

*History tells the future.*

## 7.1 Introduction

All Bayesian classification methods are based on the Bayes' theorem which is given below:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\bar{A})P(\bar{A})} \quad (7.1)$$

where

- $A$  and  $B$  are random events and  $\bar{A}$  is the complement of  $A$ .
- $A$  is a hypothesis to be tested or predicted.
- $B$  is the new data or observation, it is the new evidence to predict  $A$ .
- $P(A)$  is called the *prior* probability and  $P(B|A)$  is called the *likelihood*, they represent our experience or prior knowledge.
- $P(B)$  is the observation probability or the chance to observe event  $B$ .
- $P(A|B)$  is called the *posterior* probability.

The idea of Bayes' theorem is to convert the computation of probability of  $P(A|B)$  to  $P(B|A)$  which is easier to compute. This is extremely helpful when two random events  $A$  and  $B$  are dependent on each other and the prediction of event  $A$  is difficult due to lack of evidence or information; in this situation, the information about  $B$  can be employed to help predicting  $A$  more accurately. The information about  $B$  can usually be obtained from historical data or experience.

This idea of predicting one event using other related events has been practiced by human beings all the time, for example, we use symptoms to predict a disease, use cloud to predict rain, use a man's culture/background to predict his behavior, use rainfall to predict harvest, etc. In the following, we use two simple examples to get some firsthand understanding of how Bayesian theorem works in real-world applications.

For the first example, we are planning for a sports event at a weekend in a local club and we want to know if the weather will be fine at the weekend. We know the weather and humidity are highly related, we can use humidity to help us predict if the weather is fine so that the sports can go ahead. Formally, let

$$R = \text{Rain} \quad \bar{R} = \text{No rain} \quad H = \text{High humidity (humidity} > 80\%)$$

Suppose we know from history (e.g., Bureau of meteorology) the following prior information:

$$P(R) = 35\% \quad P(\bar{R}) = 65\% \quad P(H|\bar{R}) = 30\%$$

Suppose further we know there will be high humidity at the weekend based on the most recent day weather, then we can predict the chance of no rain at the weekend by the Bayes' theorem as given below:

$$\begin{aligned} P(\bar{R}|H) &= \frac{P(H|\bar{R})P(\bar{R})}{P(H)} = \frac{P(H|\bar{R})P(\bar{R})}{P(H|\bar{R})P(\bar{R}) + P(H|R)P(R)} \\ &= \frac{0.3 \times 0.65}{0.3 \times 0.65 + 1 \times 0.65} = \frac{0.195}{0.845} \approx 23\% \end{aligned}$$

This information is useful for making a reasonable decision on if the sports event should go ahead. Notice that without the prior information of  $P(\bar{R})$  and  $P(H|\bar{R})$ , the prediction and decision would have been made arbitrarily. In practice, factors such as humidity, temperature, and atmospheric pressure are combined to obtain an even more accurate prediction.

Another example is the application of the Bayes' theorem on image classification. Suppose we have detected some black and white strips (through feature extraction) in an image, based on our experience, we believe it is likely a zebra image. But how likely the image is a zebra, 70% of the chance, 80 or 99%? For many other cases like financial, economic, or military situations, this likelihood is crucial to make a right decision. It is clear we need more evidence to determine the accurate likelihood. The answer is in statistics.

If we were able to sample all the images in the world just like a population census in a country, we would be able to calculate the statistics and tell how many non-zebra images would have the black and white strips. This statistic would help

us to determine how likely the image with black and white strips is a zebra image. Unfortunately, we are not able to do a census on all the images in the world, all we can do is to sample part of the image population and create an image database, then use the statistics calculated from the image database to approximate those in the image population.

Formally, let

$Z$  = Zebra image    $\bar{Z}$  = non-Zebra image    $BWS$  = Black and white strips

Now, suppose we know the following probabilities from a training set or an image database (our experience or prior information):

$$P(BWS|Z) = 1.0 \quad P(Z) = 0.05 \quad P(BWS|\bar{Z}) = 0.01$$

Then, given the black and white strips in an image, we can predict if there is a zebra in the image by the Bayes' theorem as given below:

$$\begin{aligned} P(Z|BWS) &= \frac{P(BWS|Z)P(Z)}{P(BWS)} = \frac{P(BWS|Z)P(Z)}{P(BWS|Z)P(Z) + P(BWS|\bar{Z})P(\bar{Z})} \\ &= \frac{1.0 \times 0.05}{1.0 \times 0.05 + 0.01 \times 0.95} = \frac{0.05}{0.0595} \approx 84\% \end{aligned}$$

Therefore, we can say there is a high chance that the image is a zebra image and we have high confidence to classify the image into the zebra image category.

Bayes' theorem can be extended to multiple events  $A_1, A_2, \dots, A_n$  as follows:

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} = \frac{P(B|A_i)P(A_i)}{P(B|A_1)P(A_1) + \dots + P(B|A_n)P(A_n)} \quad (7.2)$$

In this case,  $B$  is related to or dependent on multiple other events  $A_i$ , and based on what we know about the relationship or dependency between  $B$  and each  $A_i$ :  $P(B|A_i)$ , we can predict if a new observation of  $B$  is from any of the events  $A_i$ .

