ data points

   ii) Arrows representing both covariance eigenvectors. The eigenvector arrows should originate from the mean and have magnitude equal to their corresponding eigenvalues.

(e) By placing the eigenvectors of the covariance matrix into the columns of a matrix $U = [v_1\ v_2]$, where v_1 is the eigenvector corresponding to the largest eigenvalue, we can use U' as a rotation matrix to rotate each of our sampled points from our original (X_1, X_2) coordinate system to a coordinate system aligned with the eigenvectors (without the transpose, U can rotate back to the original axes). Center your data points by subtracting the mean and then rotate each point by U', specifically $x_{\text{rotated}} = U'(x - \mu)$. Plot these rotated points on a new two dimensional grid with both axes ranging from [-15,15].

```
# Problem 1: Visualizing covariance matrices
X1 = np.random.normal(4, 2, 100)
X2 = 0.5 * X1 + np.random.normal(3, 3, 100)
X = np.vstack((X1, X2)).T
mean = np.mean(X, axis=0)
```

```
cov = np.cov(X.T)
eigenval, eigenvec = np.linalg.eig(cov)
plt.scatter(X1, X2)
ax = plt.gca()
ax.arrow(mean[0], mean[1],
         eigenvec[0][0] * eigenval[0],
         eigenvec[1][0] * eigenval[0],
         head_width=0.7, head_length=1.1,
         fc='k', ec='k')
ax.arrow(mean[0], mean[1],
         eigenvec[0][1] * eigenval[1],
         eigenvec[1][1] * eigenval[1],
         head_width=0.7, head_length=1.1,
         fc='k', ec='k')
plt.draw()
plt.xlim([-30, 30])
plt.ylim([-30, 30])
plt.show()

zero_mean_X = X - mean
zero_mean_X = eigenvec.T.dot(zero_mean_X.T).T
ax = plt.gca()
ax.arrow(0, 0, eigenval[0], 0,
         head_width=0.7, head_length=1.1,
         fc='k', ec='k')
ax.arrow(0, 0, 0, eigenval[1],
         head_width=0.7, head_length=1.1,
         fc='k', ec='k')
plt.scatter(zero_mean_X[:, 0], zero_mean_X[:, 1])
plt.draw()
plt.xlim([-15, 15])
plt.ylim([-15, 15])
plt.show()
```

Problem 2: Covariance Matrixes and Decompositions

As described in lecture, a covariance matrix $\Sigma \in \mathbb{R}^{N \times N}$ for a random variable $X \in \mathbb{R}^N$ with the following values, where $\text{cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$ is the covariance between the i-th and j-th elements of the random vector X:

$$\Sigma = \begin{bmatrix} \text{cov}(X_1, X_1) & ... & \text{cov}(X_1, X_n) \\ ... & & ... \\ \text{cov}(X_n, X_1) & ... & \text{cov}(X_n, X_n) \end{bmatrix} . \tag{1}$$

For now, we are going to leave the formal definition of covariance matrices aside and focus instead on some transformations and properties. The motivating example we will use is the N dimensional Multivariate Gaussian Distribution defined as follows when Σ is positive definite:

$$f(x) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)} . \tag{2}$$

Here, $|\Sigma|$ denotes the determinant of the matrix Σ.

(a) We usually assume that Σ^{-1} exists, but in many cases it will not. Describe the conditions for which Σ_X^{-1} corresponding to random variable X will not exist. Explain how to convert the random variable X into a new random variable X' without loss of information where $\Sigma_{X'}^{-1}$ does exist.

(b) Consider a data point x drawn from a zero mean Multivariate Gaussian Random Variable $X \in \mathbb{R}^N$ like shown above. Prove that there exists matrix $A \in R^{N,N}$ such that $x^\top \Sigma^{-1} x = \|Ax\|_2^2$ for all vectors x. What is the matrix A?

(c) In the context of Multivariate Gaussians from the previous problem, what is the intuitive meaning of $x^\top \Sigma^{-1} x$ when we transform it into $\|Ax\|_2^2$?

(d) Lets constrain $\|x\|_2 = 1$. In other words, the ℓ_2 norm (or magnitude) of vector x is 1. In this case, what is the maximum and minimum value of $\|Ax\|_2^2$? If we have $X_i \perp\!\!\!\perp X_j \ \forall i, j$, then what is the intuitive meaning for the maximum and minimum value of $\|Ax\|_2^2$? To maximize the probability of $f(x)$, which x should we choose?

