

# Supervised Learning and Bayesian Classification

Erik G. Learned-Miller  
Department of Computer Science  
University of Massachusetts, Amherst  
Amherst, MA 01003

September 12, 2011

## **Abstract**

This document discusses Bayesian classification in the context of supervised learning. Supervised learning is defined. An approach is described in which feature likelihoods are estimated from data, and then classification is done by computing class posteriors given features using Bayes rule. Estimating of feature likelihoods, independence of features, quantization of features, and information content of features are discussed.

# 1 Supervised learning

Supervised learning is simply a formalization of the idea of learning from examples. In supervised learning, the learner (typically, a computer program) is provided with two sets of data, a *training set* and a *test set*. The idea is for the learner to “learn” from a set of labeled examples in the training set so that it can identify unlabeled examples in the test set with the highest possible accuracy. For example, a training set might consist of images of different types of fruit (say, peaches and nectarines), where the identity of the fruit in each image is given to the learner. The test set would then consist of more unidentified pieces of fruit, but from the same classes. The goal is for the learner to identify the elements in the test set. There are many different approaches which attempt to build the best possible method of classifying examples of the test set by using the data given in the training set. We will discuss a few of these in this document, after defining supervised learning more formally.

*supervised  
learning  
training set  
test set*

In supervised learning, the training set consists of a set of  $n$  ordered pairs  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)$ , where each  $\mathbf{x}_i$  is some measurement or set of measurements of a single example data point, and  $y_i$  is the label for that data point. For example, an  $\mathbf{x}_i$  might be a vector of 5 measurements for a patient in a hospital including height, weight, temperature, blood sugar level, and blood pressure. The corresponding  $y_i$  might be a classification of the patient as “healthy” or “not healthy”.

The test data in supervised learning is another set of  $m$  measurements *without* labels:  $(\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \dots, \mathbf{x}_{n+m})$ . As described above, the goal is to make educated guesses about the labels for the test set (such as “healthy” or “not healthy”) by drawing inferences from the training set.

## 1.1 Common assumptions about training and test sets

To have any utility, the training data and test data for a given supervised learning problem should have *some* relationship to each other. Let us assume, for the moment, that the *test data* are drawn from a distribution  $p(\mathbf{x}, y)$ . If the training data are drawn from some probability distribution  $q(\mathbf{x}, y)$  that is very different from the distribution of the test data, then we should have no expectation that inferences we make from the training data will help us classify elements of the test set. So, in most cases, people only apply supervised learning methods if they have some expectation that there is some useful relationship between the distribution of the training data and the distribution of the test data. We now discuss two common scenarios in which data sets are generated for supervised learning. In both cases, we shall assume that the *test data* are drawn from a distribution  $p(\mathbf{x}, y)$ .

**Case 1.** In this case, the training data are also drawn from  $p(\mathbf{x}, y)$ . That is, the training and test data are drawn from the same distribution. Theoretically, with enough data, we should be able to estimate from the training data any conditional or marginal distribution of the training distribution, and hence of the test distribution, that is derivable from the distribution  $p(\mathbf{x}, y)$ . That is,

we should be able to estimate  $p(y|\mathbf{x})$  directly for any  $y$  and  $\mathbf{x}$ . If we estimate these values perfectly, we should be able to build a Bayes optimal classifier with minimum expected error. Alternatively, we could estimate the likelihoods  $p(\mathbf{x}|y)$  and the priors  $p(y)$ , and then use Bayes' rule to obtain the posteriors. In the limit of infinite data, both of these methods work perfectly. With finite data, it is not always clear which method is preferable.

**Case 2.** In this case, the training data are not drawn directly from  $p(\mathbf{x}, y)$ . Rather, we draw a separate sample of each class by drawing samples from  $p(\mathbf{x}|y)$ . That is, we fix  $y$  to be the first class and draw some samples of  $\mathbf{x}$  for the first class. Then we sample some values of  $\mathbf{x}$  for the second class, and so on.

