Pattern Recognition with Applications to Biomedical Images
Independent Study in Mathematics – CSUN Spring 2006
Module 2–Classification 'by hand'

P. Perona & K. F. Stevenson

January 22, 2007

Exercise 0.1. *In order to build intuition carry out the following excercise:*

Feature selection and classifier training - *Two students download the '4 Categories' training images from the 'Data' page of the Wiki. They look at the training set of each category and produce a recipe for classifying the images. The recipe consists of 2 steps: (A - Feature selection) They each pick 2 'features' that seem useful in discriminating the images, e.g. 'roundness' could be one such feature. (B - Classification) They specify how the features they selected should be used to classify each class: eg. "class C1 is almost perfectly round, while the other classes are not". These rules will be written down and passed on to the third student, along with a sheet on which pictures from the test set are shown in random order, without indicating the class to which each belongs.*

Classification - *The third student uses the 'rules' produced in the previous step to classify the given test set of images.*

Error analysis - *What are the possible types of errors? Are all types of errors equal?*

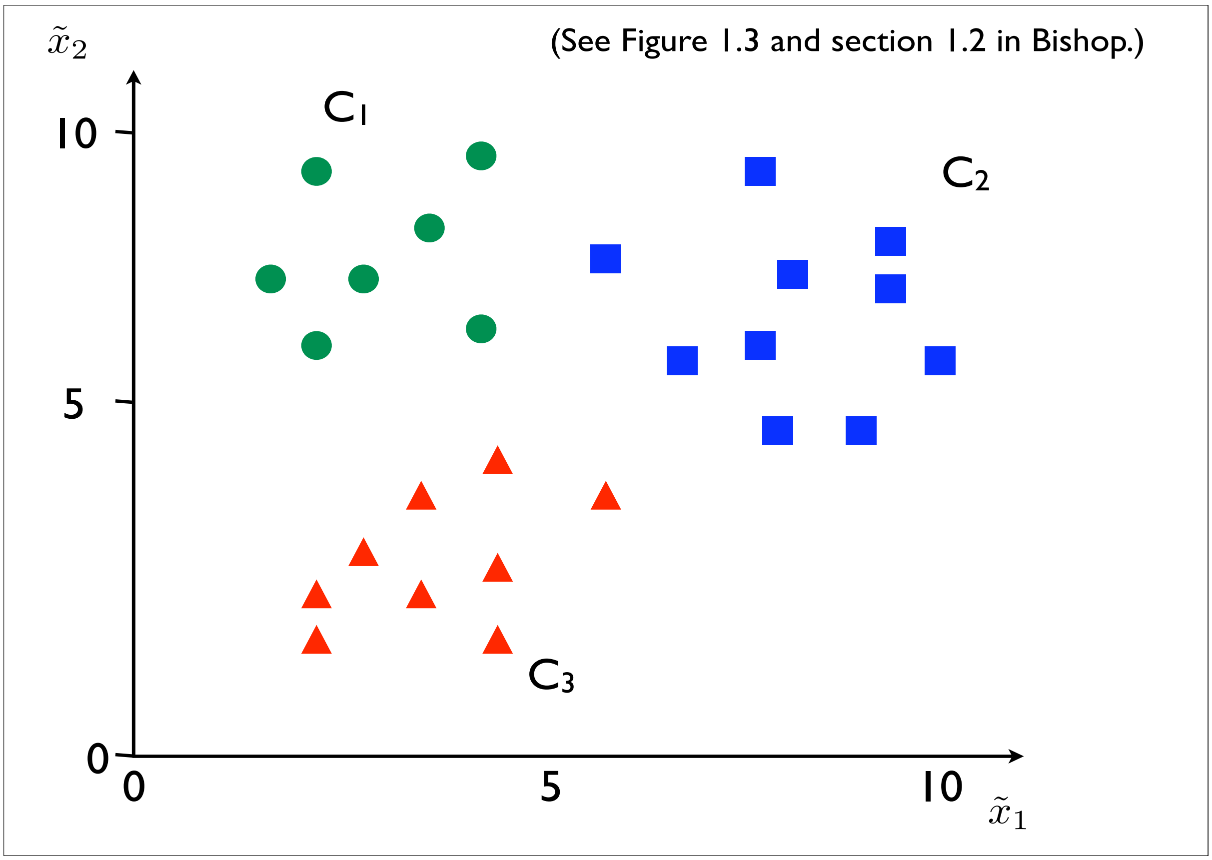
Improve your classifier - *In the light of the errors made, make improved rules list. Ask a friend to use the new rules to classify the test set. See if you did better.*