For example, fever ( $B$ ) can be caused by many diseases or medical conditions ( $A_i$ ), such as infection, flu, pneumonia, chickenpox, measles, HIV, meningitis, cancers, malaria, Dengue, etc. However, each disease or medical condition has different chances of causing fever:  $P(B|A_i)$ . Now, given a patient with fever, (7.2) can be used to determine if the patient has flu:  $P(A_i|B)$ . In clinical practice, however, symptoms are typically combined to nail a disease, e.g., by combining fever with running nose and headache, flu can be diagnosed with very high accuracy.

## 7.2 Naïve Bayesian Image Classification

### 7.2.1 NB Formulation

The *naïve Bayesian* (NB) methods are based on a simple application of the above Bayes' theorem on numerical and high-dimensional image data.

- Given a set of  $N$  images:  $I = \{I_1, I_2, \dots, I_N\}$ .
- And a set of  $n$  semantic classes  $C = \{C_1, C_2, \dots, C_n\}$  (events).
- $I \in C$ .
- Each image  $I$  is represented by a feature vector  $\mathbf{x} = (x_1, x_2, \dots, x_m)$  (observation).

According to Bayes' theorem, the classification or annotation of image  $I$  to class  $C_i$  is given by

$$P(C_i|I) = P(C_i|\mathbf{x}) = \frac{P(\mathbf{x}|C_i)P(C_i)}{P(\mathbf{x})} = \frac{P(\mathbf{x}|C_i)P(C_i)}{P(\mathbf{x}|C_1)P(C_1) + \dots + P(\mathbf{x}|C_n)P(C_n)} \quad (7.3)$$

or

$$P(C_i|I) = P(C_i|\mathbf{x}) = \frac{P(\mathbf{x}|C_i)P(C_i)}{\sum_{k=1}^n P(\mathbf{x}|C_k)P(C_k)} \quad (7.4)$$

Because the denominator  $P(\mathbf{x}) = \sum_{k=1}^n P(\mathbf{x}|C_k) \times P(C_k)$  is independent of class  $C_i$  ( $i = 1, 2, \dots, n$ ) and is a constant, (7.4) can be written as follows:

$$P(C_i|I) = P(C_i|\mathbf{x}) = \frac{1}{Z} P(\mathbf{x}|C_i)P(C_i) \quad (7.5)$$

where  $Z = \sum_{k=1}^n P(\mathbf{x}|C_k)P(C_k)$  is a scaling factor. The class of image  $I$  can be decided using the *maximizing a posterior* (MAP) criterion

$$P(C_j|I) = \hat{C} = \arg \max_i P(C_i|\mathbf{x}) = \arg \max_i \{P(\mathbf{x}|C_i)P(C_i)\} \quad (7.6)$$

The prior probabilities  $P(C_i)$  is usually uniform for all classes; otherwise, they can be found by the frequency or proportion of samples belonging to class  $C_i$  among all classes. Therefore, the classification of image  $I$  comes down to modeling the likelihood probability of  $P(\mathbf{x}|C_i)$ .

Since image features are typically numerical and continuous, they need to be discretized before the likelihood modeling. In practice, the following procedure is used to compute the  $P(\mathbf{x}|C_i)$  in (7.6).

*Training:*

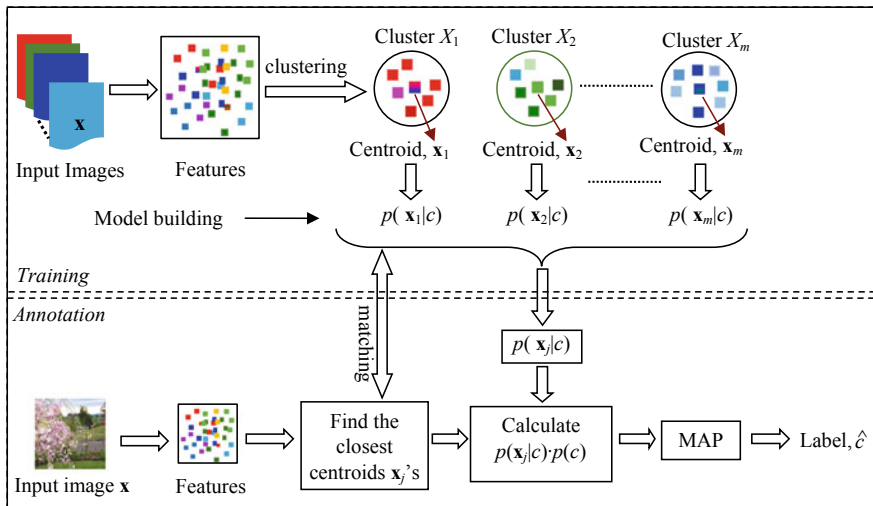
- A training database of images from all the  $n$  classes  $\{C_1, C_2, \dots, C_n\}$  are created.
- Image features from the training database are clustered into  $m$  clusters  $X_j$  using a certain vector quantization algorithm.
- Next, a cluster centroid  $\mathbf{x}_j$  is computed for each of the clusters  $X_j$ ;  $\mathbf{x}_j, j = 1, 2, \dots, m$ .
- Then, the likelihood  $P(\mathbf{x}_j|C_i)$  is calculated by finding the frequency of samples in  $X_j$  belonging to class  $C_i$ .

