Bayesian Classifier Based Letter Recognition

School of Electronics, Tianjin University of Technology and Education,

Twaha Kabika, Msc. Signal and Information Process, JANUARY, 2013

ABSTRACT: The Main task of this project is to design a Bayesian classifier, which would distinguish between two letters K={'A', 'C'}, and that using only a single measurement: *x* = (sum of pixel intensities in the left half of the image) - (sum of pixel intensities in the right half of the image). This measurement (so-called feature) assigns a real number to each image. In this report the assumption is percentage of letters 'A' and 'C' in a given dataset is known, i.e. we know the *a priori* probabilities p(A) and p(C). Further, Also another assumption is conditional probabilities p(*x*|A) and p(*x*|C) are also known, that is the probabilities that a value *x* is measured if the letter is 'A' or 'C' respectively. Probabilities p(*x*|A) and p(*x*|C) are Gaussian distributions with mean value and variance given for each class. In this problem, the set of decisions D coincides with the set of hidden states K and the loss function W takes just two values: 0 for correctly recognized and 1 for incorrectly recognized letter. In this special case the classificator q expresses as q(x) = argmax_k p(k|x) = argmax_k p(k|x)/p(x) = argmax_k p(k|x) = argmax_k p(k) p(x|k). Where p(k|x) is called *a posteriori* probability of class k given the measurement x. Symbol argmax_k denotes finding k maximizing the argument.

1.1 Definition

The term pattern recognition refers to the task of placing some object to a correct class based on the measurements about the object. Usually this task is to be performed automatically with the help of computer. Objects to be recognized, Measurements about the objects, and possible classes can be almost anything in the world. For this reason, there are very different pattern recognition tasks. A system that makes measurements about certain objects and there after classes these objects is called a pattern recognition system. For example, a bottle recycling machine is a pattern recognition system. The customer inputs his/her bottles (and cans) into the machine, the machine recognizes the bottles, delivers them in proper containers, computes the amount of compensation for the customer and prints a receipt for the customer. A spam (junk-mail) filter is another example of pattern recognition systems. A spam filter recognizes automatically junk e-mails and places them in a different folder (e.g. /dev/null) than the user's inbox.

1.2 Pattern Recognition System

The task of the pattern recognition system is to classify an object into a correct Class based on the measurements about the object. Note that possible classes are usually well defined already before the design of the pattern recognition system. Many pattern recognition systems can be thought to consist of five stages:

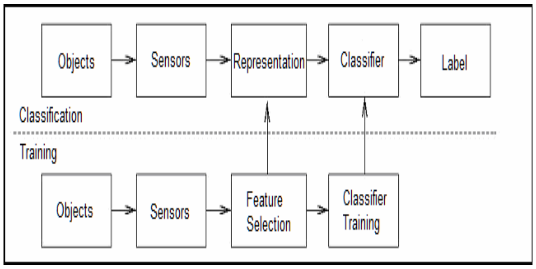
1. Sensing (measurement);

2. Pre-processing and segmentation;

3. Feature extraction;

4. Classification;

5. Post-processing;



1.3 Supervised and Unsupervised Learning and Classification

The classifier component of a pattern recognition system has to be taught to identify certain feature vectors to belong to a certain class. The points here are that

1) It is impossible to define the correct classes for all the possible feature vectors 1 and

2) The purpose of the system is to assign an object which it has not seen previously to a correct class

It is important to distinguish between two types of machine learning when considering the pattern recognition systems. The (main) learning types are supervised learning and unsupervised learning. We also use terms supervised classification and unsupervised classification

In supervised classification, we present examples of the correct classification (a feature vector along with its correct class) to teach a classifier. Based on these examples, that are sometimes termed prototypes or training samples, the classifier then learns how to assign an unseen feature vector to a correct class. The generation of the prototypes (i.e. the classification of feature vectors/objects they represent) has to be done manually in most cases. This can mean lot of work: After all, it was because we wanted to avoid the hand-labeling of objects that we decided to design a pattern recognition system in the first place. That is why the number of prototypes is usually very small compared to the number of possible inputs received by the pattern recognition system. Based on these examples we would have to deduct the class of a never seen object. Therefore, the classifier design must be based on the assumptions made about the classification problem in addition to prototypes used to teach the classifier. These assumptions can often be described best in the language of probability theory.

In unsupervised classification or clustering, there is neither explicit teacher nor training samples. The classification of the feature vectors must be based on similarity between them based on which they are divided into natural groupings. Whether any two feature vectors are similar depends on the application. Obviously, unsupervised classification is a more difficult problem than supervised classification and supervised classification is the preferable option if it is possible. In some cases, however, it is necessary to resort to unsupervised learning. For example, this is the case if the feature vector describing an object can be expected to change with time.

