Chapter 6

Learning and inference in vision

At an abstract level, the goal of computer vision problems is to use the observed image data to infer something about the world. For example, we might observe adjacent frames of a video sequence and infer the camera motion, or we might observe a facial image and infer the identity.

The aim of this chapter is to describe a mathematical framework for solving this type of problem and to organize the resulting models into useful subgroups, which will be explored in subsequent chapters.

6.1 Computer vision problems

In vision problems, we take visual data $\mathbf{x}$ and use them to infer the state of the world $\mathbf{w}$. The world state $\mathbf{w}$ may be continuous (e.g., the 3D pose of a body model) or discrete (e.g., the presence or absence of a particular object). When the state is continuous, we call the inference process *regression*. When the state is discrete, we call it *classification*.

Unfortunately, the measurements $\mathbf{x}$ may be compatible with more than one world state $\mathbf{w}$. The measurement process is noisy and there is inherent ambiguity in visual data: a lump of coal viewed under bright light may produce the same luminance measurements as white paper in dim light. Similarly, a small object seen close-up may produce the same image as a larger object that is further away.

In the face of such ambiguity, the best that we can do is to return the *posterior probability distribution* $Pr(\mathbf{w}|\mathbf{x})$ over possible states $\mathbf{w}$. This describes everything we know about the state after observing the visual data. So, a more precise description of an abstract vision problem is that we wish to take observations $\mathbf{x}$ and return the whole posterior probability distribution $Pr(\mathbf{w}|\mathbf{x})$ over world states.

In practice, computing the posterior is not always tractable; we often have to settle for returning the world state $\hat{\mathbf{w}}$ at the peak of the posterior (the maximum a posteriori solution). Alternatively, we might draw samples from the posterior and use the collection of samples as an approximation to the full distribution.

6.1.1 Components of the solution

To solve a vision problem of this kind, we need three components.

- We need a *model* that mathematically relates the visual data $\mathbf{x}$ and the world state $\mathbf{w}$. The model specifies a family of possible relationships between $\mathbf{x}$ and $\mathbf{w}$ and the particular relationship is determined by the model parameters $\boldsymbol{\theta}$.
- We need a *learning algorithm* that allows us to fit the parameters $\boldsymbol{\theta}$ using paired training examples $\{\mathbf{x}_i, \mathbf{w}_i\}$ where we know both the measurements and the underlying state.
- We need an *inference algorithm* that takes a new observation $\mathbf{x}$ and uses the model to return the posterior $Pr(\mathbf{w}|\mathbf{x}, \boldsymbol{\theta})$ over the world state $\mathbf{w}$. Alternately, it might return the MAP solution or draw samples from the posterior.

The rest of this book is structured around these components: each chapter focusses on one model or one family of models, and discusses the associated learning and inference algorithms.

6.2 Types of model

The first and most important component of the solution is the model. Models relating the data $\mathbf{x}$ to the world $\mathbf{w}$ fall into one of two categories. We either:

1. model the contingency of the world state on the data $Pr(\mathbf{w}|\mathbf{x})$ or
2. model the contingency of the data on the world state $Pr(\mathbf{x}|\mathbf{w})$.

The first type of model is termed *discriminative*. The second is termed *generative*; here, we construct a probability model over the data and this can be used to generate (confabulate) new observations. Let us consider these two types of model in turn and discuss learning and inference in each.

6.2.1 Model contingency of world on data (discriminative)

To model $Pr(\mathbf{w}|\mathbf{x})$, we choose an appropriate form for the distribution $Pr(\mathbf{w})$ over the world state $\mathbf{w}$ and then make the distribution parameters a function of the data $\mathbf{x}$. So if the world state was continuous, we might model $Pr(\mathbf{w})$ with a normal distribution and make the mean $\boldsymbol{\mu}$ a function of the data $\mathbf{x}$.

The value that this function returns also depends on a set of parameters, $\boldsymbol{\theta}$. Since the distribution over the state depends on both the data and these parameters, we write it as $Pr(\mathbf{w}|\mathbf{x}, \boldsymbol{\theta})$ and refer to it as the *posterior distribution*.