$$P(\mathbf{x}_j|C_j) = \frac{\text{No. of samples in } X_j \text{ which are from class } C_i}{\text{Total no. of samples in cluster } X_j} \quad (7.7)$$

*Classification/Annotation:*

- Given a new image  $I$  with feature  $\mathbf{x}$ .
- Match feature  $\mathbf{x}$  to the closest cluster centroids  $\mathbf{x}_j$ 's.
- Apply the MAP of (7.6) by replacing the likelihood  $P(\mathbf{x}|C_i)$  with (7.7) to obtain the posterior probability  $P(C_j|I)$ .

The classification and annotation of an image with Naïve Bayesian method is illustrated in Fig. 7.1 [1, 2]. There are two major modules in an NB classifier: *training* and *annotation*, and each of the major modules consists of three



**Fig. 7.1** Image classification with Naïve Bayesian method

sub-modules. The training module consists of *feature extraction*, *clustering*, and *model building*, while the annotation consists of *feature extraction*, *matching*, and *decision-making* using MAP.

### 7.2.2 NB with Independent Features

Assume  $x_1, x_2, \dots, x_m$  are independent of each other, then the likelihood is given as follows:

$$P(\mathbf{x}|C_i) = \prod_{j=1}^m P(x_j|C_i) \quad (7.8)$$

There are many situations where the features of a data are independent on each other, e.g., nominal features extracted by a web crawler, such as tree, grass, sand, water, etc. Suppose we are using a set of nominal features  $\mathbf{x} = (\text{sand}, \text{water}, \text{sky}, \text{people})$  to classify a collection of images into “beach” and “non-beach” categories, then (7.8) can be employed to modeling the likelihood in the Bayes’ theorem. Often different types of numerical image features are combined into a more powerful feature vector, e.g.,  $\mathbf{x} = (\text{color}, \text{shape}, \text{texture})$ , again, the likelihood probability in the Bayes’ theorem can be computed using (7.8).

### 7.2.3 NB with Bag of Features

If an image  $I$  is segmented into  $k$  regions, and each region is represented as a feature vector  $\mathbf{x}_j, j = 1, 2, \dots, k$ ,  $I$  can be represented as a *bag of features*:  $I = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$ . Typically, regions in an image are independent of each other; therefore, the conditional probability of  $P(I|C_i)$  is given by

$$P(I|C_i) = P(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k|C_i) = \prod_{j=1}^k P(\mathbf{x}_j|C_i) \quad (7.9)$$

---

## 7.3 Image Annotation with Word Co-occurrence

In the above naïve Bayesian classification, images are not individually labeled, instead, they are simply classified into categories. The categories can be regarded as *implicit image annotation* or collective image annotation. However, individual images can be pre-labeled and the annotation of images can be done explicitly. Vast amount of labeled images are available on the web, they can be employed to annotate new images. One of the earliest works on *explicit image annotation* or individual image annotation is the *Word Co-occurrence model* (WCC) introduced

by Mori et al. [3]. The idea is to establish the relationship between image features and the labels, and use the relationship as a likelihood model to label new images. Specifically, features from pre-labeled image are clustered into clusters and a *word histogram* is computed from each cluster as the likelihood model. The idea of their method can be summarized as follows:

1. Collecting training images with pre-labeled keywords,
2. Divide each image into parts and extract features from each part,
3. Each divided part inherits all words from its original image,
4. Make clusters from all divided images using vector quantization,
5. Accumulate the frequencies of words of all partial images in each cluster, and calculate the likelihood for every word,
6. For an unknown image, divide it into parts, extract their features, and match the image parts with the above clusters. Combine the likelihoods of the image parts and determine which words are most plausible.

The *algorithm* of the word co-occurrence model is given as follows:

- **Collect and label training images.** Given a training dataset of  $n$  images  $\mathbf{I} = (I_1, I_2, \dots, I_n)$  and each image  $I_i$  is pre-labeled with a set of semantic words  $\mathbf{w}_i$ :

$$(\mathbf{I}, \mathbf{w}) = \{(I_1, \mathbf{w}_1), (I_2, \mathbf{w}_2), \dots, (I_n, \mathbf{w}_n)\}$$

- **Obtain the vocabulary of the training images.** The semantic vocabulary of the dataset consists of  $m$  words:

$$\mathbf{w} = (w_1, w_2, \dots, w_m)$$

- **Divide training images into blocks.** Each training image is divided into small blocks and each block inherits all the annotations from its parent image.
- **Vector Quantization (VQ)**

- Blocks from all the training images are clustered into  $v$  clusters represented by the centroids  $c_1, c_2, \dots, c_v$ .
- Each cluster  $c_i$  is represented as a feature vector  $\mathbf{x}_i$  (each cluster is called a *visual word* or VW, which is corresponding to a region in the training images):

$$\mathbf{c} = (c_1, c_2, \dots, c_v) = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_v)$$

