

# CAP5638 Project 1

## Classification Using Maximum-likelihood, Parzen Window, and k-Nearest Neighbor

Suhib Sam Kiswani

October 26, 2015

### TODO

---

*You need to turn in a report, including description of your algorithms and implementations, experimental results, and comparison and evaluation and analysis of the three different methods.*

---

### Abstract

For each dataset, you need to first estimate the necessary data normalization and parameters using the training set only (you need to document the results for different choices of parameters on the training set in the leaveone- out sense for Parzen windows and k-nearest neighbor rule) and then document the classification results on the test set using each of the methods. Then you need to compare different methods in terms of classification performance and required time for classification, and give justifications for your observed empirical results.

## 1 Implementation Details

The algorithms were implemented in *Python 3.4*, with a dependence on the *numpy* module from the *scipy* [1] library.

### 1.1 Maximum likelihood estimation

#### TODO

---

*You need to choose proper parametric forms for the underlying probability distributions. In the report, you need to specify the parametric forms, the maximum-likelihood estimate for the parametric forms, and the resulting discriminant functions for classification for each dataset.*

---

### 1.1.1 Gaussian Model

The parameters  $\hat{\mu}^{\omega_i}$  and  $\hat{\Sigma}^{\omega_i}$  are calculated for each class  $\omega_i$  in the training data set:

$$\hat{\mu}^{\omega_i} = \frac{1}{n} \sum_{k=0}^n x_k^{\omega_i} \quad \hat{\Sigma}_{i,j}^{\omega_i} = \frac{1}{n} \sum_{k=0}^n (x_{k,i}^{\omega_i} - \hat{\mu}_i^{\omega_i})(x_{k,j}^{\omega_i} - \hat{\mu}_j^{\omega_i})^T$$

Then, the discriminant functions are calculated:

$$W = -0.5 (\hat{\Sigma}^{\omega_i})^{-1} \quad w = (\hat{\Sigma}^{\omega_i})^{-1} \cdot \hat{\mu}^{\omega_i}$$
$$\alpha^{\omega_i} = -0.5 [\hat{\mu}^{\omega_i, T} \cdot \hat{\Sigma}^{\omega_i, -1} \cdot \hat{\mu}^{\omega_i} - \log |\hat{\Sigma}^{\omega_i}|]$$
$$g_{\omega_i}(x) = x^T W x + w^T x + \alpha^{\omega_i}$$

A sample  $x$  is assigned to class  $\omega_i = \arg \max_{\omega_i} |g_{\omega_i}(x)|$ .

### 1.1.2 Uniform Model

TODO

---

write out eqns

---

## 1.2 Parzen window estimation

TODO

---

*You need to choose proper window functions, which you need to specify in the report along with the resulting discriminant functions for classification for each dataset. Here we choose the parameter values using the leave-one-out performance on the training set: For a set of candidate values, we compute the leave-one-out performance on the training set for each candidate and the optimal one is the one that gives the best leave-one-out performance (in case there are ties, specify how the ties will be broken).*

---

## 1.3 Basic k-nearest neighbor

TODO

---

*In this case, you need to implement the k-nearest neighbor classifier by first finding the first k nearest neighbors of an input and then classifying it as the class that appeared most among the nearest neighbors. Here you need to specify the distance measure between two inputs. Here we choose the parameter values using the leave-one-out performance on the training set: For a set of candidate values, we compute the leave-one-out performance on the training set for each candidate and the optimal one is the one that gives the best leave-one-out performance (in case there are ties, specify how the ties will be broken).*

---

## 2 Experimental Results

### TODO

---

*Document the classification results on the test set using each of the methods. For each dataset, you need to first estimate the necessary data normalization and parameters using the training set only (you need to document the results for different choices of parameters on the training set in the leaveone-out sense for Parzen windows and k-nearest neighbor rule)*

---

### 2.1 Iris Data Set

#### TODO

---

*you need to first estimate the necessary data normalization and parameters using the training set only (you need to document the results for different choices of parameters on the training set in the leaveone-out sense for Parzen windows and k-nearest neighbor rule) and then document the classification results on the test set using each of the methods.*

---

### 2.2 UCI Wine Data Set

#### TODO

---

*you need to first estimate the necessary data normalization and parameters using the training set only (you need to document the results for different choices of parameters on the training set in the leaveone-out sense for Parzen windows and k-nearest neighbor rule) and then document the classification results on the test set using each of the methods.*

---

### 2.3 Handwritten Digits Data Set

#### TODO

---

*you need to first estimate the necessary data normalization and parameters using the training set only (you need to document the results for different choices of parameters on the training set in the leaveone-out sense for Parzen windows and k-nearest neighbor rule) and then document the classification results on the test set using each of the methods.*

---

## 3 Analysis

### TODO

---

*You need to compare different methods in terms of classification performance and required time for classification, and give justifications for your*

*observed empirical results.*

---

## 4 Extra Credit

### TODO

---

*Please state clearly in your report if you have implemented any of the following extra credit options.*

---

### 4.1 Speed up using k-d tree

#### TODO

---

*With  $k$  from 1 to 10 with an increment of 1, first build a k-d tree from the training set and then classify the test samples using the k-nearest neighbor classifier by finding the nearest neighbors using the k-d tree. Compare the classification accuracy and the number of distance calculations with the basic k nearest neighbor implementation on the three datasets. Summarize your observations and justify your results.*

---

### 4.2 Recognition of my handwritten digits

#### TODO

---

*Apply the best classifier you have for hand written digit recognition on a test set consisting of your own written digits (you need to create the dataset). Document the classification performance, what you have done to improve the performance, and any additional issues you have handled.*

---

## References

- [1] Jones E, Oliphant E, Peterson P, *et al.* **SciPy: Open Source Scientific Tools for Python**, 2001-, <http://www.scipy.org/> [Online; accessed 2015-10-24].