The goal of the learning algorithm is to fit the parameters $\boldsymbol{\theta}$ using paired training data $\{\mathbf{x}_i, \mathbf{w}_i\}_{i=1}^{I}$. This can be done using the maximum likelihood (ML), maximum a posteriori (MAP), or Bayesian approaches (chapter 4).

The goal of inference is to find a distribution over the possible world states $\mathbf{w}$ for a new observation $\mathbf{x}$. In this case, this is easy: we have already directly

constructed an expression for the posterior distribution $Pr(\mathbf{w}|\mathbf{x}, \boldsymbol{\theta})$, and we simply evaluate it with the new data.

6.2.2 Model contingency of data on world (generative)

To model $Pr(\mathbf{x}|\mathbf{w})$, we choose the form for the distribution $Pr(\mathbf{x})$ over the data and make the distribution parameters a function of the world state $\mathbf{w}$. For example, if the data were discrete and multi-valued then we might use a categorical distribution and make the parameter vector $\boldsymbol{\lambda}$ a function of the world state $\mathbf{w}$.

The value that this function returns also depends on a set of parameters $\boldsymbol{\theta}$. Since the distribution $Pr(\mathbf{x})$ now depends on both the world state and these parameters, we write it as $Pr(\mathbf{x}|\mathbf{w}, \boldsymbol{\theta})$ and refer to it as the *likelihood*. The goal of learning is to fit the parameters $\boldsymbol{\theta}$ using paired training examples $\{\mathbf{x}_i, \mathbf{w}_i\}_{i=1}^{I}$.

In inference, we aim to compute the posterior distribution $Pr(\mathbf{w}|\mathbf{x})$. To this end we specify a prior $Pr(\mathbf{w})$ over the world state and then use Bayes' rule,

$$Pr(\mathbf{w}|\mathbf{x}) = \frac{Pr(\mathbf{x}|\mathbf{w})Pr(\mathbf{w})}{\int Pr(\mathbf{x}|\mathbf{w})Pr(\mathbf{w})d\mathbf{w}}. \tag{6.1}$$

Here we have modeled both the likelihood $Pr(\mathbf{x}|\mathbf{w})$ and the prior $Pr(\mathbf{w})$ and multiplied these together in the numerator of Bayes' rule. However, notice that we could have equivalently modeled the joint distribution $Pr(\mathbf{x}, \mathbf{w}) = Pr(\mathbf{x}|\mathbf{w})Pr(\mathbf{w})$ directly. Sometimes generative models are presented in this form (see section 7.9.5).

Summary

We've seen that there are two distinct approaches to modeling the relationship between the world state $\mathbf{w}$ and the data $\mathbf{x}$, corresponding to modeling the posterior $Pr(\mathbf{w}|\mathbf{x})$, or the likelihood $Pr(\mathbf{x}|\mathbf{w})$.

The two model types result in different approaches to inference. For the discriminative model, we describe the posterior $Pr(\mathbf{w}|\mathbf{x})$ directly and there is no need for further work. For the generative model, we compute the posterior using Bayes' rule. This sometimes results in complex inference algorithms.

To make these ideas concrete, we now consider two toy examples. For each case, we will investigate using both generative and discriminative models. At this stage, we won't present the details of the learning and inference algorithms; these are presented in subsequent chapters anyway. The goal here is to introduce the main types of model used in computer vision in their most simple form.

6.3 Example 1: regression

Consider the situation where we make a univariate continuous measurement x and use this to predict a univariate continuous state w. For example, we might predict the distance to a car in a road scene based on the number of pixels in its silhouette.