- **Obtain a word histogram in each cluster.** Because each block has inherited a set of words from its parent image, by counting the occurrence of words, a histogram of words from the vocabulary can be created:

$$P(w_j|c_i) = P(w_j|\mathbf{x}_i) = (W_1, W_2, \dots, W_m) \quad (7.10)$$

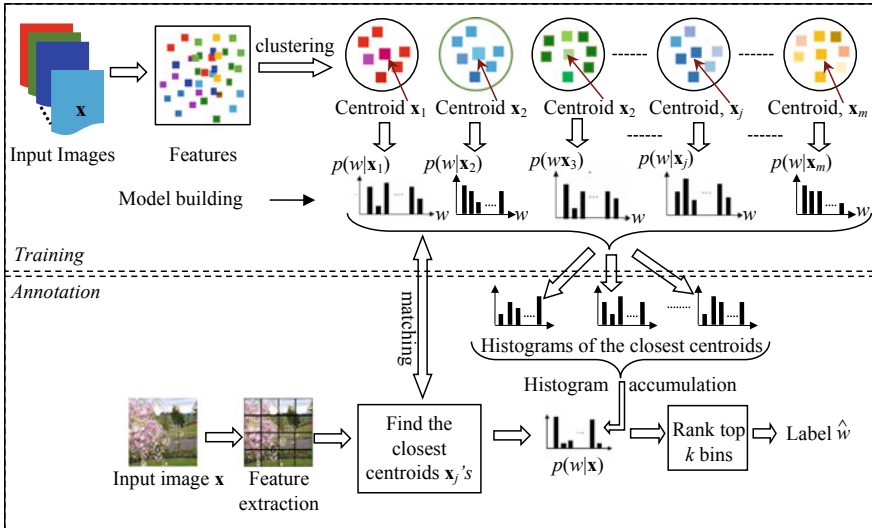
where  $W_j$  represents the frequency of word  $w_j$  in cluster  $c_i$ ,  $P(w_j | c_i)$  represents the likelihood of word  $w_j$ .

• **Annotate an unknown image**

- Given an unknown image  $I_u$ , it is also divided into small blocks and the blocks of the unknown image are also clustered into clusters.
- Each unknown cluster is matched with the VWs and the nearest  $l$  VWs are found for the unknown image.
- The matching is done by calculating the distance between each feature of the unknown image  $\mathbf{x}_u$  and each VW  $\mathbf{x}_i$ :  $\|\mathbf{x}_u - \mathbf{x}_i\|$ .
- The annotation of image  $I_u$  to a semantic word  $w_j$  ( $j = 1, 2, \dots, k$ ) is given by first summing up of the histograms of matched clusters  $c_i$  ( $i = 1, \dots, l$ ) and then selecting the top  $k$  bins as the annotations:

$$P(w_1, \dots, w_k | I_u) = \text{top } k \text{ bins} \left( \sum_{i=1}^l P(w_j | c_i) \right) \quad (7.11)$$

The co-occurrence annotation method can be illustrated in Fig. 7.2 [1, 2]. There are two key differences between Figs. 7.2 and 7.1. The first is in the training module, while the NB builds a model of  $p(\mathbf{x}_i|c)$ , the WCC builds a model of  $p(w|c)$ . The second difference in the annotation module, while the NB makes a decision based on MAP, the WCC makes a decision based on top histogram bins which means an image can be classified into several classes.



**Fig. 7.2** Image annotation with co-occurrence of words

Although the co-occurrence method uses image blocks for the VQ, the blocks can be replaced with pre-segmented image regions. This is because regardless of blocks or regions, they are all represented with a feature vector  $\mathbf{x}$ , and the VQ is done based on feature vector  $\mathbf{x}$ .

## 7.4 Image Annotation with Joint Probability

The word co-occurrence model is a significant development to traditional image classification, it can be generalized into a joint probability model which is described in the following:

- Given a training dataset of  $n$  pre-annotated images:

$$(\mathbf{I}, \mathbf{w}) = (I_1, \mathbf{w}_1), (I_2, \mathbf{w}_2), \dots, (I_n, \mathbf{w}_n)$$

- The semantic vocabulary of the dataset consists of  $m$  words:

$$\mathbf{w} = (w_1, w_2, \dots, w_m)$$

- The annotation or association of an unknown image  $I$  to a word  $w$  in the vocabulary can be found by the joint probability of  $P(w, I)$  or  $P(w, \mathbf{x})$ , where  $\mathbf{x}$  is the feature of  $I$ .
- In order to compute  $P(w, I)$ , a latent variable  $c$  is introduced

$$P(w|I) = P(w, I) = P(w|c) \times P(c|I) \quad (7.12)$$

The computation of conditional probabilities  $P(w|c)$  and  $P(c|I)$  are given in the following procedure:

- The training images are clustered into  $v$  clusters or VWs (*the latent variables*):

$$\mathbf{c} = (c_1, c_2, \dots, c_v) = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_v)$$

- An image to be annotated is represented as a histogram or distribution of VWs:

$$P(c_i|I) = (X_1, X_2, \dots, X_v) \quad (7.13)$$

where  $X_i$  is the frequency of VW  $\mathbf{x}_i$  in image  $I$

- Each cluster is represented as a histogram or distribution of vocabulary words:

$$P(w_j|c_i) = (W_1, W_2, \dots, W_m) \quad (7.14)$$

where  $W_j$  represents the frequency of word  $w_j$  in cluster  $c_i$ ,

- Finally, the annotation of image  $I$  to a word  $w_j$  is given by

$$P(w_j|I) = P(w_j|c_i) \times P(c_i|I) \quad (7.15)$$

As discussed in Sect. 7.2, images can be segmented into regions and represented as a *bag of features* for annotation.