Solutions

(a) We first characterize when the covariance matrix Σ_X is not invertible.

Lemma: Let $X = (X_1, ..., X_n)$ be a random vector with covariance matrix Σ_X. We have that Σ_X is singular iff there exists a $1 \leq i \leq n$, a vector $\alpha \in \mathbb{R}^{n-1}$, and a scalar $\beta \in \mathbb{R}$ such that $X_i = (X_1, X_2, ..., X_{i-1}, X_{i+1}, ..., X_n)^\top \alpha + \beta$ almost surely.

Proof. We first prove the lemma under the assumption that $\mathbb{E}X = 0$.

($\Longleftarrow$). Wlog, we assume that $i = n$. Define the random variable $Z := (X_1, X_2, ..., X_{n-1})$ and let $M := \mathbb{E}ZZ^\mathsf{T}$ denote its covariance matrix. We are given that $X_n = Z^\mathsf{T}\alpha$ a.s. (in this case, we must have $\beta = 0$ by the zero-mean assumption). Therefore,

$$\Sigma_X = \mathbb{E}\begin{bmatrix} ZZ^\mathsf{T} & ZZ^\mathsf{T}\alpha \\ \alpha^\mathsf{T}ZZ^\mathsf{T} & \alpha^\mathsf{T}ZZ^\mathsf{T}\alpha \end{bmatrix} = \begin{bmatrix} M & M\alpha \\ \alpha^\mathsf{T}M & \alpha^\mathsf{T}M\alpha \end{bmatrix} .$$

We recognize the last column of Σ_X as $\begin{bmatrix} M \\ \alpha^\mathsf{T}M \end{bmatrix}\alpha \in \mathcal{R}(\begin{bmatrix} M \\ \alpha^\mathsf{T}M \end{bmatrix})$, where $\mathcal{R}(\cdot)$ denotes the range. Therefore, we have Σ_X is rank deficient.

($\Longrightarrow$)[1]. Since Σ_X is rank deficient, there exists a vector $\mathbb{R}^n \ni \lambda \in \text{Kern}(\Sigma_X)$ with $\lambda \neq 0$. Define the random variable $Z := \lambda^\mathsf{T}X$. Observe that

$$\mathbb{E}Z^2 = \mathbb{E}\lambda^\mathsf{T}XX^\mathsf{T}\lambda = \lambda^\mathsf{T}\Sigma_X\lambda = 0 .$$

This allows us to conclude that $Z = 0$ a.s. But we have $0 = Z = \lambda^\mathsf{T}X = \sum_{i=1}^n \lambda_i X_i$. Since $\lambda \neq 0$, there must exist some $1 \leq i \leq n$ with $\lambda_i \neq 0$. We can therefore rearrange to conclude

$$X_i = \sum_{j \neq i} \frac{\lambda_j}{\lambda_i} X_j .$$

This is the desired conclusion with $\beta = 0$.

To pass to the case when $\mathbb{E}X \neq 0$, apply the result on the zero-mean vector $Y := X - \mathbb{E}X$. □

We now show how to change basis to get a random variable X' with a full rank covariance matrix. Assume again that $\mathbb{E}X = 0$ for notational simplicity. Put $r := \text{rank}(\Sigma_X)$. Let $\Sigma_X = Q\Lambda Q^\mathsf{T}$ be the eigen-decomposition of Σ_X, with Q partitioned as $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$, $Q_1 \in \mathbb{R}^{n\times r}$, $Q_2 \in \mathbb{R}^{n\times(n-r)}$, and $\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix}$ with $\Lambda_1 \in \mathbb{R}^{r\times r}$ a positive diagonal matrix. Now define

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} = Q^\mathsf{T}X = \begin{bmatrix} Q_1^\mathsf{T}X \\ Q_2^\mathsf{T}X \end{bmatrix} .$$

Now check the following properties

$$\begin{aligned} \mathbb{E}Z_1 = 0, \qquad & \mathbb{E}Z_1Z_1^\mathsf{T} = \mathbb{E}Q_1^\mathsf{T}XX^\mathsf{T}Q_1 = \Lambda_1 \succ 0 \\ \mathbb{E}Z_2 = 0, \qquad & \mathbb{E}Z_2Z_2^\mathsf{T} = \mathbb{E}Q_2^\mathsf{T}XX^\mathsf{T}Q_2 = 0 . \end{aligned}$$

[1]This argument comes from `http://math.stackexchange.com/questions/241997/about-linear-dependence-in-covariance-matrix-and-implications-on-the-variables`.