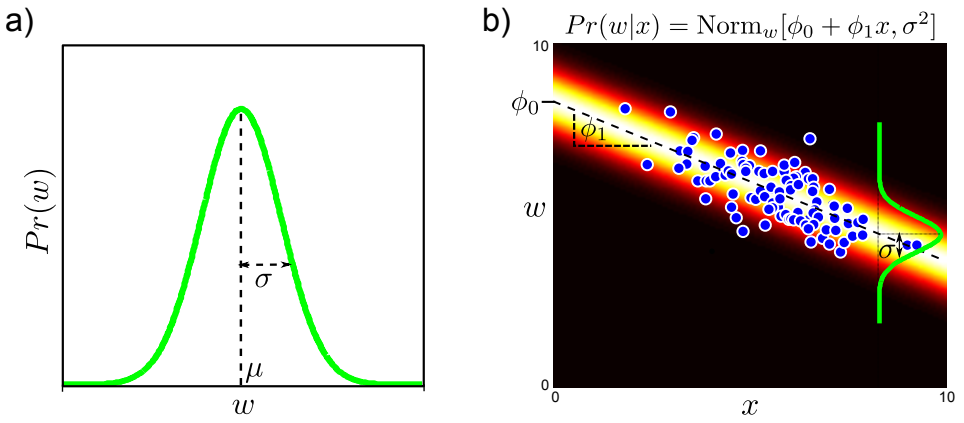

Figure 6.1 Regression by modeling the posterior $Pr(w|x)$ (discriminative). a) We model the world state w as normally distributed. b) We make the normal parameters a function of the observations x: the mean is a linear function $\mu = \phi_0 + \phi_1 x$ of the observations, and the variance σ^2 is fixed. The learning algorithm fits the parameters $\boldsymbol{\theta} = \{\phi_0, \phi_1, \sigma^2\}$ to example training pairs $\{x_i, w_i\}_{i=1}^I$ (blue dots). In inference we take a new observation x and compute the posterior distribution $Pr(w|x)$ over the state.

6.3.1 Model contingency of world on data (discriminative)

Problem 6.5

We define a probability distribution over the world state w and make its parameters contingent on the data x. Since the world state is univariate and continuous, we chose the univariate normal. We fix the variance, σ^2 and make the mean μ a linear function $\phi_0 + \phi_1 x$ of the data. So we have

$$Pr(w|x, \boldsymbol{\theta}) = \text{Norm}_w\left[\phi_0 + \phi_1 x, \sigma^2\right], \tag{6.2}$$

where $\boldsymbol{\theta} = \{\phi_0, \phi_1, \sigma^2\}$ are the unknown parameters of the model (figure 6.1). This model is referred to as *linear regression*.

The learning algorithm estimates the model parameters $\boldsymbol{\theta}$ from paired training examples $\{x_i, w_i\}_{i=1}^I$. For example, in the MAP approach, we seek

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\text{argmax}}\left[Pr(\boldsymbol{\theta}|w_{1\ldots I}, x_{1\ldots I})\right] \\
&= \underset{\boldsymbol{\theta}}{\text{argmax}}\left[Pr(w_{1\ldots I}|x_{1\ldots I}, \boldsymbol{\theta})Pr(\boldsymbol{\theta})\right] \\
&= \underset{\boldsymbol{\theta}}{\text{argmax}}\left[\prod_{i=1}^{I} Pr(w_i|x_i, \boldsymbol{\theta})Pr(\boldsymbol{\theta})\right],
\end{aligned} \tag{6.3}$$

where we have assumed that the I training pairs $\{x_i, w_i\}_{i=1}^I$ are independent, and defined a suitable prior $Pr(\boldsymbol{\theta})$.

