Exercises for
Pattern Recognition
Peter Fischer, Shiyang Hu
Assignment 3, 4./7.11.2014

SOLUTION

General Information:

Exercises (1 SWS):	Tue 12:15 – 13:45 (0.154-115) and Fri 08:15 – 09:45 (0.151-115)
Certificate:	Oral exam at the end of the semester
Contact:	peter.fischer@fau.de, shiyang.hu@fau.de

Naive Bayes and Gaussian Classifier

Exercise 1 In the previous exercises, we assumed that all dimensions of the feature vector were mutually dependent. Thanks to the low dimensional feature space, the computational complexity was still very low. However, in this exercise, we will assume mutual independent feature vector components.

Assume that the features are samples from normally distributed classes. Implement the Naive Bayes approach with

$$p(\boldsymbol{x}|y) = \prod_{i=1}^{d} \mathcal{N}(x_i; \mu_i, \sigma_i^2) \tag{1}$$

where d is the dimensionality of the feature space.
Repetition: Naive Bayes
Generative model $\rightarrow$ assume a distribution with some parameterization (here: Gaussian) $p(\boldsymbol{x}|y) \sim \mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}, \Sigma)$

Assumption:
$p(\boldsymbol{x}, y) = p(\boldsymbol{x}|y) \cdot p(y) \overset{ind.}{=} p(x_1|y) \cdot \ldots \cdot p(x_n|y) \cdot p(y)$

Decision rule:
$\hat{y} = \operatorname{argmax}\left(p(y|\boldsymbol{x})\right) = \operatorname{argmax}\left(\frac{p(\boldsymbol{x}|y) \cdot p(y)}{p(\boldsymbol{x})}\right) = \operatorname{argmax}\left(\prod p(x_i|y)p(y)\right)$

Estimation of parameters for given class: independently compute mean and variance for each component, thus drastically reducing the total number of parameters.

(a) Create a new classifier file for the Naive Bayes approach in the toolbox.

(b) Use the simplification (1) of the class dependent probability to estimate the parameters μ_i and σ_i^2 for each class.

(c) Apply the trained parameters in the classification step.

Exercise 2 Implement a new toolbox classifier to solve a 2-class problem that is based on Gaussian class conditional densities (= one Gaussian for each class).

(a) Implement the ML formulas for the parameters of the Gaussian as the training step of the classifier.

Given observed random variable $\boldsymbol{X} = \boldsymbol{x_1}, \boldsymbol{x_2}, ...\boldsymbol{x_n}$, find parameter $\boldsymbol{\Theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that:

$$\begin{aligned}
\hat{\boldsymbol{\Theta}} &= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \quad L(\boldsymbol{X};\boldsymbol{\Theta}) \\
&= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \quad p(\boldsymbol{X};\boldsymbol{\Theta}) \\
&= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \prod_{k=1}^{n} p(\boldsymbol{x}_k;\boldsymbol{\Theta}) \\
&= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{k=1}^{n} log(p(\boldsymbol{x}_k;\boldsymbol{\Theta})) \\
&= \underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \sum_{k=1}^{n} \quad (-\frac{1}{2}(log(|2\pi\boldsymbol{\Sigma})|) - \frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}))
\end{aligned} \tag{2}$$

$$\nabla_{\boldsymbol{\mu}} L(\boldsymbol{\Theta}) = \sum_{k=1}^{n} -\frac{1}{2}(-2\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu})) = 0 \tag{3}$$

$$\nabla_{\boldsymbol{\Sigma}} L(\boldsymbol{\Theta}) = \sum_{k=1}^{n} \frac{1}{n}(\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu})(\boldsymbol{x}_k - \boldsymbol{\mu})^T) \tag{4}$$

So we have:

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{x}_k \tag{5}$$

$$\boldsymbol{\Sigma} = \frac{1}{n}\sum_{k=1}^{n} (\boldsymbol{x}_k - \boldsymbol{\mu}) \cdot (\boldsymbol{x}_k - \boldsymbol{\mu})^T \quad \text{bias} \tag{6}$$

In unbias case, $\boldsymbol{\Sigma} = \frac{1}{n-1}\sum_{k=1}^{n} (\boldsymbol{x}_k - \boldsymbol{\mu}) \cdot (\boldsymbol{x}_k - \boldsymbol{\mu})^T$

MATLAB commands *mean* and *cov*.
Priors are determined by relative frequencies.

(b) **Approach 1**: Derive a formulation of the parametric decision boundary $F(\boldsymbol{x}) = 0$ for the two-class problem. The boundary is given by:

$$p_1\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1) = p_0\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0)$$

The formalism is similar to the theory of logistic regression (cmp. lecture). Implement a classifier based on the decision boundary.

Equations for the solution:

$$\boldsymbol{A} = \frac{1}{2}\left(\boldsymbol{\Sigma}_1^{-1} - \boldsymbol{\Sigma}_0^{-1}\right) \tag{7}$$

$$\boldsymbol{\alpha}^T = \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}_0^{-1} - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1} \tag{8}$$

$$\alpha_0 = \log\frac{p_0}{p_1} + \frac{1}{2}\left(\log\frac{\det(2\pi\boldsymbol{\Sigma}_1)}{\det(2\pi\boldsymbol{\Sigma}_0)} + \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0^T\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right) \tag{9}$$

Determine classification using:

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\alpha}^T\boldsymbol{x} + \alpha_0 < 0$$

(c) **Approach 2**: Implement the parametric Bayesian classification on the basis of the trained parameters (Hint: Compute the posterior probabilities for each class).

(d) Compare the MLE classification results with the Naive Bayes classifiers.