Now read sections 1.1 thru 1.3 in Bishop. The appendix to this document provides some help in understanding those concepts better. After you have read and understood those concepts carry out the following excercise:

Exercise 0.2. *Working out variant on sections 1.2 and 1.3 in Bishop.*

1. *Take Figure 1. After reading Section 1.2 draw reasonable linear decision boundaries (how many should there be?).*

2. *Give approximate equations for these lines. Your equations should have the form:* $z_i = a_{i1} \cdot \tilde{x}_1 + a_{i2} \cdot \tilde{x}_2 + a_{i3} = 0$. *I.e. you need to specify the* $a_{i,j}$.



$\tilde{x}_2$
(See Figure 1.3 and section 1.2 in Bishop.)
10
5
0
C_1
C_2
C_3
0
5
10
$\tilde{x}_1$

Figure 1:

3. *How are these related to the* w_k*?*

4. *Define appropriate functions* $y_k = f(\tilde{\mathrm{x}}, w_k)$ *which make use of the* z_i *and which map a point in the feature space* $\tilde{\mathrm{x}} = [\tilde{x}_1, \tilde{x}_2]$ *to an appropriate class label* (y_1, y_2, y_3)*. Hint 1: consider the Heaviside step function computed on the (signed) distance of each point from the decision boundary. Hint 2: observe that if you scale the* a_{ij} *parameters appropriately, such a distance is very easy to compute.*

A Appendix on Classifiers and Training Sets

In Figures 2, 3, 4, 5 you will find some diagrams explaining the ideas in Bishop (Section 1.1-1.3) in terms of functions and vector spaces, and to give some more examples. You should look at these notes in conjunction with reading Bishop's book and working on a MatLab implementation.