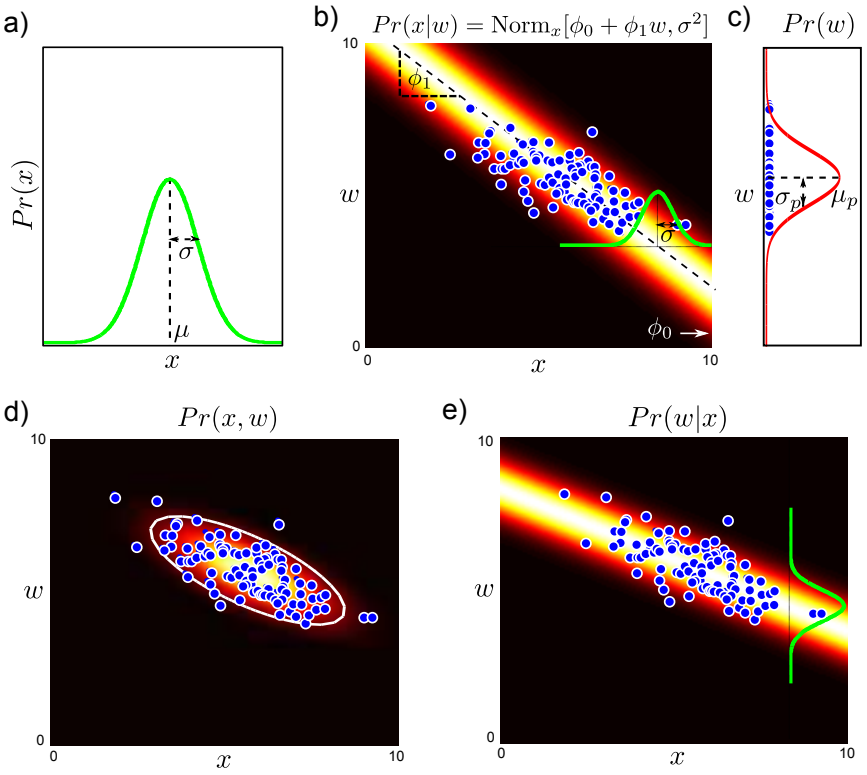

Figure 6.2 Regression by modeling likelihood $Pr(x|w)$ (generative). a) We represent the data x with a normal distribution. b) We make the normal parameters functions of the world state w. Here the mean is a linear function $\mu = \phi_0 + \phi_1 w$ of the world state and the variance σ^2 is fixed. The learning algorithm fits the parameters $\boldsymbol{\theta} = \{\phi_0, \phi_1, \sigma^2\}$ to example training pairs $\{x_i, w_i\}_{i=1}^{I}$ (blue dots). c) We also learn a prior distribution over the world state w (here modeled as a normal distribution with parameters $\boldsymbol{\theta}_p = \{\mu_p, \sigma_p\}$). In inference we take a new datum x and compute the posterior $Pr(w|x)$ over the state. d) This can be done by computing the joint distribution $Pr(x, w) = Pr(x|w)Pr(w)$ (weighting each row of (b) by the appropriate value from the prior) and e) normalizing the columns $Pr(w|x) = Pr(x, w)/Pr(x)$. Together these operations implement Bayes' rule: $Pr(w|x) = Pr(x|w)Pr(w)/Pr(x)$.

We also need an *inference algorithm* that takes visual data x and returns the posterior distribution $Pr(w|x, \boldsymbol{\theta})$. Here this is very simple: we simply evaluate equation 6.2 using the data x and the learned parameters $\hat{\boldsymbol{\theta}}$.

6.3.2 Model the contingency of data on world (generative)

In the generative formulation, we choose a probability distribution over the data x and make its parameters contingent on the world state w. Since the data are univariate and continuous, we will model the data as a normal distribution with fixed variance, σ^2 and a mean μ that is a linear function $\phi_0 + \phi_1 w$ of the world state (figure 6.2) so that

$$Pr(x|w, \boldsymbol{\theta}) = \text{Norm}_x\left[\phi_0 + \phi_1 w, \sigma^2\right]. \tag{6.4}$$

We also need a prior $Pr(w)$ over the world states which might also be normal so

$$Pr(w) = \text{Norm}_w[\mu_p, \sigma_p^2]. \tag{6.5}$$

The learning algorithm fits the parameters $\boldsymbol{\theta} = \{\phi_0, \phi_1, \sigma^2\}$ using paired training data $\{x_i, w_i\}_{i=1}^I$, and fits the parameters $\boldsymbol{\theta}_p = \{\mu_p, \sigma_p^2\}$ using the world states $\{w_i\}_{i=1}^I$. The inference algorithm takes a new datum x and returns the posterior $Pr(w|x)$ over the world state w using Bayes' rule

$$Pr(w|x) = \frac{Pr(x|w)Pr(w)}{Pr(x)} = \frac{Pr(x,w)}{Pr(x)}. \tag{6.6}$$

In this case, the posterior can be computed in closed form and is again normally distributed with fixed variance and a mean that is proportional to the data x.

Discussion

We have presented two models that can be used to estimate the world state w from an observed data example x, based on modeling the posterior $Pr(w|x)$ and the likelihood $Pr(x|w)$, respectively.