There exists a third type of learning: reinforcement learning. In this learning type, the teacher does not provide the correct classes for feature vectors but the teacher provides feedback whether the classification was correct or not. In OCR, assume that the correct class for a feature vector is 'R'. The classifier places it into the class 'B'. In reinforcement learning, the feedback would be that 'the classification is incorrect'. However the feedback does not include any information about the correct class.

2.0 Bayesian Decision Theory

The Bayesian decision theory offers a solid foundation for classifier design. It tells us how to design the optimal classifier when the statistical properties of the classification problem are known. The theory is a formalization of some common-sense principles, but it offers a good basis for classifier design.

2.1Classification Error

The classification problem is by nature statistical. Objects having same feature vectors can belong to different classes. Therefore, we cannot assume that it would be possible to derive a classifier that works perfectly. (The perfect classifier would assign every object to the correct class). This is why we must study the classification error: For a specified classification problem our task to derive classifier that makes as few errors (misclassifications) as possible. We denote the classification error by the decision rule α by E (α). Because the classifier α assigns an object to a class only based on the feature vector of that object, we study the probability E(α(x)|x) that objects with the feature vector x are misclassified.

Let us take a closer look into the classification error and show that it can be computed if the preliminary assumptions of the previous section hold. From the item 1 in Theorem 1, it follows that

$$E\ (\alpha(x)|x) = 1 - P\ (\alpha(x)|x),$$

Where P (α(x)|x) is the probability that the class α(x) is correct for the object. The classification error for the classifier α can be written as

$$E(\alpha) = \int_F E(\alpha(x)|x)p(x)dx = \int_F [1 - P(\alpha(x)|x)]p(x)dx.$$

We need to still know P (α(x)|x). From the Bayes rule we get

$$P(\alpha(\mathrm{x})|\mathrm{x}) = \frac{p(\mathrm{x}|\alpha(\mathrm{x}))P(\alpha(\mathrm{x}))}{p(\mathrm{x})}.$$

And hence the classification error is

$$E(\alpha) = 1 - \int_{\mathbb{F}} p(\mathrm{x}|\alpha(\mathrm{x}))P(\alpha(\mathrm{x}))d\mathrm{x},$$

That is, the classification error of α is equal to the probability of the complement of the event f{(x, α(x)) : x ∈ F}.

With the knowledge of decision regions, we can rewrite the classification error as

$$E(\alpha) = \sum_{i=1}^{c} \int_{R_i} [1 - p(\mathrm{x}|\omega_i)P(\omega_i)]d\mathrm{x} = 1 - \sum_{i=1}^{c} \int_{R_i} p(\mathrm{x}|\omega_i)P(\omega_i)d\mathrm{x},$$

Where R_i, i = 1, . . . , c are decision regions for the decision rule α.

2.2 Bayes Minimum Error Classifier

In this section, we study Bayes minimum error classifier which, provided that the assumptions of section 2.1 hold, minimizes the classification error. (The assumptions were that class conditional pdfs and prior probabilities are known.) This means that Bayes minimum error classifier, later on just Bayes classifier, is optimal for a given classification problem.

The Bayes classifier is defined as

$$\alpha_{Bayes}(\mathrm{x}) = \arg \max_{\omega_i, i=1,\dots,c} p(\mathrm{x}|\omega_i)P(\omega_i).$$

If two or more classes have the same posterior probability given x, we can freely choose between them. In practice, the Bayes classification of x is performed by computing p(x|ωi)P(ωi) for each class ωi; i = 1; : : : ; c and assigning x to the class with the maximum p(x|ωi)P(ωi).

By its definition the Bayes classifier minimizes the conditional error E(α(x)|x) =1-P(α(x)|x) for all x. Because of this and basic properties of integrals, the Bayes classifier minimize also the classification error:

Note: that the definition of the Bayes classifier does not require the assumption that the class conditional pdfs are Gaussian distributed. The class conditional pdfs can be any proper pdfs.