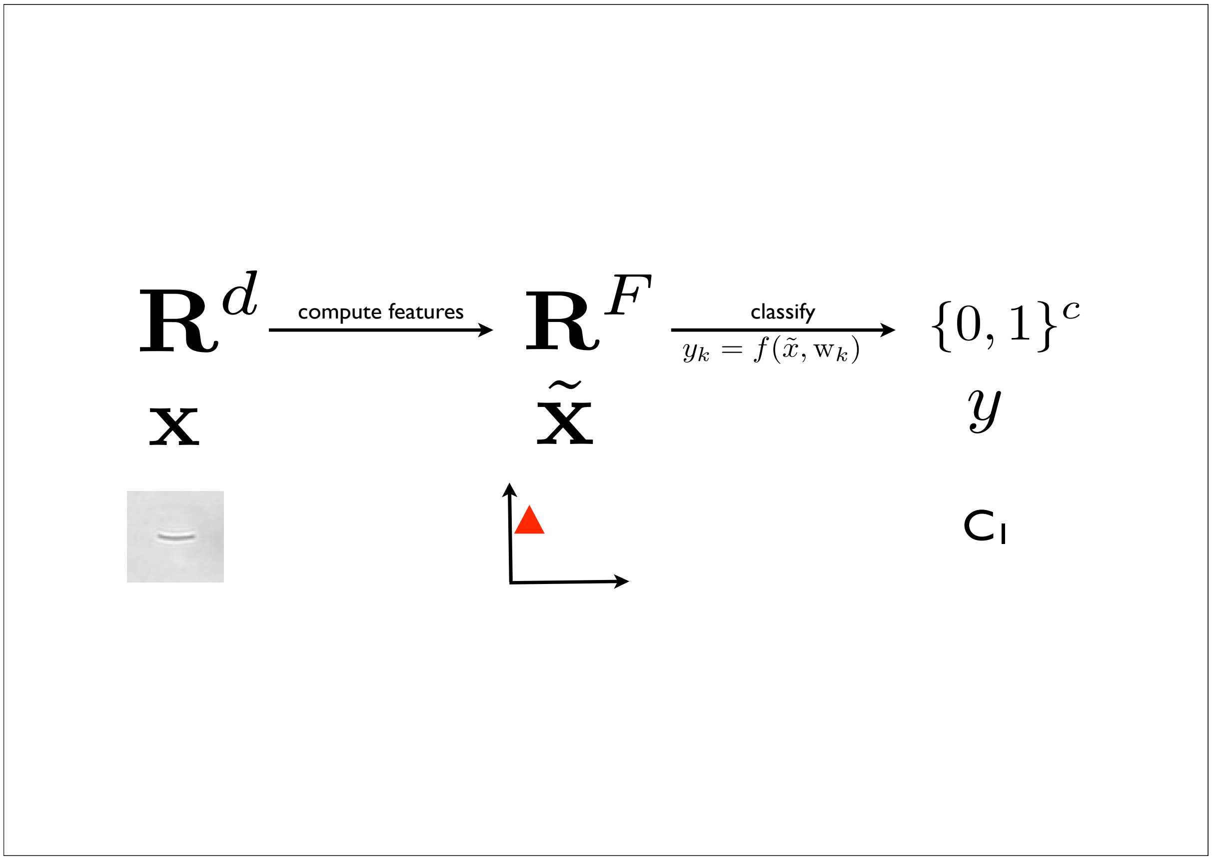


Figure 2: A classifier works in two steps: feature extraction (from image to features), followed by classification (from features to classes). The Image is the raw data. In our case this data is the image of a particle that was found in urine, it may be thought of as a very long vector whose components represent the intensity of light at each pixel in the image. Thus our images are vectors in $\mathbf{R}^d$ where d is HUGE. See the section "The curse of dimensionality" in Bishop's book to understand why this is a problem. In order to reduce the dimension of the space in which our data live, we compute a few features, each feature is a (more or less arbitrary) scalar function of the image vector. Thus each image vector is mapped to a feature vector in some $\mathbf{R}^F$ with $F << d$. If we chose our features cleverly, the vectors $\tilde{x}_i$ group naturally into clusters corresponding to the desired classes: e.g. Bacteria, Blood cell, crystal. If this is true, then it will be easy to select 'classifier functions' $y(\tilde{x})$ that map each $\tilde{x}$ into its appropriate class C. Another way to think about this last step is to select many functions, $y_k(\tilde{x})$ each one of which outputs a '1' when its input $\tilde{x}$ belongs to the corresponding class C_k, and '0' otherwise. This process of selecting good 'classifier functions' is called 'Classifier Training' and is discussed in the next diagram.