The models were carefully chosen so that they predict exactly the same posterior $P(w|x)$ over the world state (compare figures 6.1b and 6.2e). This is only the case with maximum likelihood learning: in the MAP approach we would have placed priors on the parameters, and because each model is parameterized differently, they would, in general, have different effects.

6.4 Example 2: binary classification

As a second example, we will consider the case where the observed measurement x is univariate and continuous, but the world state w is discrete and can take one of two values. For example, we might wish to classify a pixel as belonging to a skin or non-skin region based on observing just the red channel.

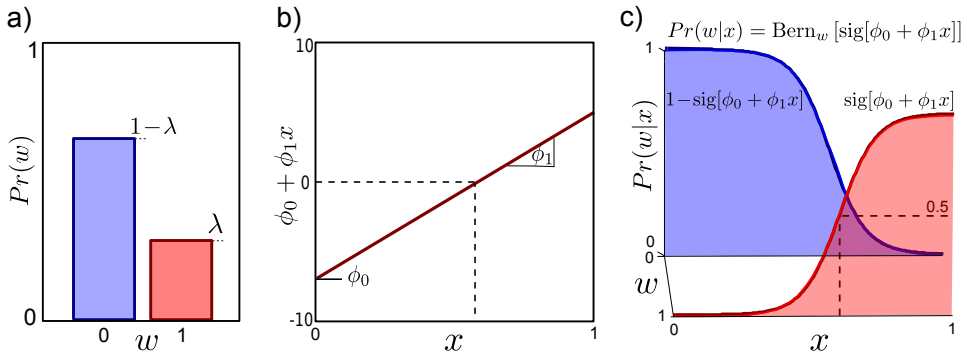

Figure 6.3 Classification by modeling posterior $Pr(w|x)$ (discriminative). a) We represent the world state w as a Bernoulli distribution. We make the Bernoulli parameter λ a function of the observations x. b) To this end we form a linear function $\phi_0 + \phi_1 x$ of the observations. c) The Bernoulli parameter $\lambda = \text{sig}[\phi_0 + \phi_1 x]$ is formed by passing the linear function through the logistic sigmoid $\text{sig}[\bullet]$ to constrain the value to lie between 0 and 1, giving the characteristic sigmoid shape (red curve). In learning we fit parameters $\boldsymbol{\theta} = \{\phi_0, \phi_1\}$ using example training pairs $\{x_i, w_i\}_{i=1}^I$. In inference we take a new datum x and evaluate the posterior $Pr(w|x)$ over the state.

6.4.1 Model contingency of world on data (discriminative)

We define a probability distribution over the world state $w \in \{0, 1\}$ and make its parameters contingent on the data x. Since the world state is discrete and binary, we will use a Bernoulli distribution. This has a single parameter λ, which determines the probability of success so that $Pr(w = 1) = \lambda$.

We make λ a function of the data x, but in doing so we must ensure the constraint $0 \leq \lambda \leq 1$ is obeyed. To this end, we form linear function $\phi_0 + \phi_1 x$ of the data x, which returns a value in the range $[-\infty \ \infty]$. We then pass the result through a function $\text{sig}[\bullet]$ that maps $[-\infty \ \infty]$ to $[0 \ 1]$, so that

$$Pr(w|x) = \text{Bern}_w[\text{sig}[\phi_0 + \phi_1 x]] = \text{Bern}_w\left[\frac{1}{1+\exp[-\phi_0 - \phi_1 x]}\right]. \tag{6.7}$$

This produces a sigmoidal dependence of the distribution parameter λ on the data x (figure 6.3). The function $\text{sig}[\bullet]$ is called the *logistic sigmoid*. This model is confusingly termed *logistic regression* despite being used here for classification.

In learning, we aim to fit the parameters $\boldsymbol{\theta} = \{\phi_0, \phi_1\}$ from paired training examples $\{x_i, w_i\}_{i=1}^I$. In inference, we simply substitute in the observed data value x into equation 6.7 to retrieve the posterior distribution $Pr(w|x)$ over the state.

