CS 189:289A Spring 2017 Introduction to Machine Learning

Homework 3: Gaussians
Due Mon., February 27

Instructions

- We prefer that you typeset your answers using LaTeX or other word processing software. Neatly handwritten and scanned solutions will also be accepted.
- Please make sure to start **each question on a new page**, as grading (with Gradescope) is much easier that way!
- Deliverables: Submit a **PDF of your writeup** to the Homework 3 assignment on Gradescope. Include an appendix of your code a the end of your writeup. Submit your **code zip** to the Homework 3 Code assignment on Gradescope. Finally, submit **your predictions** for the test sets to Kaggle. Be sure to include your Kaggle display name and scores in your writeup.
- The assignment covers concepts on Gaussian distributions and classifiers. Some of the material may not have been covered in lecture; you are responsible for finding resources to understand it.
- Due **Monday, February 27, 2017 at 11:59 PM.**

Q1. Independence vs. Correlation

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

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

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

(iii) 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 $(\frac{1}{2})$.

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

(b) Consider three Bernoulli random variables B, C, and D which take values $\{0, 1\}$ with equal probability. Construct three more random variables X, Y, Z such that $X = B \oplus C$, $Y = C \oplus D$, and $Z = B \oplus D$, where $\oplus$ is the XOR (exclusive or) operator. Are X, Y, and Z pairwise independent? Mutually independent? Prove it.

Q2. Isocontours of Normal Distributions

Let $f(\mu, \Sigma)$ be the density function of a normally distributed random variable in $\mathbb{R}^2$. 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 $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$.

Q3. Eigenvectors of the Gaussian Covariance Matrix

Consider two one-dimensional random variables $X_1 \sim \mathcal{N}(3, 9)$ and $X_2 \sim \frac{1}{2}X_1 + \mathcal{N}(4, 4)$, where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. In software, draw $N = 100$ random two-dimensional sample points from (X_1, X_2) such that the ith value sampled from X_2 is calculated based on the ith value sampled from X_1.

(a) Compute the mean (in $\mathbb{R}^2$) of the sample.

(b) Compute the 2×2 covariance matrix of the sample.

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

(d) On a two-dimensional grid with a horizonal axis for X_1 with range $[-15, 15]$ and a vertical axis for X_2 with range $[-15, 15]$, plot

   (i) all $N = 100$ data points, and

   (ii) arrows representing both covariance eigenvectors. The eigenvector arrows should originate at the mean and have magnitudes equal to their corresponding eigenvalues.

(e) Let $U = [v_1 \quad v_2]$ be a 2×2 matrix whose columns are the eigenvectors of the covariance matrix, where v_1 is the eigenvector with the larger eigenvalue. We use $U^\top$ as a rotation matrix to rotate each sample point from the (X_1, X_2) coordinate system to a coordinate system aligned with the eigenvectors. (As $U^\top = U^{-1}$, the matrix U reverses this rotation, moving back from the eigenvector coordinate system to the original coordinate system). *Center* your sample points by subtracting the mean μ from each point; then rotate each point by $U^\top$, giving $x_{\text{rotated}} = U^\top(x - \mu)$. Plot these rotated points on a new two dimensional-grid, again with both axes having range $[-15, 15]$.

Q4. Maximum Likelihood Estimation

Let $X_1, \ldots, X_n \in \mathrm{R}^d$ be n sample points drawn independently from a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$.

(a) Suppose the normal distribution has an unknown diagonal covariance matrix

$$\Sigma = \begin{bmatrix} \sigma_1^2 & & & & \\ & \sigma_2^2 & & & \\ & & \sigma_3^2 & & \\ & & & \ddots & \\ & & & & \sigma_d^2 \end{bmatrix}$$

and an unknown mean μ. Derive the maximum likelihood estimates, denoted $\hat{\mu}$ and $\hat{\sigma}_i$, for μ and σ_i. Show all your work.

(b) Suppose the normal distribution has a known covariance matrix Σ and an unknown mean $A\mu$, where Σ and A are known $d \times d$ matrices, Σ is positive definite, and A is invertible. Derive the maximum likelihood estimate, denoted $\hat{\mu}$, for μ.

Q5. Covariance Matrices and Decompositions

As described in lecture, the covariance matrix $\mathrm{Var}(R) \in \mathbb{R}^{d\times d}$ for a random variable $R \in \mathbb{R}^d$ with mean μ is

$$\mathrm{Var}(R) = \mathrm{Cov}(R, R) = \mathbb{E}[(R-\mu)\,(R-\mu)^\top] = \begin{bmatrix} \mathrm{Var}(R_1) & \mathrm{Cov}(R_1, R_2) & \dots & \mathrm{Cov}(R_1, R_d) \\ \mathrm{Cov}(R_2, R_1) & \mathrm{Var}(R_2) & & \mathrm{Cov}(R_2, R_d) \\ \vdots & & \ddots & \vdots \\ \mathrm{Cov}(R_d, R_1) & \mathrm{Cov}(R_d, R_2) & \dots & \mathrm{Var}(R_d) \end{bmatrix},$$

where $\mathrm{Cov}(R_i, R_j) = \mathbb{E}[(R_i - \mu_i)\,(R_j - \mu_j)]$ and $\mathrm{Var}(R_i) = \mathrm{Cov}(R_i, R_i)$.