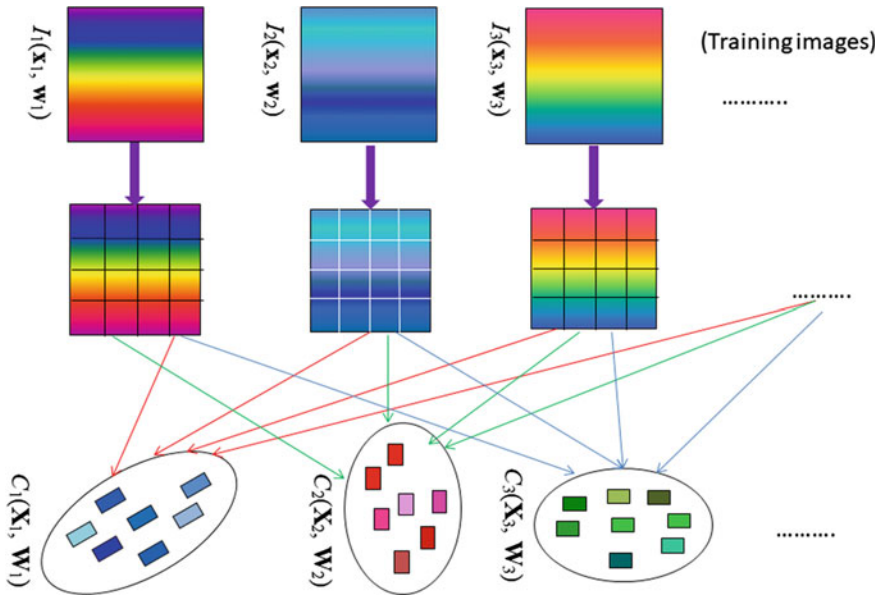
- If an image  $I$  is represented as a *bag of features* (pre-clustered):  $I = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k\}$
- The conditional probability of  $P(c_i|I)$  in (7.15) can be computed using MAP:

$$P(c_i|I) = \arg \max_{c_j} P(c_j|I) = \arg \max_{c_j} \{P(I|c_j) \times P(c_j)\} \quad (7.16)$$

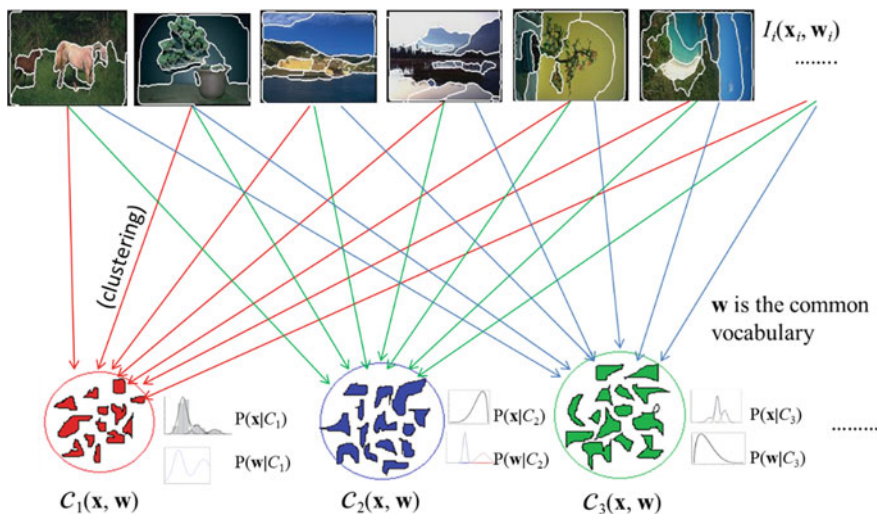
where

$$P(I|c_j) = P(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k|c_j) = \prod_{l=1}^k P(\mathbf{x}_l|c_j) \quad (7.17)$$

The key idea of image annotation based on the joint probability model is the association of semantic words with visual words. This is achieved through the VQ process. Once image features are clustered, each cluster (VW) and the semantic words are bound together or associated, because each image feature inherits the semantic word(s) from its parent image. This idea can be illustrated in Fig. 7.3.



**Fig. 7.3** Association of semantic words with block features



**Fig. 7.4** Association of semantic words with region features

Once image features are clustered and visual words are generated, two types of distribution can be created from each cluster:  $P(\mathbf{x}|c_i)$  and  $P(w|c_i)$ .  $P(w|c_i)$  is the word distribution in cluster  $c_i$ , it connects the VWs to the semantic vocabulary.  $P(\mathbf{x}|c_i)$  is the feature distribution in cluster  $c_i$ , it connects each VW with each of the images in the database (Fig. 7.4). By combining these two important information, new image can be annotated as shown in the above sections.