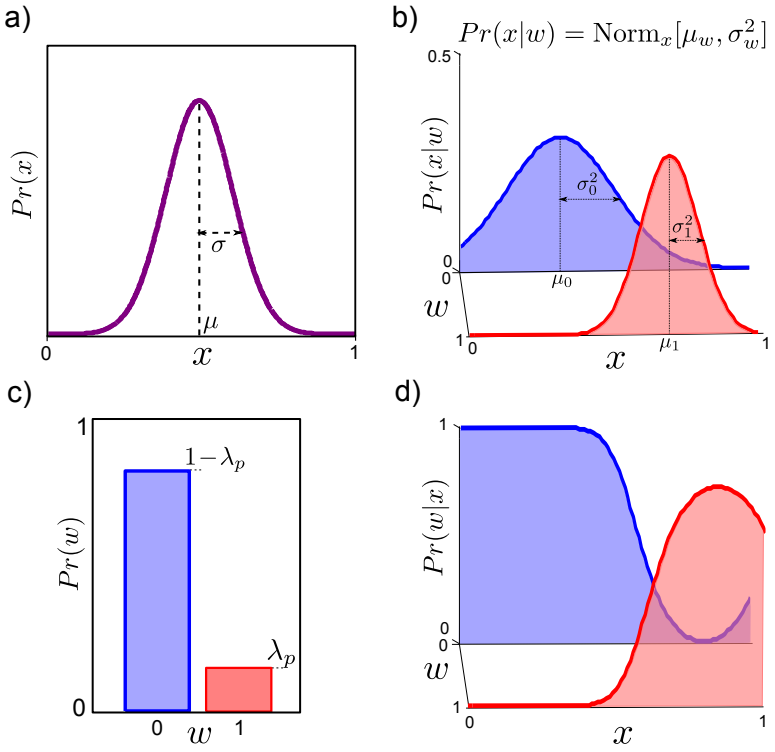

Figure 6.4 Classification by modeling the likelihood $Pr(x|w)$ (generative). a) We choose a normal distribution to represent the data x. b) We make the parameters $\{\mu, \sigma^2\}$ of this normal a function of the world state w. In practice, this means using one set of mean and variance parameters when the world state $w = 0$ and another when $w = 1$. The learning algorithm fits the parameters $\boldsymbol{\theta} = \{\mu_0, \mu_1, \sigma_0^2, \sigma_1^2\}$ to example training pairs $\{x_i, w_i\}_{i=1}^I$. c) We also model the prior probability of the world state w with a Bernoulli distribution with parameter λ_p. d) In inference we take a new datum x and compute the posterior $Pr(w|x)$ over the state using Bayes' rule.

6.4.2 Model contingency of data on world (generative)

Algorithm 6.1

We choose a probability distribution over the data x and make its parameters contingent on the world state w. Since the data are univariate and continuous, we will choose a univariate normal and allow the variance σ^2 and the mean μ to be functions of the binary world state w (figure 6.4) so that the likelihood is

Problem 6.7
Problem 6.8

$$Pr(x|w, \boldsymbol{\theta}) = \text{Norm}_x\left[\mu_w, \sigma_w^2\right]. \tag{6.8}$$

In practice this means that we have one set of parameters $\{\mu_0, \sigma_0^2\}$ when the state of the world is $w = 0$ and a different set $\{\mu_1, \sigma_1^2\}$ when the state is $w = 1$ so

$$\begin{aligned} Pr(x|w=0) &= \text{Norm}_x\left[\mu_0, \sigma_0^2\right] \\ Pr(x|w=1) &= \text{Norm}_x\left[\mu_1, \sigma_1^2\right]. \end{aligned} \tag{6.9}$$

These are referred to as *class conditional density functions* as they model the density of the data for each class separately.

We also define a prior distribution $Pr(w)$ over the world state,

$$Pr(w) = \text{Bern}_w[\lambda_p], \tag{6.10}$$

where λ_p is the prior probability of observing the state $w = 1$.

In learning, we fit the parameters $\boldsymbol{\theta} = \{\mu_0, \sigma_0^2, \mu_1, \sigma_1^2, \lambda_p\}$ using paired training data $\{x_i, w_i\}_{i=1}^I$. In practice, this consists of fitting the parameters μ_0 and σ_0^2 of the first class conditional density function $Pr(x|w=0)$ from just the data x where the state w was 0, and the parameters μ_1 and σ_1^2 of $P(x|w=1)$ from the data x where the state was 1. We learn the prior parameter λ_p from the training world states $\{w_i\}_{i=1}^I$.