If the random variable R is sampled from the multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ with the PDF

$$f(x) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{((x-\mu)^\top \Sigma^{-1} (x-\mu))/2},$$

then $\mathrm{Var}(R) = \Sigma$.

Given n points $X_1, X_2, \dots, X_n$ sampled from $\mathcal{N}(\mu, \Sigma)$, we can estimate Σ with the maximum likelihood estimator

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \mu)\,(X_i - \mu)^\top,$$

which is also known as the covariance matrix of the sample.

(a) The estimate $\hat{\Sigma}$ makes sense as an approximation of Σ only if $\hat{\Sigma}$ is invertible. Under what circumstances is $\hat{\Sigma}$ not invertible? Make sure your answer is complete; i.e., it includes all cases in which the covariance matrix of the sample is singular. Express you answer in terms of the geometric arrangement of the sample points X_i.

(b) Suggest a way to fix a singular covariance matrix estimator $\hat{\Sigma}$ by replacing it with a similar but invertible matrix. Your suggestion may be a kludge, but it should not change the covariance matrix too much. Note that infinitesimal numbers do not exist; if your solution uses a very small number, explain how to calculate a number that is sufficiently small for your purposes.

(c) Consider the normal distribution $\mathcal{N}(0, \Sigma)$ with mean $\mu = 0$. Consider all vectors of length 1; i.e., any vector x for which $|x| = 1$. Which vector(s) x of length 1 maximizes the PDF $f(x)$? Which vector(s) x of length 1 minimizes $f(x)$? (Your answers should depend on the properties of Σ.) Explain your answer.

Q6. Gaussian Classifiers for Digits and Spam

In this problem, you will build classifiers based on Gaussian discriminant analysis. Unlike Homework 1, you are NOT allowed to use any libraries for out-of-the-box classification (e.g. `sklearn`). You may use anything in `numpy` and `scipy`.

The training and test data can be found on Piazza in the post corresponding to this homework. Don't use the training/test data from Homework 1, as they have changed for this homework. Submit your predicted class labels for the test data on the Kaggle competition website and be sure to include your Kaggle display name and scores in your writeup. Also be sure to include an appendix of your code at the end of your writeup.

(a) Taking pixel values as features (no new features yet, please), fit a Gaussian distribution to each digit class using maximum likelihood estimation. This involves computing a mean and a covariance matrix for each digit class, as discussed in lecture. *Tip:* You may, and probably should, contrast-normalize the images before using their pixel values. One way to normalize is to divide the pixel values of an image by the l_2 norm of its pixel values.

(b) (Written answer) Visualize the covariance matrix for a particular class (digit). How do the diagonal terms compare with the off-diagonal terms? What do you conclude from this?

(c) Classify the digits in the test set on the basis of posterior probabilities with two different approaches.

(i) Linear discriminant analysis (LDA). Model the class conditional probabilities as Gaussians $\mathcal{N}(\mu_C, \Sigma)$ with different means μ_C (for class C) and the same covariance matrix Σ, the average covariance matrix of the 10 classes.

Hold out 10,000 randomly chosen training points for a validation set. Classify each image in the validation set into one of the 10 classes (with a 0-1 loss function). Compute the error rate and plot it over the following numbers of randomly chosen training points: [100, 200, 500, 1,000, 2,000, 5,000, 10,000, 30,000, 50,000]. (Expect some variance in your error rate when few training points are used.)

(ii) Quadratic discriminant analysis (QDA). Model the class conditionals as Gaussians $\mathcal{N}(\mu_C, \Sigma_C)$, where Σ_C is the estimated covariance matrix for class C. (If any of these covariance matrices turn out singular, implement the trick you described in Q5.(b). You are welcome to use k-fold cross validation to choose the right constant(s) for that trick.) Repeat the same tests and error rate calculations you did for LDA.

(iii) (Written answer.) Which of LDA and QDA performed better? Why?

(iv) Train your best classifier with `train.mat` and classify the images in `test.mat`. Submit your labels to the online Kaggle competition. Record your optimum prediction rate in your submission. You are welcome to compute extra features for the Kaggle competition. If you do so, please describe your implementation in your assignment. Please use extra features **only** for this portion of the assignment.

In your submission, include plots of error rate versus number of training examples for both LDA and QDA. Also include tables giving the error rates (as percentages) for each number of training examples for both LDA and QDA. Include written answers where indicated.

(d) Next, apply LDA or QDA (your choice) to spam. Submit your test results to the online Kaggle competition. Record your optimum prediction rate in your submission. If you use additional features (or omit features), please describe them.

Optional: If you use the defaults, expect relatively low classification rates. The TAs suggest using a bag-of-words model. You may use third-party packages to implement that if you wish. Also, normalizing your vectors might help.

(e) *Extra for Experts:* Using the `training_data` and `training_labels` in `spam.mat`, identify 10 words in your features set corresponding to the maximum and minimum variances. Use *k*-fold cross validation to train your classifier using only 10 variance-maximum words and record your average error rate. Do the same with the 10 minimum-variance words. What do you notice?