Exercises for
Pattern Analysis
Marco Bögel, Sebastian Käppler
Assignment 5, 28.05.2015

General Information:

Lecture (3 SWS):	Mo 08.30 – 10.00 (H16) and Tue 08.15 – 09.45 (H16)
Exercises (1 SWS):	Tue 12.15 – 13.15 (02.134-113) and Thu 8.30 – 9.30 (E1.12)
Certificate:	Oral exam at the end of the semester
Contact:	marco.boegel@fau.de sebastian.kaeppler@fau.de

Manifold Forests

Exercise 1

Manifold Learning deals with the problem of learning the structure of high dimensional data and the mapping of these data into a lower dimensional space. Decision forests, as introduced in the lecture, can also be used for manifold learning.

Given a set of N unlabeled measurements $\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_N\}$, with $\boldsymbol{v}_i \in \mathbb{R}^d$, we want to determine a smooth mapping function $\boldsymbol{f} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, with $\boldsymbol{f}(\boldsymbol{v}_i) = \boldsymbol{v}'_i$ that approximately preserves the geodesic distances between each data point, where $d' << d$.

(a) The mapping function $\boldsymbol{f}$ is supposed to preserve the distances (similarity, affinity) between the data points. Define the affinity model for N data points.

(b) There are different ways to define the distances between measurements. Write down the definitions for the Binary affinity and the Gaussian affinity.

(c) What is the benefit of using a collection of random trees opposed to a single decision tree. How is the affinity matrix defined for the manifold forest?

(d) Finally, the low-dimensional embedding function is estimated. Here, we use Laplacian eigenmaps. Given a graph whose nodes are the N input points and the affinity matrix W, define the $N \times N$ normalized Graph-Laplacian matrix L!

(e) Use the eigen-decomposition on L to find the mapping function $\boldsymbol{f}$. Let $\boldsymbol{e}_0, \boldsymbol{e}_1, ..., \boldsymbol{e}_{N-1}$ be the solutions, corresponding to increasing eigenvalues. Write down the resulting mapping function.

(f) After the manifold forest was trained on the N input points, additional data has become available. How can a point $\boldsymbol{a}$ be mapped to the low-dimensional space without retraining the entire manifold forest?

Exercise 2 Matlab exercise

The goal of this exercise is to gain practical experience in using the Random Forest for classification.

(a) Download the Matlab code from the exercise homepage.

(b) In *RF_example.m*, the Random Forest classifier is used to classify the data from the previous exercise sheet. Matlab's *treebagger* function with default parameters is used to train the Forest. However, the classifier performs poorly due to overfitting. Think about why this occurs. Additionally, you should explore what parameters of the random forest can be adjusted to improve its performance on this data set.

(c) **Bonus exercise**: Download *RF_competition.mat* from the homepage. This data set contains features and class labels for a three-class problem with a four dimensional feature vector. Train a Random Forest classifier on the data. Try to find the parameters of the Forest which optimize the classification rate. Remember that optimizing the training error might not be the same as optimizing the test error. Once done, submit the parameters of your tree until June 6th. The winner will be rewarded with a small prize.