Hence, $Z_2 = 0$ a.s. This shows we can use the subspace associated to the top r eigenvectors to change the coordinates of X to a basis where the first r components are non-degenerate random variables, and the last $n - r$ components are zero (almost surely).

Furthermore, there is no loss of information. Given a realization of Z_1, we can get back a realization of X by computing the inverse transform $X = Q_1 Z_1$.

(b) Use the Spectral Decomposition Theorem to convert Σ into the following, where U is a unitary matrix of orthonormal eigenvectors $\vec{e_i}$ $\forall i \in [0...N]$ and D is a diagonal matrix with eigenvalues λ_i $\forall i \in [0...N]$ located at indices corresponding to eigenvectors in U. Note: all eigenvalues > 0 since Σ is positive definite

$$\Sigma = UDU^\top \Rightarrow \Sigma^{-1} = (UDU^\top)^{-1} = (U^\top)^{-1} D^{-1} U^{-1} = UD^{-1}U^\top \tag{3}$$

This is since a unitary matrix U is such that $U^{-1} = U^\top$. Note that if diagonal matrix D has values $d_{i,i}$ $\forall i$, then D^{-1} has value $\frac{1}{d_{i,i}}$ $\forall i$. Once again, since Σ was positive definite, the value $\frac{1}{d_{i,i}}$ exists.

Now, we decompose D^{-1} into its square-root by defining Q as a diagonal matrix with diagonal values $\frac{1}{\sqrt{d_{i,i}}}$. Verify that $QQ = D^{-1}$ and that $Q^\top = Q$. Thus, we have:

$$\Sigma^{-1} = UD^{-1}U^\top = UQQU^\top = UQQ^\top U^\top \tag{4}$$

$$\Sigma^{-1} = A^\top A \tag{5}$$

Where we defined $(UQ)^\top = A$. Therefore

$$x^\top \Sigma^{-1} x = x^\top A^\top A x = (Ax)^\top (Ax) = \|Ax\|_2^2 \tag{6}$$

Note: This process is closely related to Cholesky Decomposition, which will require one to use QR Decomposition to show that $\Sigma^{-1} = LL^\top$ for all invertible covariance matrices Σ where L is a lower triangular matrix.

(c) $x^\top \Sigma^{-1} x$ is a scalar written in vector quadratic form. It looks like an incomprehensible value, but when we convert it to $\|Ax\|_2^2$, we see that in reality its just the squared L2 norm of Ax, which measures the squared distance from the data vector x from the mean (in this case 0). Note that we can change the mean to be any arbitrary value without loss of generality.

(d) Recall from Part B our decomposition for Σ^{-1}, which was as follows where U is a unitary matrix, D is a diagonal matrix.

$$\Sigma^{-1} = UD^{-1}U^\top = A^\top A \tag{7}$$

Note that $\|x\|_2 = 1$ and $\|Ux\|_2 = 1$ since unitary matrices are orthonormal and preserve magnitude. Define $q = Ux$, we have

$$\|Ax\|_2^2 = x^\top A^\top Ax = x^\top UD^{-1}U^\top x = q^\top D^{-1}q \tag{8}$$

We can choose our x such that q will be any Euclidean Basis Vector $\vec{e_i}$ such that the ith element is 1 and all other elements are 0. Therefore, the maximum value that $\|Ax\|_2^2$ is $\frac{1}{\lambda_i}$, where λ_i is the minimum eigenvalue of Σ. The minimum value that $\|Ax\|_2^2$ is $\frac{1}{\lambda_j}$, where λ_j is the maximum eigenvalue of Σ.