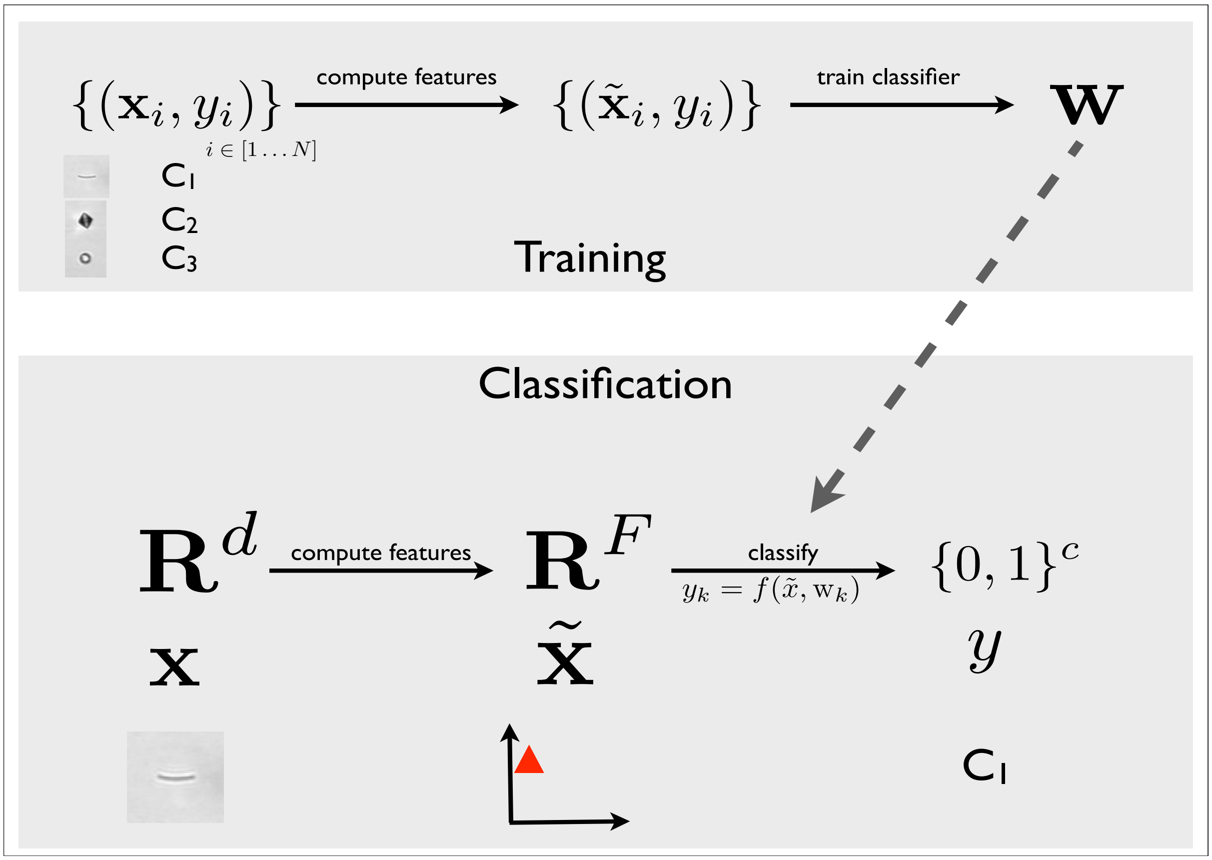


Figure 3: In order to select the appropriate 'classifier functions' we need to work with a subset of our images where we DO know their classifications: The **training set**. Once our classifier is trained, i.e. parameters [**w**] have been selected, we may use it to classify images for which a class is not known, the **test set**. If, as a test set, we use a set of images for which the class is known (but we hide that information from the classifier), we can compare the 'guesses' made by our classifier with the ground truth and measure that classifier's performance. Once we are satisfied that our classifier works well enough we go on to apply the classifier function on data for which the class is truly not known.