## 7.5 Cross-Media Relevance Model

Although VQ is typical in building the likelihood models, models can also be built “on the fly” by a set of training images which are relevant to the new observation. The *cross-media relevance model* (CMRM) [4] provides an alternative method to the VQ and is another joint probability model.

Given an image which is represented by a set of blobs:  $I = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ , the association of  $I$  with concept  $c$  is given by the joint probability of  $p(c, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m)$  as given below:

$$\begin{aligned}
 p(c, \mathbf{x}_1, \dots, \mathbf{x}_m) &= \sum_{J \in T} p(J) \times p(c, \mathbf{x}_1, \dots, \mathbf{x}_m | J) \\
 &= p(J) \times \sum_{J \in T} p(c | J) \times \prod_{i=1}^m p(\mathbf{x}_i | J)
 \end{aligned} \tag{7.18}$$

where

$$p(c|J) = (1 - \alpha_J) \times \frac{\#(c, J)}{|J|} + \alpha_J \times \frac{\#(c, T)}{|T|} \quad (7.19)$$

$$p(\mathbf{x}_i|J) = (1 - \beta_J) \times \frac{\#(\mathbf{x}_i, J)}{|J|} + \beta_J \times \frac{\#(\mathbf{x}_i, T)}{|T|} \quad (7.20)$$

where

- $J$  is an image in the training set  $T$ ,
- $\alpha_J$  and  $\beta_J$  are the interpolation parameters,
- $\#(c, J)$  is the number of times concept  $c$  appears in  $J$ ,
- $\#(\mathbf{x}_i, J)$  is the number of times blob  $\mathbf{x}_i$  appears in  $J$ ,
- $\#(c, T)$  is the number of times concept  $c$  appears in  $T$ , and
- $\#(\mathbf{x}_i, T)$  is the number of times blob  $\mathbf{x}_i$  appears in  $T$ .

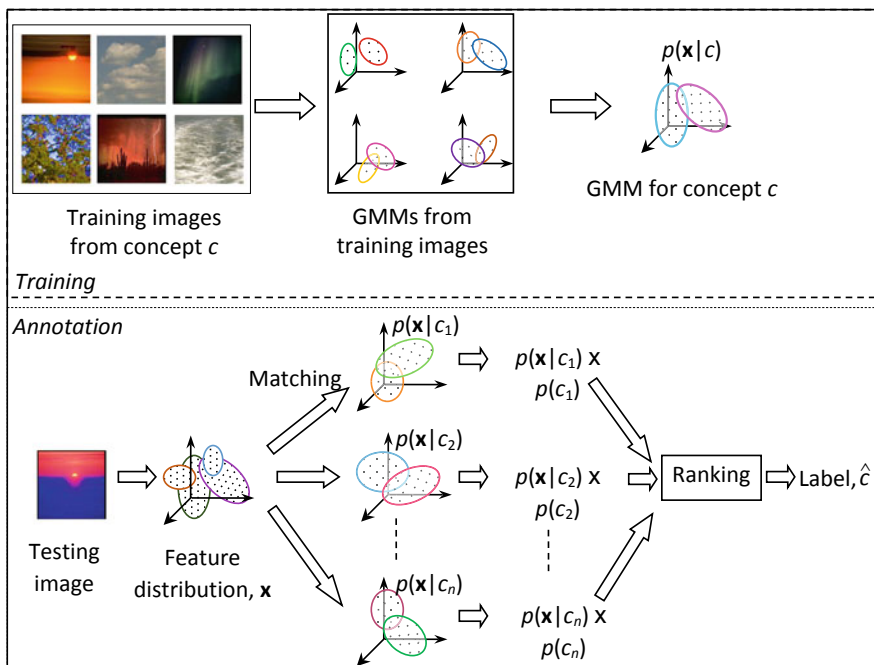
It can be seen from (7.18), given a new observation  $I = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ , CMRM attempts to find all the relevant images in the training set that have both concept  $c(p(c|J) \neq 0)$  and feature  $\mathbf{x}_i(p(\mathbf{x}_i|J) \neq 0)$ . A joint probability model is built by aggregating all the models from the relevant images. This is equivalent to build a class model for concept  $c$  “on the fly”. From (7.19) and (7.20), it can be seen that the performance of this model depends on the choice of the weights  $\alpha_J$  and  $\beta_J$ . In practice, this can be a difficult decision to make.

---

## 7.6 Image Annotation with Parametric Model

One of the classic ways of model building is the parametric method using the *expectation-maximization* (EM) algorithm. The idea of image annotation with parametric model is similar to the CMRM method, that is, to build a model for each of the individual images and aggregate the similar individual models into a class model. However, instead of combining training and annotation into a single process as in the CMRM method, parametric model separates training and annotation into two different processes.