Why would we ever do this instead of drawing from  $p(\mathbf{x}, y)$ ? Consider trying to build a classifier for a rare disease. We would like to have a sample of patients with the disease and a sample of healthy individuals so that we could build a classifier to distinguish the two. At test time, we will be picking random samples from the general population and trying to decide if they have the disease. Hence, the "test" data  $p(\mathbf{x}, y)$  are represented by the general population. Now suppose that patients with the disease only occur in 1 out of 100,000 people in the general population. If we sample people randomly from  $p(\mathbf{x}, y)$ , we will have to sample a million people before we expect to have even 10 examples of subjects with the rare disease. Hence, it is very impractical to sample from the joint distribution. Instead, we would like to visit a specialist physician who treats the disease and obtain a large sample of the disease population, measuring their symptoms. Sampling this population  $p(\mathbf{x}|y = \text{has rare disease})$  separately from the healthy population is clearly far more efficient than sampling from the distribution  $p(\mathbf{x}, y)$ .

Note that in Case 1, we may estimate the priors  $p(y)$  directly from the samples, but in Case 2, we must have a separate method for estimating the priors, since our data do not reflect  $p(y)$  in any way. Often, people use another data source for the prior, or simply make an educated guess.

## 2 Supervised learning and Bayes rule

Let the set of  $k$  possible class labels be denoted  $\mathcal{Y} = \{c_1, c_2, \dots, c_k\}$ . Suppose that, for each class in our classification problem, we are given the conditional probability of the class given an observation. That is, we are given  $p(y|\mathbf{x})$  for all possible values of  $y$  and  $\mathbf{x}$ . Recall that in this scenario, the *Bayes optimal classifier*, i.e. the classifier that minimizes the expected *probability of error* for an observation  $\mathbf{x}$  is

$$\arg \max_{y \in \mathcal{Y}} p(y|\mathbf{x}).$$

What exactly does it mean to minimize the probability of error? Assume that a classifier is a deterministic function  $f(\mathbf{x})$  which returns a class label for any data vector  $\mathbf{x}$ . Then given a *joint distribution* of data vectors and labels,  $p(\mathbf{x}, y)$ , we

*probability of error*

can write the the probability of error as

$$Prob(Error) = \sum_{\mathbf{x} \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(\mathbf{x}, y) I(y \neq f(\mathbf{x})), \quad (1)$$

where  $I(exp)$  evaluates to 1 if expression  $exp$  is true and to 0 otherwise. No function  $f(\mathbf{x})$  has lower probability of error than the Bayes optimal classifier.

## 2.1 Estimating $p(y|\mathbf{x})$

Now if we are not given the values of the so-called *posterior probabilities*  $p(y|\mathbf{x})$ , we may estimate these values from the training data. There are many ways to do this, and there is continuing debate among statisticians and machine learning practitioners about what methods are best under various circumstances for estimating posteriors. For the moment, we will avoid the discussion of which method is best, and simply present some commonly used methods of posterior estimation.

One method for estimating the posteriors is to first estimate the *class likelihoods*

$$p(\mathbf{x}|y)$$

and the *priors*

$$p(y)$$

from the training data, and then to use Bayes' rule to compute an estimated value of the posterior. If  $\hat{p}(\mathbf{x}|y)$  is our estimate of the class likelihoods and  $\hat{p}(y)$  is our estimate of the prior, then

$$\hat{p}(y|\mathbf{x}) = \frac{\hat{p}(\mathbf{x}|y)\hat{p}(y)}{\sum_{z \in \mathcal{Y}} \hat{p}(\mathbf{x}|z)\hat{p}(z)}.$$

As mentioned above, when our training data is drawn from conditional distributions of the form  $p(\mathbf{x}|y)$  (Case 2) rather than from the same distribution  $p(\mathbf{x}, y)$  as the test data (Case 1), we cannot estimate the priors from the training data, but need to obtain a prior from somewhere else. (One commonly used method is to assume the prior probabilities  $p(y)$  are all equal, and hence equal to  $\frac{1}{k}$ , where  $k$  is the number of classes.)