2.3 Bayes Minimum Risk Classifier

The Bayes minimum error classifier is a special case of the more general Bayes minimum risk classifier. In addition to the assumptions and terminology introduced previously, we are given a actions $\alpha_1 . . . \alpha_a$. These correspond to the actions that follow the classification. For example, a bottle recycling machine, after classifying the bottles and cans, places them in the correct containers and returns a receipt to the customer. The actions are tied to the classes via a loss function λ. The value $\lambda(\alpha_i|\omega j)$ quantifies the loss incurred by taking an action α_i when the true class is ω_j. The decision rules are now functions from the feature space onto the set of actions. We need still one more definition. The conditional risk of taking the action α_i when the observed feature vector is x is defined as

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i|\omega_j) P(\omega_j|\mathbf{x}).$$

The definition of the Bayes minimum risk classifier is then:

The Bayes minimum risk classifier chooses the action with the minimum conditional risk.

The Bayes minimum risk classifier is the optimal in this more general setting: it is the decision rule that minimizes the total risk given by

$$R_{total}(\alpha) = \int R(\alpha(\mathbf{x})|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}.$$

The Bayes classifier of the previous section is obtained when the action α_i is the classification to the class ω_j and the loss function is zero-one loss:

$$\lambda(\alpha_i|\omega_j) = \begin{cases} 0 & \text{if} \quad i = j \\ 1 & \text{if} \quad i \neq j \end{cases}$$

The number of actions a can be different from the number of classes c. This is useful e.g. when it is preferable that the pattern recognition system is able to separate those feature vectors that cannot be reliably classified.

3.0 CLASIFICATION OF THE PROBLEM

The task is to design a classifier, which would distinguish between two letters K={'A', 'C'}, and that using only a single measurement:

x = (sum of pixel intensities in the left half of the image) - (sum of pixel intensities in the right half of the image)

This measurement (so-called feature) assigns a real number to each image. Let us assume that the percentage of letters 'A' and 'C' in a given dataset is known, i.e. we know the *a priori* probabilities p(A) and p(C). Further, let us assume that the conditional probabilities p(*x*|A) and p(*x*|C) are also known, that is the probabilities that a value *x* is measured if the letter is 'A' or 'C' respectively. Let probabilities p(*x*|A) and p(*x*|C) be Gaussian distributions with mean value and variance given for each class.

In this problem, the set of decisions D coincides with the set of hidden states K and the loss function W takes just two values: 0 for correctly recognized and 1 for incorrectly recognized letter. In this special case the classificator q expresses as

q(x) = argmax_k p(k|x) = argmax_k p(k|x)/p(x) = argmax_k p(k|x) = argmax_k p(k) p(x|k),

Where p(k|x) is called *a posteriori* probability of class k given the measurement x. Symbol argmax_k denotes finding k maximizing the argument.

3.1 EXPLANATION OF WHAT THE PROGRAM DOES:

1. For given values p(A), p(C) and parameters of p(*x*|A), p(*x*|C), it compute the coefficients of the quadratic discriminative function (obtained from the MATLAB file given). It evaluates the discriminative function for each image, and classifies it as either 'A' or 'C'. Display the classification

results, i.e. show separately images classified as 'A' and as 'C'.

2. Function [risk,epsA,epsC,interA] = bayeserror(D1,D2), which returns the risk of the Bayesian strategy and the classification thresholds. The inputs are the structures D1 and D2 with the parameters of p(*x*|A), p(*x*|C), p(A) and p(C).
3. In order to have the correct classification of images (contained in the array labels), it compute the actual classification errors for the given dataset. Compare the errors with the output of your bayeserror function.
4. By using the pgmm function from the stprtool toolbox, the second figure displays the probability densities and the classification thresholds t_1, t_2.
5. In a first figure, show empirical estimates of p(x|A) and p(x|B) computed from data. Use functions hist and bar. To obtain empirical estimates for distributions of continuous variable (they must integrate to 1) normalize the histograms as appropriate.

3.2 RESULTS:

BayesError:
risk:0.127723
epsA:1.520766e-001
epsC:8.713456e-002
chyba: 11.875000%