If we have $X_i \perp\!\!\!\perp X_j \ \forall i, j$, then $cov(X_i, X_j) = 0 \ \forall i, j$ meaning that off diagonal terms for Σ are 0. Thus, we can find Σ^{-1} directly, where

$$\Sigma^{-1}_{i,j} = \begin{cases} \frac{1}{\sigma_i^2} \text{ if } i == j \\ 0 \text{ else} \end{cases}$$

Therefore, if we have $X_i \perp\!\!\!\perp X_j \ \forall i, j$, the maximum value that $\|Ax\|_2^2$ is $\frac{1}{\sigma_i^2}$, where σ_i^2 is the minimum variance. The minimum value of $\|Ax\|_2^2$ is $\frac{1}{\sigma_j^2}$, where σ_j^2 is the maximum variance.

To maximize $f(X)$, we want the superscript above the exponent to be minimal since there is a negative sign. Thus, for $\|Ax\|_2^2$ to be minimal, we want to choose x to be the vector corresponding to the eigenvector corresponding to the maximal eigenvector λ_j or maximum variance σ_j^2 if independent.

Problem 3: Isocontours of Normal Distributions

Let $f(\mu, \Sigma)$ denote the density function of a Gaussian random variable. Plot isocontours of the following functions:

a) $f(\mu, \Sigma)$, where $\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$.

b) $f(\mu, \Sigma)$, where $\mu = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$.

c) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ and $\Sigma_1 = \Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$.

d) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$.

e) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$.

For solutions, please see the attached appendix.

Problem 4: Featurized Linear Classifers and Gradient Descent

Recall the previous homework where we choose a linear classifier to minimize the least squares objective:

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

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 $\|W\|_F^2$ corresponds to the Frobenius norm of W, i.e. $\|\text{vec}(W)\|_2^2$. Remember that you are **NOT** allowed to use any of the prebuilt classifiers in `sklearn`. Feel free to use any method from `numpy` or `scipy`.

In this problem, you will also be implementing a feature lifting by constructing a *random* lift of the data. In particular, we will be using random features[2] of the form $\cos(W^\mathsf{T} X + b)$ where the elements of $W \in \mathbb{R}^d$ are distributed as i.i.d. zero-mean Gaussians (with some tunable σ^2) and $b \in \mathbb{R}$ distributed uniformly in the range $[0, 2\pi]$. These features look pretty simple, but are deceptively powerful. They map the data into a higher dimensional space where the classes are often more separable. And no complicated feature engineering is required.

a) In the file `hw2.py`, complete the featurization function `phi` using the scheme described above. Use the closed form solution for W^* from last time to classify MNIST digits.

b) Write out the batch gradient update equation, then apply batch gradient descent using the gradient ∇_W computed with all training examples. Our method was able to converge in approximately 15000 iterations for our choice of d and σ^2, but feel free to experiment with α, λ, d and σ to see the effect on training.

c) Similarly, write out the stochastic gradient descent update equation, then apply stochastic gradient descent using one example at a time to compute the gradient.

d) Plot the training error as a function of the number of iterations for batch gradient descent and stochastic gradient descent. Report the choice of learning rate and regularization term, and explain the difference in behavior between the two gradient descent algorithms.

e) You have been training on the 60000 digits in the training data and validating on the 10000 digits in the official MNIST test set. We've prepared a never-before-seen set of 10000 digits that will serve as your test set. Download this test set from Kaggle, generate predictions for the Kaggle data, and save your predictions to a CSV file. Your CSV file should have the header line `Id,Category`, and the ID's are zero-indexed. Submit these predictions to Kaggle. You can only submit twice per day, so get started early! Finally, write your Kaggle score in your writeup.

[2] A. Rahimi and B. Recht, Random features for large-scale kernel machines, in Proceedings of the 21st Annual Conference on Advances in Neural Information Processing Systems (NIPS), 2007.

a) Should provide code for sampling W and b, then applying the transformation ϕ. The rest - solving for the optimal weight matrix W^* - should be similar to the previous assignment.

b) Simply take the gradient of the loss function with respect to W:

$$\begin{aligned}\nabla_W \text{Loss} &= 2\sum_{i=0}^{n}(W^\mathsf{T}x_i - y_i)x_i^\mathsf{T} + 2\lambda W^\mathsf{T} \\ &= 2(W^\mathsf{T}X^\mathsf{T}X - Y^\mathsf{T}X) + 2\lambda W^\mathsf{T} \\ &= (X^\mathsf{T}X + \lambda I)W - X^\mathsf{T}Y\end{aligned}$$