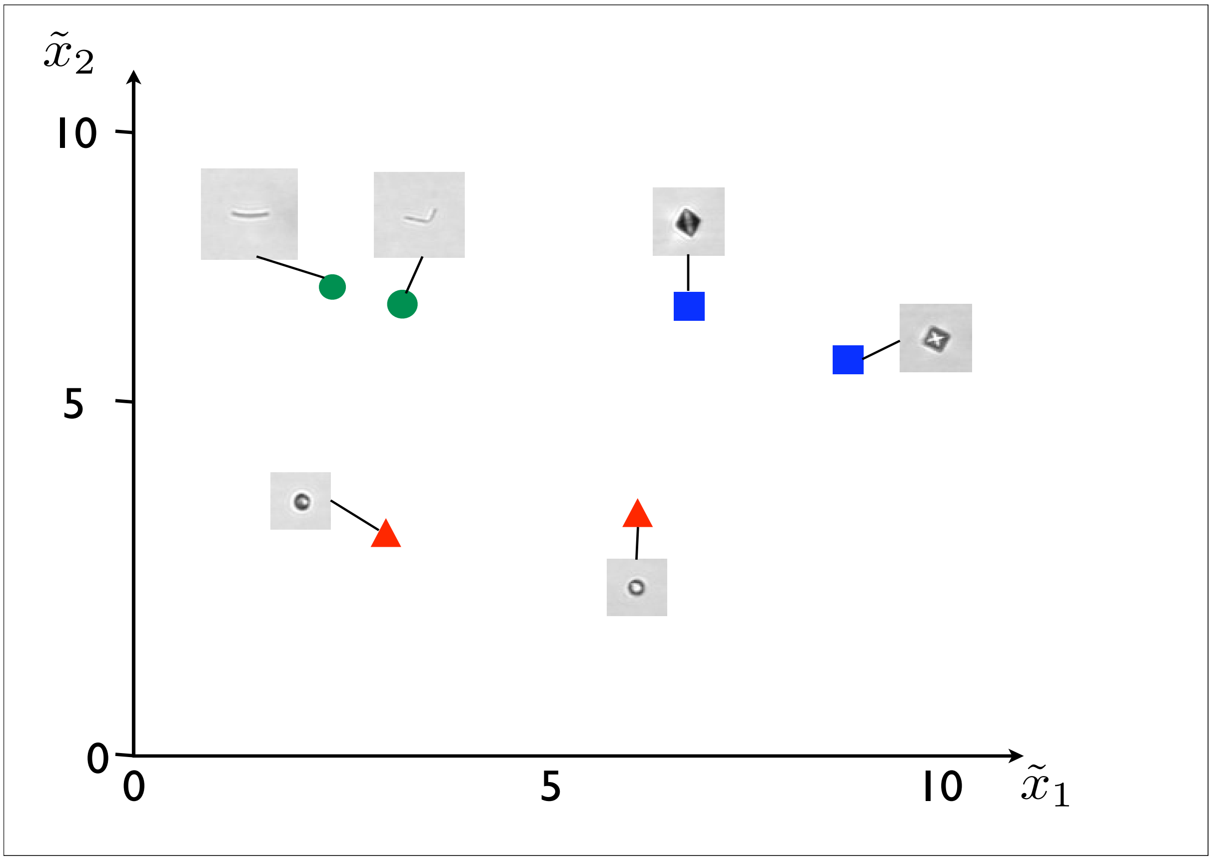


Figure 4: This diagram plots the features associated to a number of images. Two features were chosen, $\tilde{x}_1$ and $\tilde{x}_2$ – it does not matter here what they really were (see the code in `explore_features.m` for some practical examples). Each of the images is mapped by the feature function to a point in this space. Here we are supposing that there are three classes of objects (C_1, C_2, C_3) and two feature begin "measured" on each image $(\tilde{x}_1, \tilde{x}_2)$. Thus we end up with three clusters of points in $\mathbf{R}^2$.

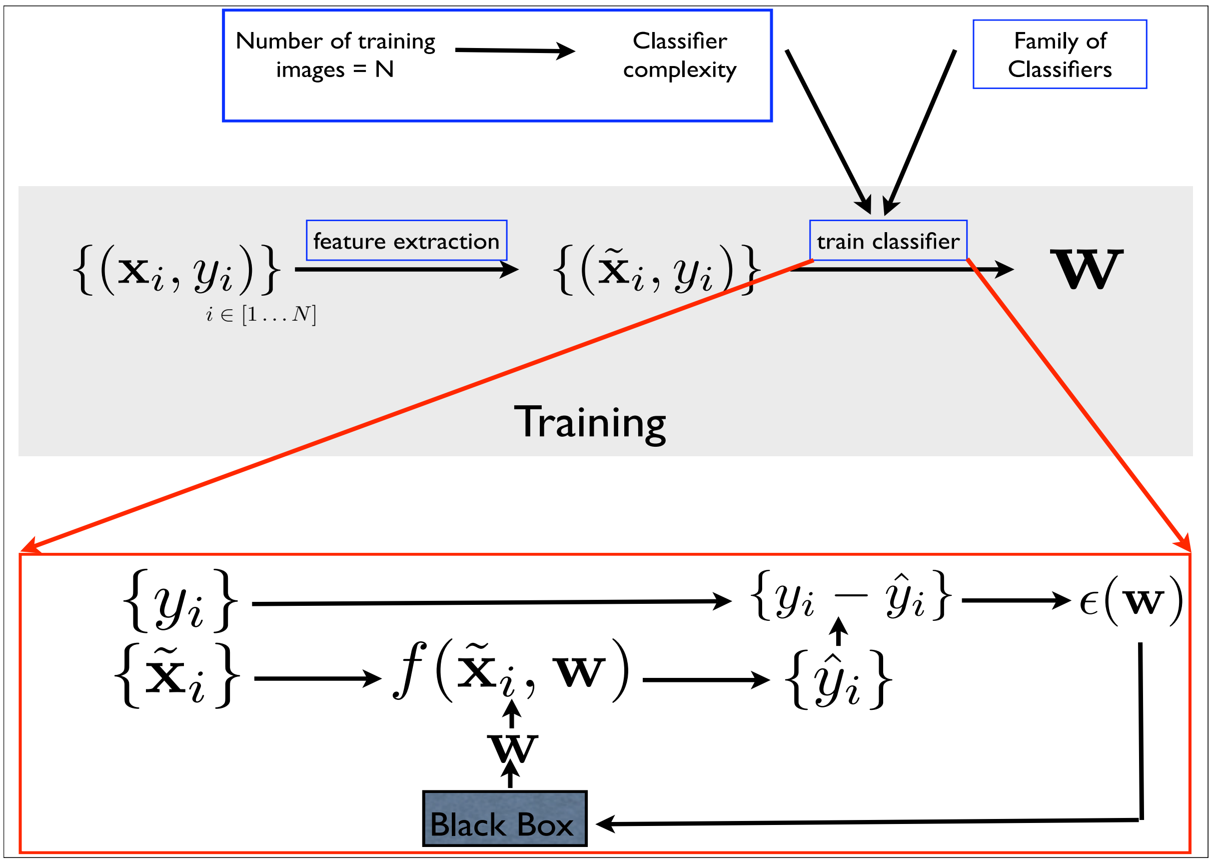


Figure 5: This is a pictorial demonstration of how the different "modules" of the class fit together within the structure of the training phase.