COSC 522 Machine Learning (Fall 2024)

Project 1: Supervised Learning Using Baysian Decision Rule - Two Category Classification (Due 09/12)

Objective

The objective of this project is, first of all, to learn how to implement supervised learning algorithms based on Baysian decision theory. The second objective is to get you familiar with the design flow and design consideration when applying machine learning algorithms - basically, to get more insights into problem solving using machine learning. Some practical considerations include, for example, 1) the selection of the right pdf model to characterize the data distribution in the training set, 2) the effect of incorporating prior probability, 3) the different ways to evaluate the performance of the learning algorithm, and 4) how differently the same ML algorithm performs when applied to different datasets.

Data Sets

We will create our own dataset, given known pdfs! We call this a synthetic dataset. Following is a detailed description of samples in the dataset:

- **Dimension** (or the number of features): Each sample has 2 features (2-d), representing the coordinates of a point on the Cartesian plane.
- **pdf:** The samples are taken from a Gaussian distribution. Since we have a 2-d feature/data set, the pdf is a 2-d Gaussian instead of the 1-d Gaussian we used in Lectures 2 & 3. In a 2-d Gaussian, both "$\mathbf{x}$" and "$\boldsymbol{\mu}$" are 2-d column vectors. Instead of using std (σ) to quantify the spread as in 1-d Gaussian, the so-called "covariance matrix (Σ)" is used. For samples with 2 features, the dimension of the covariance matrix is 2x2,

  $$\Sigma_{\mathrm{j}} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix} \quad \mathrm{j}=1, 2$$

  where *j* is the class index, the two diagonal elements, (σ_{11}, σ_{22}), indicate the std (or spread) along each dimension, and the off-diagonal elements, ($\sigma_{12} = \sigma_{21}$), indicate the correlation between the two features. If the two features are completely independent from each other (i.e., you cannot use one feature to infer the other), then $\sigma_{12} = \sigma_{21}$=0. Please do a self-study on covariance matrices if you are not familiar with the physical meaning of each element of the matrix.

$$p(\vec{x}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1}(\vec{x}-\vec{\mu})\right]$$

$\vec{x}$: d - component column vector

$\vec{\mu}$: d - component mean vector

Σ : d - by - d covariance matrix

$|\Sigma|$: determinant

Σ^{-1} : inverse

- **Number of classes**: The samples belong to one of two classes and you will be provided with two pdf's, $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_2)$. See the concrete values in the given playbook.
- **Number of samples**: Generate 300 samples from $p(\mathbf{x}|\omega_1)$ and 300 samples from $p(\mathbf{x}|\omega_2)$ to form a "training set" of 600 samples. Generate 150 samples from $p(\mathbf{x}|\omega_1)$ and 150 samples from $p(\mathbf{x}|\omega_2)$ to form a "testing set" of 300 samples.
- The notebook that generates samples from a 2-d Gaussian is provided to help you jump start on the project.

Algorithm

You need to implement both parametric learning (with three variants) and non-parametric learning (i.e., kNN) based on Baysian decision theory. The three variants of parametric learning include 1) minimum Euclidean distance classifier (linear machine), 2) minimum Mahalanobis distance classifier (linear machine), and 3) the generic form of Baysian decision rule (quadratic machine).

Performance Metrics

Three metrics are used to evaluate the performance of the ML algorithms, including 1) overall classification accuracy, 2) classwise classification accuracy, and 3) run time.

Tasks

- Task 1 (10 pts): Create your own synthetic dataset using the instructions given in the "Dataset" section. Play around the sample notebook using 1) $\sigma_{12} = \sigma_{21}=0$, 2) $\sigma_{12} = \sigma_{21}=0.5$, and 3) $\sigma_{12} = \sigma_{21}=-0.5$. Comment on the different patterns you observe from using different cross-correlation values. Show the scatter plots on the same figure of the 3 scenarios. After you finish playing, go back to the provided means and covariance matrices for the two classes. Save the samples as synth.tr and synth.te. Show a scatter plot of the training samples and a scatter plot of the testing samples. Use different colors for samples from different classes.
- Task 2 (15 pts): Implement kNN yourself. Plot a figure with the x-axis showing the different "k" values in kNN and the y-axis showing the overall classification accuracy.
- Task 3 (50 pts): Implement the three variants of parametric learning with Gaussian pdf. Assuming equal prior probability, generate a table summarizing the overall classification accuracy, classwise accuracy, and run time of the four supervised learning algorithms,

with each row indicating a learning algorithm and each column indicating a performance metric. For kNN, choose the best "k" you obtained from Task 2.

- Task 4 (10 pts): Provide a comprehensive discussion (0.5 ~ 1 page) on the results shown in the table, including the effect of using different assumptions of the covariance matrices.
- Task 5 (15 pts): Illustrate the four decision boundaries from the four supervised classification algorithms on the same figure as the scatter plot of the testing dataset. Comment on the differences.
- Bonus Task 1 (10 pts): Experience skewed dataset and effect of unequal prior probability. Generate 300 samples from $p(\mathbf{x}|\omega_1)$ and 100 samples from $p(\mathbf{x}|\omega_2)$ to form a "training set" of 400 samples. Generate 150 samples from $p(\mathbf{x}|\omega_1)$ and 30 samples from $p(\mathbf{x}|\omega_2)$ to form a "testing set" of 180 samples. Note that we intentionally create an unbalanced dataset (or a skewed dataset) so that we can investigate the effect of unequal prior probability. Assuming minimum Euclidean distance classifier is used, generate a plot with the y-axis indicating the overall classification accuracy and the x-axis indicating $P(\omega_1)$ changing from 0 to 1 (correspondingly, $P(\omega_2)$ changes from 1 to 0).

Deliverables

- Report with requested contents from each of the five tasks in .pdf file
- Source code in a .tar or .zip file
- Please name your report and the source code using the following convention: xY_proj1.z where "x" is your firstname, "Y" is the first letter of your last name (capitalized), and "z" is either ".pdf" for report or ".tar" or ".zip" for source code. For example, if I submit for project 1, the file name should be "hairongQ_proj1.pdf"
- Grading policy: If you use jupyter notebook or colab, the TA should be able to follow your notebook and regenerate the results you reported. None of your submitted results should be the same due to the randomness in data generation. Please save the data in two files, synth.tr and synth.te, and tar/zip these two files with your source code submission.