## 2.2 Estimating likelihoods from training data

Just as there are many ways of estimating posteriors, there are many ways of estimating likelihoods from training data. For a full discussion, see any statistical textbook on the topic of estimation. We shall start with one of the simplest estimators of the likelihood, which is just the frequency of  $\mathbf{x}$  in the training set given a particular value of  $y$ . That is,

$$\hat{p}(\mathbf{x}|y = c) = \frac{N(\mathbf{x}, y = c)}{N(y = c)},$$

where  $N(\mathbf{x}, y = c)$  is the number of training points with label  $c$  and observation equal to  $\mathbf{x}$  and  $N(y = c)$  is simply the number of training points whose label is  $c$ .

**Example 1.** Suppose we wish to have a procedure to decide whether a patient has cancer based upon the outcome of a certain procedure which tests for the presence of a particular antibody in the blood stream. For training data, suppose we perform the test on 100 cancer patients and 200 healthy subjects. Assume that 90 of the cancer patients tested positive for the antibody and 40 of the healthy subjects tested positive. Then we could estimate the following likelihoods.

1.  $\hat{p}(\text{test} = \text{positive} | \text{cancer}) = \frac{90}{100} = 0.9$ .
2.  $\hat{p}(\text{test} = \text{negative} | \text{cancer}) = \frac{10}{100} = 0.1$ .
3.  $\hat{p}(\text{test} = \text{positive} | \text{no cancer}) = \frac{40}{200} = 0.2$ .
4.  $\hat{p}(\text{test} = \text{negative} | \text{no cancer}) = \frac{160}{200} = 0.8$ .

Should we assume that the subjects were drawn from the general population and that the prior probability of having cancer is  $\frac{100}{300} = \frac{1}{3}$ ? Probably not. More likely, it makes sense to establish the prior from another study in which the rate of cancer in the general population is estimated. Let's say this rate is estimated at  $\hat{p}(\text{cancer}) = 0.001$ , i.e., about 1 in a thousand people have cancer.

Then Bayes' rule allows us to combine the likelihoods and the priors to obtain

$$\hat{p}(\text{cancer} | \text{test} = \text{positive}) = \frac{0.9 \times 0.001}{0.9 \times 0.001 + 0.2 \times 0.999} = \frac{0.0009}{0.2007} \approx 0.005. \quad (2)$$

Notice the result, which some people find quite surprising. Even testing positive, while the chance of having cancer has approximately doubled, the chance of having cancer is still small (0.005). This, of course, is because of the extremely low prior probability of having cancer, and the fact that the test is not very discriminative. Strangely, enough, this means we would *never* classify a patient as having cancer using Bayes' rule and this particular test, if our goal was to minimize error. Finally, it is important to note that in medical diagnosis, the goal may not be to strictly minimize the number of errors, since one type of error (declaring a sick patient healthy) may be far more costly than another type of error (declaring a healthy patient sick). Since errors do not have the same cost, the doctor may want to skew the decision to reduce the number of costly errors even if the total error rate goes up. This is the subject of *decision theory*, and we will defer our discussion of it for the moment.

### 2.3 Two, three, and more features

In the previous section, we estimated the probability that a patient had cancer based upon a single feature, the result of one test. Of course, in many classification problems, we will have more than one piece of information. We shall use

*measurement, feature, or data component* as synonyms for a piece of information in the following discussions.

**Example 2.** Suppose one is a botanist trying to develop a field test for the identification of a particular plant species  $A$  to distinguish it from another similar looking species  $B$ . You've noticed that the height of the plant and whether it has brown spots on the leaves are helpful in making the identification, although they do not allow the identification with certainty. (To be certain, you have to send a small sample of a leaf back to the lab for chemical testing.)