Setting the gradient equal to zero, we arrive at the closed form solution:
$W^* = (X^\mathsf{T}X + \lambda I)^{-1}X^\mathsf{T}Y$.

c) Using the result of the previous expression, our gradient update rule given by: $W_{t+1} = W_t + \alpha((X^\mathsf{T}X + \lambda I)W - X^\mathsf{T}Y)$.

d) Similarly, the stochastic gradient update rule is written using just one training example as $W_{t+1} = W_t + \alpha((x_i x_i^\mathsf{T} + \lambda I)W - x_i^\mathsf{T} y_i)$.

e) Batch gradient descent tends to make slower, but more steady progress towards minimizing the loss. Stochastic gradient descent, however, is much less computationally expensive and will tend to descend more rapidly in the beginning. A comparison of the two are shown below (note that this only compares the first 20000 iterations, which does not even touch all training examples in SGD).

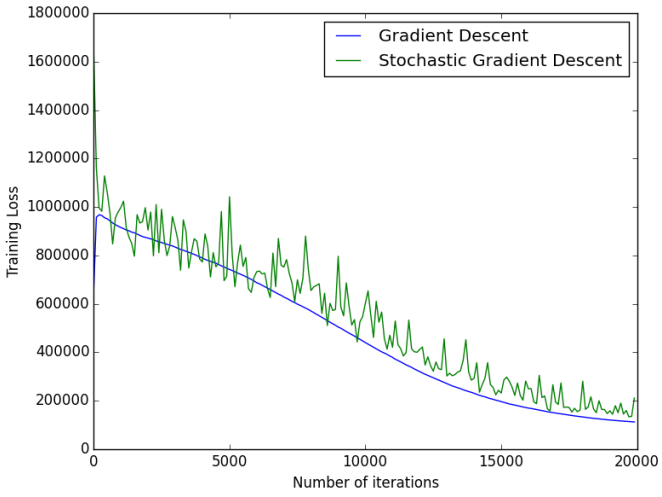

Figure 1: SGD versus Batch Gradient Descent

f) Applying linear regression with raw features yields around 69% test accuracy on the distorted images. Using this featurizer should allow you to easily reach 90% test accuracy. At least 95% test accuracy is required for full credit (?).

Problem 5: Regularization and Risk Minimization

a) Let A be a $d \times n$ matrix. For any $\mu > 0$, show that $(AA^\mathsf{T} + \mu I)^{-1}A = A(A^\mathsf{T}A + \mu I)^{-1}$.

b) Let $(x_1, y_1), \cdots, (x_n, y_n)$ be a sequence of data points. Each y_i is a scalar and each x_i is a vector in $\mathbb{R}^d$. Let $X = [x_1, \ldots, x_n]^\mathsf{T}$ and $y = [y_1, \ldots, y_n]^\mathsf{T}$. Consider the *regularized* least squares problem.

$$\min_{w \in \mathbb{R}^d} \|Xw - y\|_2^2 + \mu \|w\|_2^2$$

Show that the optimum w_* is unique and can be written as the linear combination $w_* = \sum_{i=1}^n \alpha_i x_i$ for some scalars α. What are the coefficients α_i?

c) More generally, consider the general regularized empirical risk minimization problem

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \text{loss}(w^\mathsf{T} x_i, y_i) + \mu \|w\|_2^2$$

where the loss function is convex in the first argument. Prove that the optimal solution has the form $w_* = \sum_{i=1}^n \alpha_i x_i$ If the loss function is not convex, does the optimal solution have the form $w_* = \sum_{i=1}^n \alpha_i x_i$? Justify your answer.

Solutions

a)

$$\begin{aligned}
(AA^\mathsf{T} + \mu I)^{-1}A &= A(A^\mathsf{T}A + \mu I)^{-1} \\
A &= (AA^\mathsf{T} + \mu I)A(A^\mathsf{T}A + \mu I)^{-1} \\
A(A^\mathsf{T}A + \mu I) &= (AA^\mathsf{T} + \mu I)A \\
AA^\mathsf{T}A + \mu A &= AA^\mathsf{T}A + \mu A
\end{aligned}$$

b) Start by taking the gradient of the loss function as follows:

$$\begin{aligned}
\nabla_W(\|Xw - y\|_2^2 + \mu \|w\|_2^2) &= (Xw - y)^\mathsf{T}X + \mu w^\mathsf{T} \\
&= W^\mathsf{T}X^\mathsf{T}X - Y^\mathsf{T}X + \mu w^\mathsf{T} \\
&= X^\mathsf{T}Xw - X^\mathsf{T}y + \mu w = 0
\end{aligned}$$

$$\begin{aligned}
w &= (X^\mathsf{T}X + \mu I)^{-1}X^\mathsf{T}y \\
w &= X^\mathsf{T}(XX^\mathsf{T} + \mu I)^{-1}y
\end{aligned}$$

