Exercise 4

Deadline: 24.11.2014

In the first few exercises we focused on classification tasks. This exercise is mainly dedicated to generative models. In particular we are looking into Naive Bayes and the Density Tree.

Reminder: Generative Models

Discriminative models, like the nearest neighbor classifier or decision trees base their classification decision on the **posterior probability** $p(y|\mathbf{x})$ of the label y for given features $\mathbf{x}$. While this is completely sufficient for their task of classification one does not gain any insight on the distribution of the different classes in the feature space from which the training data (and hopefully the test data as well) were drawn. By learning the **likelihood** $p(\mathbf{x}|y)$, the **prior** $p(y)$ and the **evidence/feature density** $p(\mathbf{x})$ one is, due to the Bayes' theorem:

$$p(y|x) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}, \tag{1}$$

able to reconstruct the posterior. Therefore one is able to perform classification tasks. In addition one can generate new data from the likelihood distribution.

Overall Goal of this Exercise

This time we want to teach our algorithms to produce handwritten digits. To learn a meaningful likelihood we will need more data than we used so far. On `http://yann.lecun.com/exdb/mnist/` one can find a set of overall 70000 gray-scale images of handwritten digits. The original images are of size 28×28. For the generative procedure we want to keep the number of dimensions of the feature space as small as possible. You will find a compressed and by shifting increased dataset as well as the code that produced it on `https://dl.dropboxusercontent.com/u/1537789/mnist.zip`. For debugging reasons we will develop the algorithms again in a reduced featurespace. You are supposed to create this featurespace on your own. You can eg. take a similar reduction procedure as you used in exercise 2. All exercises that you should do using the dimensionally reduced dataset will be marked with a **(A)**. Parts of exercises where you are supposed to work on the full featurespace are marked by a **(B)**.

1 Naive Bayes

We want to approximate the likelihood via a piecewise constant function - a histogram. We realized that N bins in each of the d dimensions will explode very fast. We will never be able to gather enough training data to fill all of the N^d bins for real world problems. We can resolve this problem naively by assuming independence of all features. This leads to a decomposition of the posterior:

$$p(y = k|\mathbf{x}) = \left(\prod_{j=1} p_{jk}(x_j) \right) \frac{p(y = k)}{p(\mathbf{x})} \tag{2}$$

where we use the entries p_{ljk} of one dimensional histograms (l, j and k are the indices of the bins the features and the class):

$$p_{jk}(x_j) = \sum_{l=1}^{L} p_{ljk} \mathbb{1}[x_i \in \text{bin } l] \tag{3}$$

1.1 (A) Classification (5 points)

Write a Naive Bayes classifier can distinguish threes from eights. Train it on the given training set and test it on the given test set (you will need to compute the priors as well). Think about the right binning in the two feature dimensions. Please report your error rate and visualize both likelihoods in one 2D plot (you may be able to reuse some of your code that you used in ex02 to visualize LDA/QDA).

1.2 (B) Generate Threes (7 points)

Use your functions from 1.1 to construct the likelihood from 1D histograms for the full featurespace. Now think about a method to sample from this distribution. Please explain your sample method in detail. Finally sample images from handwritten threes from your distribution and inspect your results visually.

2 Density Tree

As we learned, Density Trees are a way to keep possible interactions between features and still requires only manageable amount of bins by adaptive binsizes.

2.1 Building the DT (8 points)

Consider one class at a time. Argue about the termination criteria that you want to use. Do you think it makes sense to restrict the depth of the tree. If yes what depth do you think is appropriate? Do you think it helps to restrict the minimum number/density of instances per node or do you want to allow for empty bins? It may also be beneficial to split your dataset in each node, compute the split positions on the one set and evaluate them on the remaining samples. Choose the setup that looks most promising to you. Justify your decisions.
Implement your training algorithm in a way that the split criterion is encapsulated in one function. This way you can test the theoretically derived split criterion against a naive one: splitting in the middle of 2 samples. Implement both split criteria. For the first, the theoretically best criterion you have to pay attention to this phenomenon: If you are splitting in a feature dimension that was spitted before, the optimal split can be arbitrarily close to the old split. This is because there can be one sample directly at the boundary. Therefore a split resulting in a minimal volume will lead to a divergence in the gain. Think about adequate countermeasures. Explain your ideas in detail and implement them.

2.2 (A) Classification (2 points)

Use the likelihood that your DT gives you to build a classifier. Use it for the same classification task as in 1.1. Visualize the adaptive bins of your DT in 2D. Do both the classification and the visualization for the theoretically optimal and the naive one.

2.3 (B) Generate Threes (5 points)

Use your functions from 2.1 to construct the likelihood. Again we need a function to sample from this distribution. Please explain how you perform the sampling in detail. Generate new unseen samples of handwritten threes and inspect them visually.

3 (B) Combine Density Tree and Naive Bayes (13 points)

As we saw in the lecture, the copula allows us to analyze the interactions between the features. Since the Naive Bayes models everything but the interactions they complement each other. By modeling the Copula and not the original distribution with a density tree one can exploit the modeling power better. In general, a copula density $\tilde{p}$ mediates between the marginal cdfs F_i and the joint probability density p:

$$p(x_i, \dots, x_d)) = \tilde{p}\,(u_1 = F_1(x_1), \dots, u_d = F_d(x_d)) * p_1(x_1) * \cdots * p_d(x_d) \tag{4}$$

The Jacobian of the transformation

$$x_1, x_2, \dots, x_d \rightarrow u_1, u_2, \dots, u_d = F_1(x_1), F_2(x_2), \dots, F_d(x_d) \tag{5}$$

will, as you know from the lecture, be the marginal pdfs of $\mathbf{x}$ that the Naive Bayes can provide. $\tilde{p}$ should be modeled via a density tree. Sample from this combined method and compare your sampled threes.

Regulations

Please hand in the python code, figures and explanations (describing clearly which belongs to which). Non-trivial sections of your code should be explained with short comments, and variables should have self-explanatory names. Plots should have informative axis labels, legends and captions. Please enclose all results into a single .pdf document and hand in the .py files that created the results. Please email the solutions to `niko.krasowski@iwr.uni-heidelberg.de` before the deadline. You may hand in the exercises in teams of maximally three people, which must be clearly named on the solution sheet (one email is sufficient). Discussions between different teams about the exercises are encouraged, but the code must not be copied verbatim (the same holds for any implementations which may be available on the WWW). Please respect particularly this rule, otherwise we cannot give you a passing grade. Solutions are due by email at the beginning of the next exercise. For each exercise there will be maximally 20 points assigned. If you have 50% or more points in the end of the semester you will be allowed to take the exam.