After doing a large survey of plants randomly sampled from the relevant region, you have gathered the following data for plant species  $A$ :

Species $A$	0-1 ft.	1-2ft.	2-3ft.	3-4ft.
Brown spots	100	204	300	392
No brown spots	40	79	130	150

and for species  $B$ :

Species $B$	0-1 ft.	1-2ft.	2-3ft.	3-4ft.
Brown spots	105	310	300	85
No brown spots	9	32	28	11

Given this data, we can proceed as when we had one feature, by simply treating every likelihood as a separate probability, and estimating eight likelihoods for each class. However, note that as we have more features with more total likelihoods to estimate, we generally need more data to estimate each one accurately. This problem becomes particularly severe when doing classification of images with thousands of measurements per image. There are two potential ways to mitigate the problem of having too many likelihoods to estimate. One is to reduce the number of values associated with a feature. Another is to use the independence, or near-independence of features.

## 2.4 Binning features

In the above example, we have measured the height of a plant and placed it into one of four bins based upon that height. If we chose, we could reduce the number of options further by, say, categorizing a plant as having a height from 0-2 feet, or from 2-4 feet. Doing this reduces the information we have about a specific plant. However, it may allow us to estimate probabilities more accurately. Choosing the right number of categories for a continuous or discrete feature is not always easy, and is an area of active research. There is a trade-off between estimation accuracy and discriminability which can be difficult to get right.

## 2.5 Feature independence

Suppose we are told by a professional botanist that for plant species  $A$  in the above example, the chance of encountering a plant with brown spots is in no way related to the size of the plant. That is, these features of plant species  $A$  are *statistically independent*. This can be written formally in the following way:

*statistically independent*

$$p(\text{spots}, \text{height}|A) = p(\text{spots}|A) \times p(\text{height}|A).$$

Furthermore, the botanist tells us this is also true of plant species  $B$ . Given this new information, we can now estimate the height likelihoods  $p(\text{height}|A)$  and  $p(\text{height}|B)$  separately from the the spots likelihoods. The reason this is beneficial is because we now have more examples to estimate each likelihood, and hence, we can expect them to be more accurate. To see the effect of estimating independent quantities separately, consider the following example.

**Example 3.** Imagine rolling two fair dice, one of which is blue and one of which is red. By definition of the dice being fair, the outcome of each distinct roll should be  $\frac{1}{36}$ . Suppose that we did not know the dice were fair, and we were trying to estimate the probability of each outcome by experimentation.

Rolling both dice 100 times, we obtain the following frequencies for each roll:

		red					
		1	2	3	4	5	6
blue	1	0.02	0.02	0.02	0.04	0.01	0.02
	2	0.02	0.02	0.04	0.04	0.03	0.02
	3	0.04	0.00	0.01	0.06	0.03	0.06
	4	0.00	0.04	0.01	0.01	0.06	0.04
	5	0.06	0.07	0.03	0.01	0.05	0.01
	6	0.02	0.02	0.01	0.00	0.02	0.04

Note that the ideal value for each estimate is  $\frac{1}{36} \approx 0.0278$ . However, the deviations from this value are quite large, and include some bins with 0 probability.

Even if we don't know whether the dice are fair, it seems reasonable in most scenarios to assume they are independent of each other. Estimating the probabilities of each die separately, from the same data, and computing estimated joint probabilities by multiplying the marginals yields:

		red					
		1	2	3	4	5	6
blue	1	0.0208	0.0221	0.0156	0.0208	0.0260	0.0247
	2	0.0272	0.0289	0.0204	0.0272	0.0340	0.0323
	3	0.0320	0.0340	0.0240	0.0320	0.0400	0.0380
	4	0.0256	0.0272	0.0192	0.0256	0.0320	0.0304
	5	0.0368	0.0391	0.0276	0.0368	0.0460	0.0437
	6	0.0176	0.0187	0.0132	0.0176	0.0220	0.0209

A quick examination of the table reveals that the estimated probabilities are far more accurate. This gain is to be expected if the random variables in question are truly independent. However, when they are somewhat dependent, then there is a trade-off between the inaccuracy of assuming they are independent when they are not, and the gain in estimation accuracy from estimating each variable separately. A classifier that assumes variables are independent is called a *Naive Bayes* classifier, and the impressive performance of these classifiers shows that the loss from assuming independence may be outweighed by the gain that comes from estimating probabilities separately in many cases.