Recall that $(XX^\mathsf{T} + \mu I)$ is positive definite and has real, positive eigenvalues when $\mu > 0$. The invertibility of this matrix implies a unique solution for w_*. $(X^\mathsf{T}X + \mu I)$

can thus be diagonalized into the form $U\Lambda U^\mathsf{T}$ by the Spectral Theorem, with $U\Lambda^{-1}U^\mathsf{T}$ as its inverse. This allows us to write

$$w_* = \sum_{i=1}^{n} \alpha_i x_i$$

where

$$\alpha_i = \sum_{j=1}^{d} y_j * u_i \Lambda^{-1} u_j^\mathsf{T}$$

and u_i is the i^{th} row of U.

c) The expression $w_* = \sum_{i=1}^{n} \alpha_i x_i$ writes w_* as a linear combination of the columns of X^T. Suppose the optimal weight vector w_* does not lie in the subspace spanned by the columns of X^T and so can be written as $w'_* = \sum_{i=1}^{n} \alpha_i x_i + v = w_* + v$, where $v^\mathsf{T} x_i = 0$ for x_i's. Using w'_* as our predictor, our loss function becomes

$$\frac{1}{n} \sum_{i=1}^{n} \text{loss}((w_*^\mathsf{T} + v^\mathsf{T}) x_i, y_i) + \mu \|w + v\|_2^2$$

$$\frac{1}{n} \sum_{i=1}^{n} \text{loss}((w_*^\mathsf{T}) x_i, y_i) + \mu (\|w\|_2^2 + \|v\|_2^2)$$

v, being orthogonal to w and the x_i's, should be set to 0 to minimize the objective function. Hence, the optimal solution has the form $w_* = \sum_{i=1}^{n} \alpha_i x_i$, which is true in the cases of both convex and non-convex loss functions.

Problem 6: MLE For Simple Linear Regression
Simple linear regression refers to the case of linear regression in which the input is a scalar quantity.

Let the data set be $\{(x_i, y_i)\}_{i=1}^n$, where each sample is drawn independently from a joint distribution over input and output: $(x_i, y_i) \sim (X, Y)$. Assume the Gaussian noise setting:

$$y_i | x_i \sim \mathcal{N}(w_0 + w_1 x_i, \sigma^2)$$

.
Show that the MLE in this simple linear regression model is given by the following equations, which may be familiar from basic statistics classes:

$$w_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2} = \frac{\sum_i x_i y_i - n\bar{x}\bar{y}}{\sum_i x_i^2 - n\bar{x}^2}$$
$$w_0 = \bar{y} - w_1 \bar{x}.$$

From statistics, we know by the Law of Large Numbers that $w_1 \approx \frac{\text{cov}(X,Y)}{\text{var}(X)}$ and $w_0 \approx \mathbb{E}[y] - w_1 \mathbb{E}[X]$ as the number of samples increases (you don't have to prove this).

$$0 = 0 - \frac{1}{2\sigma^2} \sum_{i=1}^n 2(y_i - w_0 - w_1 * x_1)(-x_1) \tag{9}$$
$$0 = \sum_{i=1}^n (y_i - w_0 - w_1 * x_1)(x_1) \tag{10}$$
$$0 = \sum_{i=1}^n y_i x_i - w_0 \sum_{i=1}^n x_i - w_1 \sum_{i=1}^n x_i^2 \tag{11}$$
$$\tag{12}$$

Note that we can use the MLE of $w_0 = \bar{y} - w_1 \bar{x}$ to substitute in.