During the training, images in the training set are pre-labeled and pre-classified into different classes. A class model is then built by aggregating individual image models in each class. During the annotation, the model of the new image is built and matched with the class models, and the closest classes are selected as the annotations. The procedure of parametric method is shown in Fig. 7.5. The algorithm of this method is as follows:



**Fig. 7.5** Image annotation with parametric model

- Given a set of  $N$  training images:  $I_1, I_2, \dots, I_N$  and a set of  $n$  classes  $C_1, C_2, \dots, C_n$
- Features (e.g., block features) from each training image  $I$  are clustered within the image.
- A Gaussian Mixture Model (GMM) is learned from the clustering using the EM algorithm:

$$P(\mathbf{x}|I) = \sum_{i=1}^l \pi_l^i \mathcal{G}(\mathbf{x}, \mu_l^i, \Sigma_l^i) \quad (7.21)$$

where

- $l$  is the number of components in the mixture model of image  $I$ ,
- $\pi_l^i$  is the weight for the  $i$ th component of the mixture model,
- $\mu_l^i$  is the mean of the  $i$ th component of the mixture model, and
- $\Sigma_l^i$  is the standard deviation of the  $i$ th component of the mixture model.
- A Gaussian mixture model for each class  $C_i$  is learnt by aggregating (e.g., weighted averaging) all the image models within the class:

$$P(\mathbf{x}|C_i) = \sum_{k=1}^K \pi_{C_i}^k \mathcal{G}(\mathbf{x}, \mu_{C_i}^k, \Sigma_{C_i}^k) \quad (7.22)$$

where

- $K$  is the number of components in the mixture model of class  $C_i$ ,
  - $\pi_{C_i}^k$  is the weight for the  $k$ th component of the mixture model,
  - $\mu_{C_i}^k$  is the mean of the  $k$ th component of the mixture model, and
  - $\Sigma_{C_i}^k$  is the standard deviation of the  $k$ th component of the mixture model.
- Given a new observation image  $I_u = \mathbf{x}_u$ , the annotation of image  $I_u$  is given by the MAP:

$$\begin{aligned} P(C_j|\mathbf{x}_u) &= \hat{c} = \arg \max_{C_i} P(C_i|\mathbf{x}_u) \\ &= \arg \max_{C_i} \{P(\mathbf{x}_u|C_i) \times P(C_i)\} \end{aligned} \quad (7.23)$$

The algorithm of the parametric annotation method is illustrated in Fig. 7.5 [2, 5].

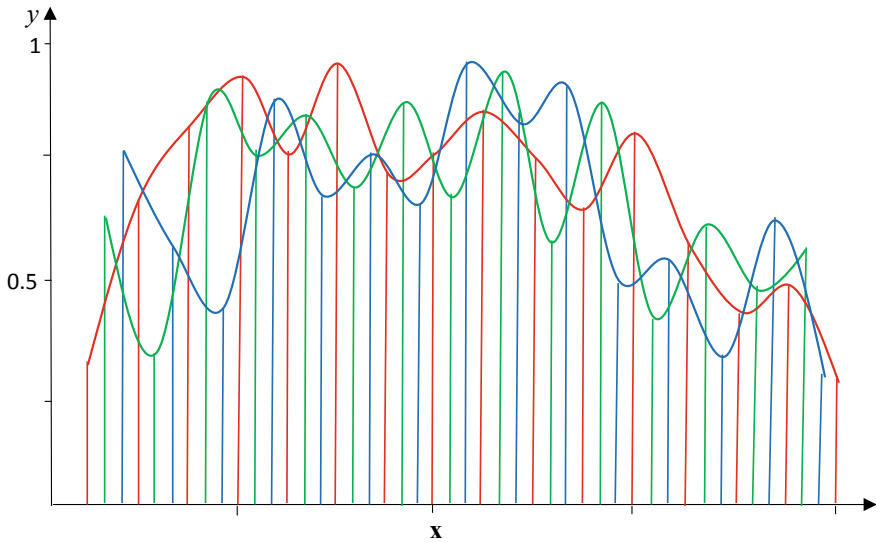
---

## 7.7 Image Classification with Gaussian Process

In Gaussian mixture, each multidimensional feature vector  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  is regarded as a data point in a  $R^n$  space, and the mixture model is built based on the statistics of the data points in a cluster.

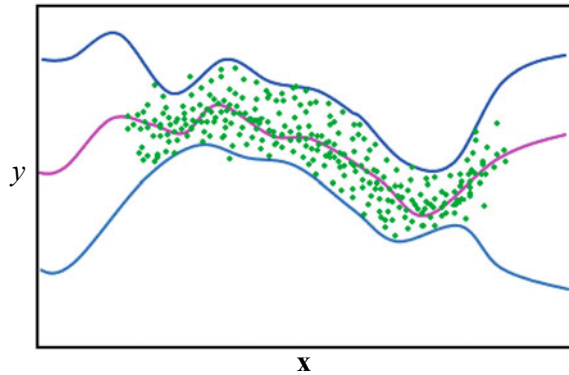
But a multidimensional data  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  can also be regarded as a discretized function  $f: X \rightarrow R$  and  $y = f(\mathbf{x}) = \{x_i = f(d_i) \mid i = 1, 2, \dots, n\}$ . A typical example of such a data is a histogram feature vector. Figure 7.6 shows three normalized histograms (vertical bars) from the same class in red, green, and blue, respectively. The corresponding functions approximating the three histograms are shown as colored curves at the top of the histograms.

