Human-Oriented Robotics Winter term 2013/2014
Prof. Dr. Kai Arras, Social Robotics Lab University of Freiburg
Lab instructors: Timm Linder, Luigi Palmieri, Billy Okal Department of Computer Science

Exercise 4: Naïve Bayes Classifier

Submission: Send your solution to `palmieri@informatik.uni-freiburg.de` until December 2, 2013 with subject line "`[exercises] Sheet 4`". All files (Matlab scripts, exported figures, handwritten notes in pdf/jpg format) should be put into a single zip file named `lastname_sheet4.zip`.

For this exercise, you will need to download a data file from the course website. The file contains a real dataset for people detection in 2D range data and a Matlab frame to be filled by you.

Exercise 4.1: Test Set and Training Set

In supervised learning, the annotated dataset is usually split into a training set for learning and an independent test set for verifying if the learned classifier generalizes well.

a) Load the provided dataset `peopleofficedata.txt` into Matlab. It contains a big matrix of space-separated numbers where each row corresponds to a training pair $(\mathbf{x}_i, y_i)$ which is either a person or not a person (a background object like a wall, tree, or chair). Let N be the number of rows/training samples.

The attributes for each training sample are $m = 15$ scalar features in columns 1–15. They make up the input vector $\mathbf{x}$ and characterize segments of adjacent 2D laser points obtained from the segmentation algorithm that we considered in Exercise 1. They encode geometrical and statistical properties of the segments such as segment width, linearity, number of points, segment curvature, convexity, and so on.

The output variable y is given in the 16th and last column. This is the annotated class label of each training pair where $y = +1$ represents the positive class (a person) and $y = -1$ represents the negative class (background/non-person).

b) Divide the dataset into a training set and a test set. The training set should contain $^2/_3$ of the data, and the test set $^1/_3$. Randomly draw samples from the original data set for this purpose. **Hint:** use `randperm`.

c) Calculate the relative frequency of positive class instances and negative class instances among the samples in your test and training sets.

Exercise 4.2: Naïve Bayes Classifier for Person Detection in 2D Range Data

Our goal is to learn a detector of people in 2D range data using a Naïve Bayes classifier.

a) Train a Naïve Bayes classifier by estimating the parameters of Gaussian likelihoods for all features and each of the two classes $\mu_{i,y}, \sigma^2_{i,y}$, with $i \in \{1, \ldots, m\}, y \in \{-1, 1\}$. Assume Bernoulli priors and estimate them using the relative frequency terms computed above.

b) Implement the inference step. First, compute the two class-conditional densities $P(\mathbf{x}|\mathcal{C}_{-1})$, $P(\mathbf{x}|\mathcal{C}_{+1})$ using the trained parameters, then calculate the posterior probabilities over the classes $P(\mathcal{C}_{-1}|\mathbf{x})$, $P(\mathcal{C}_{+1}|\mathbf{x})$ using Bayes' rule.

c) Decision. Assign input $\mathbf{x}_i$ the class with the larger posterior probability.

Exercise 4.3: Testing the Classifier

a) To evaluate the learned classifier we will use the test set. Classify all $(\mathbf{x}_i, y_i)$ pairs in the test set and consider the following performance metrics:

- If the true label (the one in the dataset) is a person and your classifier predicts a person, then this is called a **true positive** (tp)
- If the true label is a non-person and your classifier predicts a person, this is called a **false positive** (fp)
- If the true label is a person and your classifier predicts a non-person, then you have a **false negative** (fn)
- If the true label is a non-person and your classifier predicts a non-person, this is a **true negative** (tn)

Count these events on the test set as well as the training set and compute their corresponding relative frequencies in percent.

How would a good classifier perform in terms of these numbers?

b) Machine learning is usually not a one-shot process of building a dataset, defining features and learning a classifier, but rather an **iterative process** of learning, analyzing the results, modifying the data, the features and/or the classifier, and repeating. Here, we want to take a closer look onto our 15 features. Not all features may be equally informative for this classification task. Some of them might even have a negative impact onto the results.

Take out single features or multiple features at a time and re-run learning and testing (removing features is best done by deleting columns of the original data matrix). You may try any combination of features that appears interesting to you, you can even try to solve the classification task with a single feature. You are free to experiment, there is no "best" solution for this exercise.

Can you improve the classifier performance?