$$0 = \sum_{i=1}^n y_i x_i - (\bar{y} - w_1 \bar{x}) \sum_{i=1}^n x_i - w_1 \sum_{i=1}^n x_i^2 \tag{13}$$
$$0 = \sum_{i=1}^n y_i x_i - \frac{1}{n} \sum_{i=1}^n y_i \sum_{i=1}^n x_i + \frac{1}{n} w_1 \sum_{i=1}^n x_i \sum_{i=1}^n x_i - w_1 \sum_{i=1}^n x_i^2 \tag{14}$$

Thus,

$$w_1(-1 * \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} x_i^2) = \sum_{i=1}^{n} y_i x_i - \frac{1}{n}\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i \tag{15}$$

$$w_1(\sum_{i=1}^{n} x_i^2 - n(\frac{1}{n}\sum_{i=1}^{n} x_i)^2) = \sum_{i=1}^{n} y_i x_i - n(\frac{1}{n}\sum_{i=1}^{n} y_i \frac{1}{n}\sum_{i=1}^{n} x_i) \tag{16}$$

$$w_1(\sum_{i=1}^{n} x_i^2 - n\bar{x}^2) = \sum_{i=1}^{n} y_i x_i - n\bar{x}\bar{y} \tag{17}$$

$$\tag{18}$$

Therefore, $w_1 = \frac{cov(X,Y)}{var(X)}$

Problem 7: Independence vs. Correlation

(a) Consider the random variables X and Y in $\mathbb{R}$ with the following conditions.

(i) X and Y can take values $\{-1, 0, 1\}$.

(ii) When X is 0, Y takes values 1 and -1 with equal probability ($\frac{1}{2}$). When Y is 0, X takes values 1 and -1 with equal probability ($\frac{1}{2}$).

(iii) Either X is 0 with probability ($\frac{1}{2}$), or Y is 0 with probability ($\frac{1}{2}$).

Are X and Y uncorrelated? Are X and Y independent? Prove your assertions. *Hint:* Graph these points onto the Cartesian Plane. What's each point's joint probability?

(b) Consider three Bernoulli random variables B_1, B_2, B_3 which take values $\{0, 1\}$ with equal probability. Lets construct the following random variables X, Y, Z: $X = B_1 \oplus B_2$, $Y = B_2 \oplus B_3$, $Z = B_1 \oplus B_3$, where $\oplus$ indicates the XOR operator. Are X, Y, and Z pairwise independent? Mutually independent? Prove it.

(a) Essentially, there are 4 possible points (X, Y) can be, all with equal probability ($\frac{1}{4}$): $\{(0, 1), (0, -1), (1, 0), (-1, 0)\}$, If graphed onto the Cartesian Plane, these point form "crosshairs".

To show that X and Y are uncorrelated, we need to prove:

$$E[(X - \mu_X)(Y - \mu_Y)] = E[X - \mu_X]E[Y - \mu_Y]$$

$$E[XY] = E[X]E[Y] = 0$$

Since for μ_X and μ_Y, we see that

$$E[X] = E[Y] = \frac{1}{2} * 0 + \frac{1}{2} * (\frac{1}{2} + \frac{-1}{2}) = 0$$

Notice for that whenever X is nonzero, Y is zero (vice versa). Thus, E[XY] = 0 since one of the terms is always zero, and we have shown that X and Y are uncorrelated. However, to show that X and Y are independent, we must show that:

$$P(X|Y) = P(X)$$

Unfortunately, this is not the case. $P(X = 0) = \frac{1}{2}$, but $P(X = 0|Y = 1) = 1$. Thus, X and Y are not independent.

(b) To prove independence for X and Y, we need to show that $P(X, Y) = P(X)P(Y)$. We know that $P(X|Y) = P(X)$ since B_1 is random 0-1 from the perspective of Y. Similar arguments can be made for all pairs of random variables. Thus, X, Y, and Z are pairwise independent.