If we plot all the histogram features  $f(\mathbf{x}_i)$  from a class in a single coordinate system, we would see all the data fall within a band and form a cluster. Like in the linear regression which attempts to fit a line to a cluster of data points, we can also fit a curve to this cluster of data points and use this curve as the model to predict new instances. This approach is the idea behind the Gaussian Process or GP, which is demonstrated in Fig. 7.7 [6].



**Fig. 7.6** Feature vectors shown as functions. Three histograms shown as vertical color bars and their respective functions shown as colored curves on the top

**Fig. 7.7** A cluster of multidimensional data (green) and the approximation function of the data shown in pink



Now, given a set of data  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  from a certain class  $C$ , and each feature vector  $\mathbf{x}_i$  is a  $D$ -dimensional data point in space  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ . A matrix  $\mathbf{X} = D \times N = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_D)^T$  can be created as shown in the following:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{N1} \\ x_{12} & x_{22} & \dots & x_{N2} \\ \dots & \dots & \dots & \dots \\ x_{1D} & x_{2D} & \dots & x_{ND} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \dots \\ \mathbf{d}_D \end{bmatrix} \quad (7.24)$$

where  $\mathbf{d}_j$  is a  $j$ th dimensional vector. Since the elements of each  $\mathbf{d}_j$  are samples from (or follow) a normal distribution  $\mathcal{N}(\mu_i, \sigma_j)$ ,  $\mathbf{X}$  is a Gaussian process and  $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_X, \mathbf{K}_{XX})$ , where  $\boldsymbol{\mu}_X$  and  $\mathbf{K}_{XX}$  are the *mean* and *variance* which are determined by (7.25) and (7.26), respectively.

$$\boldsymbol{\mu}_X = \begin{bmatrix} \mu(\mathbf{d}_1) \\ \mu(\mathbf{d}_2) \\ \vdots \\ \mu(\mathbf{d}_D) \end{bmatrix} \quad (7.25)$$

$$\mathbf{K}_{XX} = \begin{bmatrix} k(\mathbf{d}_1, \mathbf{d}_1), k(\mathbf{d}_1, \mathbf{d}_2) \dots, k(\mathbf{d}_1, \mathbf{d}_D) \\ k(\mathbf{d}_2, \mathbf{d}_1), k(\mathbf{d}_2, \mathbf{d}_2) \dots, k(\mathbf{d}_2, \mathbf{d}_D) \\ \vdots \\ k(\mathbf{d}_D, \mathbf{d}_1), k(\mathbf{d}_D, \mathbf{d}_2) \dots, k(\mathbf{d}_D, \mathbf{d}_D) \end{bmatrix} \quad (7.26)$$

where  $k(\mathbf{d}_i, \mathbf{d}_j)$  is a *kernel* function which is typically the *covariance* function.

To predict a new data or a new set of data  $\mathbf{X}_*$ ,  $\mathbf{X}$  and  $\mathbf{X}_*$  are concatenated and the concatenated data is a new GP which follows the following normal distribution:

$$f\left(\begin{matrix} \mathbf{X}_* \\ \mathbf{X} \end{matrix}\right) \sim \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\mu}_{X_*} \\ \boldsymbol{\mu}_X \end{pmatrix}, \begin{pmatrix} \sum \mathbf{x}_* \mathbf{x}_*^T, \sum \mathbf{x}_* \mathbf{x}^T \\ \sum \mathbf{x} \mathbf{x}_*^T, \sum \mathbf{x} \mathbf{x}^T \end{pmatrix}\right) \quad (7.27)$$

Then, the probability of the new data  $\mathbf{X}_*$  given the observed data  $\mathbf{X}$  is given by (7.28)

$$p(\mathbf{X}_* | \mathbf{X}) = \mathcal{N}(\boldsymbol{\mu}_{X_*} + \mathbf{K}_{X_* X} \mathbf{K}_{XX}^{-1} (\mathbf{X} - \boldsymbol{\mu}_X), \mathbf{K}_{X_* X_*} - \mathbf{K}_{X_* X} \mathbf{K}_{XX}^{-1} \mathbf{K}_{XX_*}) \quad (7.28)$$

The proof of (7.28) is given in Appendix [7–9].

---

## 7.8 Summary

This chapter introduces the first image classification method: Bayesian classification. Several important and interesting applications of Bayesian classifier are described and demonstrated in details, including NB, word co-occurrence model, CMRM, parametric model, and Gaussian process method. The key features of Bayesian classifiers can be summarized as follows:

1. **Generative.** A Bayesian classifier is a typical generative model, it assumes the distribution or model of likelihood probability is known. This likelihood probability model is typically obtained through learning from known samples.
2. **Intuitive.** Compared with many other black-box-based classifiers such as SVM and ANN, Bayesian classifiers are intuitive, results are easily interpreted by a human being. The basic idea of the Bayesian method is to use our prior