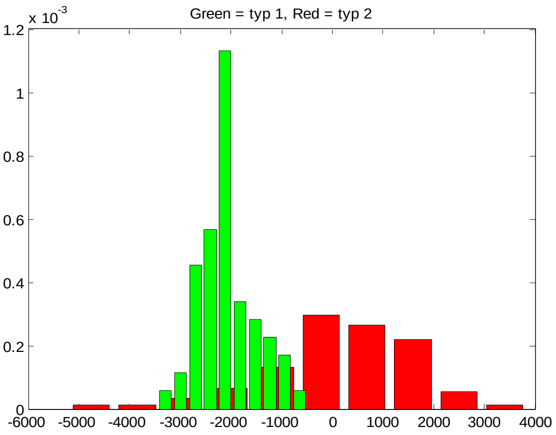


Fig3.0 **Shows empirical estimates of p(x|A) and p(x|B)**

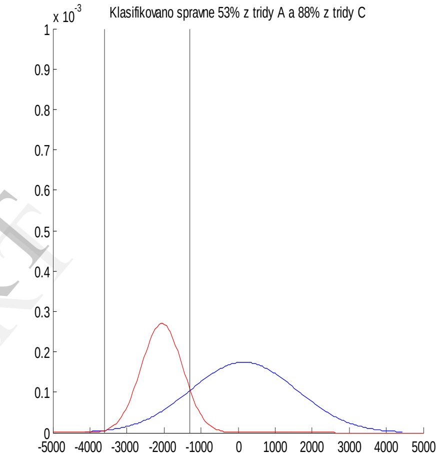


Fig3.1. Pdf and threshold **Computed from data**

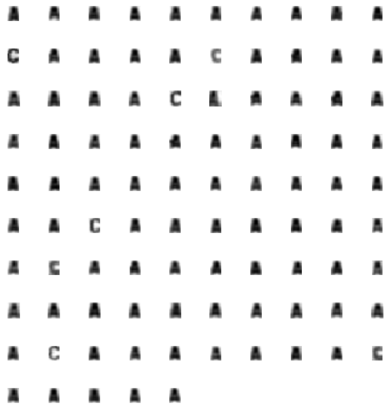


Image classified as 'A'

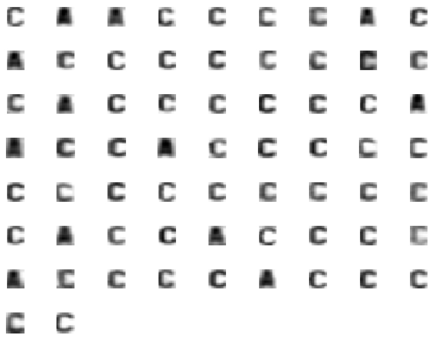


Image classified as 'C '

4.0 C0NCLUSION

Bayesian classifier (DensityBased classifier) is designed successfully for A and C letter as explained above. But more modification can be done so that the classifier will be able to recognize and differentiate all the letters from A to z.MATLAB codes are available in the main report .

5.0 REFERENCES

- Duda R., Hart P., Stock D.: Pattern Classification, 2001
- Michail I. Schlesinger, Vaclav Hlavac. Ten Lectures on Statistical and Structural Pattern Recognition. Kluwer Academic Publishers, 2002.
- slides from lecture by Vaclav Hlavac and Jiri Matas.
- Apt, K. R. & Bezem, M. (1991). Acyclic programs, *New Generation Computing* **9**(3-4): 335–363.
- Bacchus, F., Halpern, J.Y. & Levesque, H. J. (1999). Reasoning about noisy sensors and effectors in the situation calculus, *Artificial Intelligence* **111**(1–2): 171–208. **URL:** *http://www.lpaig.uwaterloo.ca/˜fbacchus/on-line.html*
- Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, Massachusetts. Two volumes.
- Boutilier, C., Dean, T. & Hanks, S. (1999). Decision-theoretic planning: Structual assumptions and computational leverage, *Journal of Artificial Intelligence Research* **11**: 1–94.
- Boutilier, C., Dearden, R. & Goldszmidt, M. (1995). Exploiting structure in policy construction, *Proc. 14th International Joint Conf. on Artificial Intelligence (IJCAI-95)*, Montréal, Québec, pp. 1104–1111.
- Boutilier, C., Friedman, N., Goldszmidt, M. & Koller, D. (1996). Context-specific independence in Bayesian networks, *in* E. Horvitz & F. Jensen (eds), *Proc. Twelfth Conf. on Uncertainty in Artificial Intelligence (UAI-96)*, Portland, OR, pp. 115–123.
- Buntine,W. L. (1994). Operations for learning with graphical models, *Journal of Artificial Intelegence research 2:199-254.*
- *Anil K. Jain, Robert P.W. Duin-* **Introduction to Pattern Recognition**
- R.L. Gregory (eds.), *The Oxford Companion to the Mind, Second Edition*, Oxford University Press, Oxford, UK, 2004, 698-703.
- Duda, R.O., Hart, P.E. and Stork, D.G. (2001), *Pattern Classification and Scene Analysis*, 2nd ed. Wiley.
- Perlovsky, L.I. (1998), 'Conundrum of combinatorial complexity'. *IEEE Trans. on Pattern Analysis and MachineIntelligence*, vol. 20, no. 6.