However, $P(X|Y,Z) \neq P(X)$ since given the information in Y and Z, we can predict the value in X by the following relation. Thus, it is not mutually independent

$$Y \oplus Z = (B_2 \oplus B_3) \oplus (B_1 \oplus B_3) = B_1 \oplus B_2 \oplus 0 = B_1 \oplus B_2 = X$$

Appendix

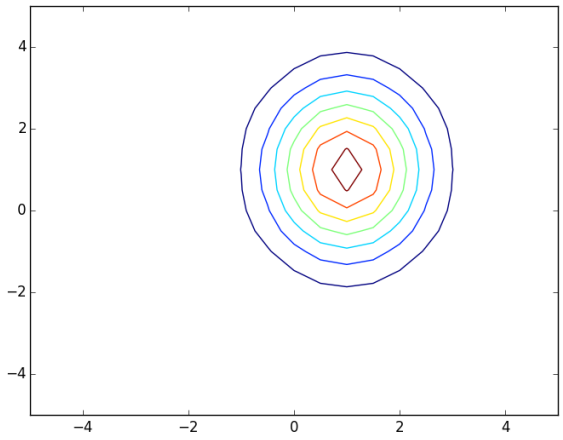

Figure 2: Problem 3i

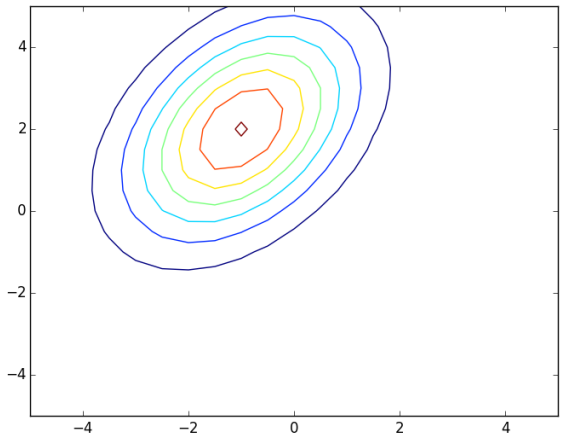

Figure 3: Problem 3ii

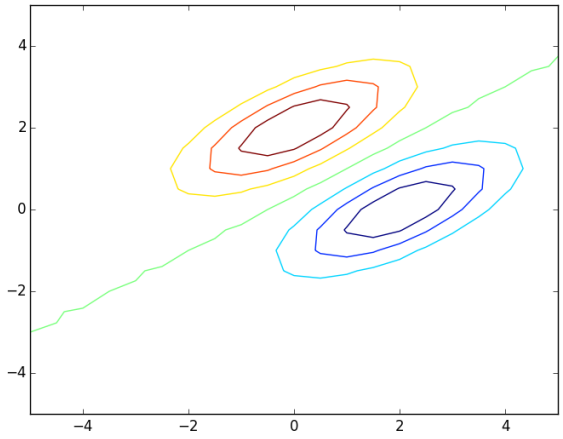

Figure 4: Problem 3iii

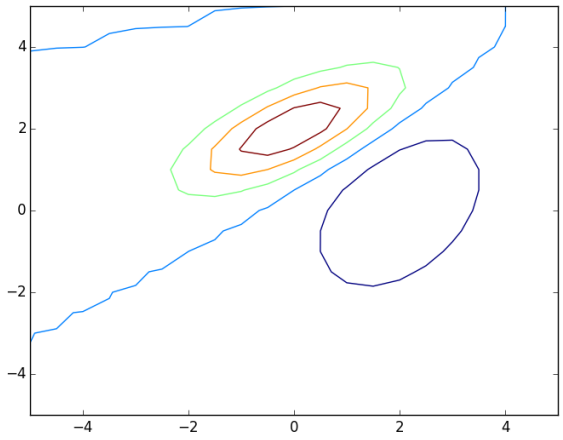

Figure 5: Problem 3iv

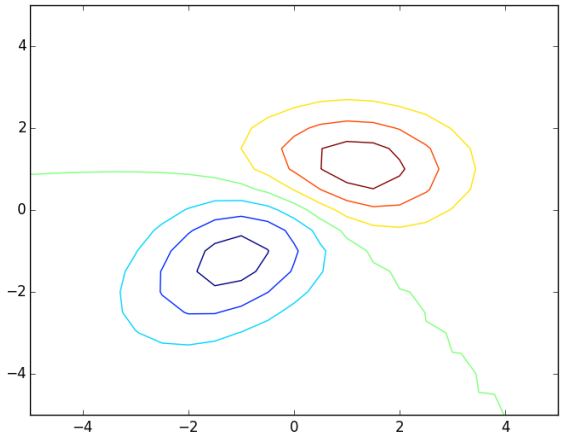

Figure 6: Problem 3v