The inference algorithm takes new datum x and returns the posterior distribution $Pr(w|x, \boldsymbol{\theta})$ over the world state w using Bayes' rule,

$$Pr(w|x) = \frac{Pr(x|w)Pr(w)}{\sum_{w=0}^{1} Pr(x|w)Pr(w)}. \tag{6.11}$$

This is very easy to compute; we evaluate the two class conditional density functions, weight each by the appropriate prior and normalize so that these two values sum to one.

Discussion

For binary classification, there is an asymmetry between the world state, which is discrete, and the measurements, which are continuous. Consequently, the generative and discriminative models look quite different, and the posteriors over the world state w as a function of the data x have different shapes (compare figure 6.3c with figure 6.4d). For the discriminative model, this function is by definition sigmoidal, but for the generative case it has a more complex form that was implicitly defined by the normal likelihoods. In general, choosing to model $Pr(w|x)$ or $P(x|w)$ will affect the expressiveness of the final model.

6.5 Which type of model should we use?

We have established that there are two different types of model that relate the world state and the observed data. But when should we use each type of model? There is no definitive answer to this question, but some considerations are:

Problem 6.9

- Inference is generally simpler with *discriminative* models. They directly model the conditional probability distribution of the world $Pr(\mathbf{w}|\mathbf{x})$ given

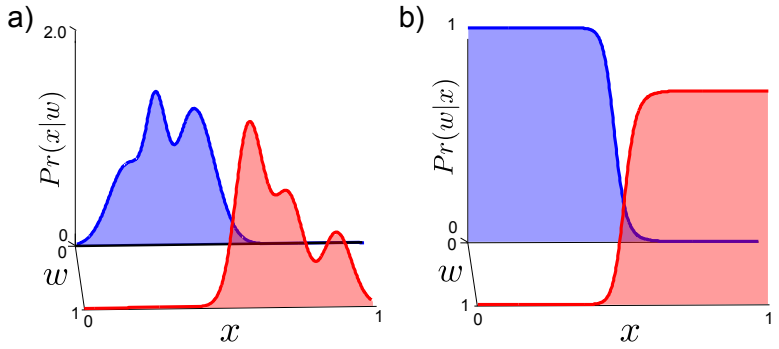

Figure 6.5 Generative vs. discriminative models. a) Generative approach: we separately model the probability $Pr(x|w)$ for each class. This may require a complex model with many parameters. b) Posterior probability distribution $Pr(w|x)$ computed via Bayes' rule with a uniform prior. Notice that the complicated structure of the individual class conditional density functions isn't reflected in the posterior: here, it would have been more efficient to take a discriminative approach and model this posterior directly.

the data. In contrast, generative models calculate this probability via Bayes' rule, and sometimes this requires a computationally expensive algorithm.

- *Generative methods* build probability models $Pr(\mathbf{x}|\mathbf{w})$ over the data whereas *discriminative models* just build a probability model $Pr(\mathbf{w}|\mathbf{x})$ over the world state. The data (usually an image) are generally of much higher dimension than the world state (some aspect of a scene), and modeling it is costly. Moreover, there may be many aspects of the data which do not influence the state; we might devote parameters to describing whether data configuration 1 is more likely than data configuration 2 although they both imply the same world state (figure 6.5).

- Modeling the likelihood $Pr(\mathbf{x}|\mathbf{w})$ mirrors the actual way that the data were created; the state of the world did create the observed data through some physical process (usually light being emitted from a source, interacting with the object and being captured by a camera). If we wish to build information about the generation process into our model, then this approach is desirable. For example, we can account for phenomena such as perspective projection and occlusion. Using the other approaches, it is harder to exploit this knowledge: essentially we have to re-learn these phenomena from the data.

- Sometimes parts of the training or test data vector $\mathbf{x}$ may be missing. Here, generative models are preferred. They model the joint distribution over all of the data dimensions and can effectively interpolate the missing elements.

- A fundamental property of the generative approach is that it allows incorporation of expert knowledge in the form of a prior. It is harder to impose prior knowledge in a principled way in discriminative models.

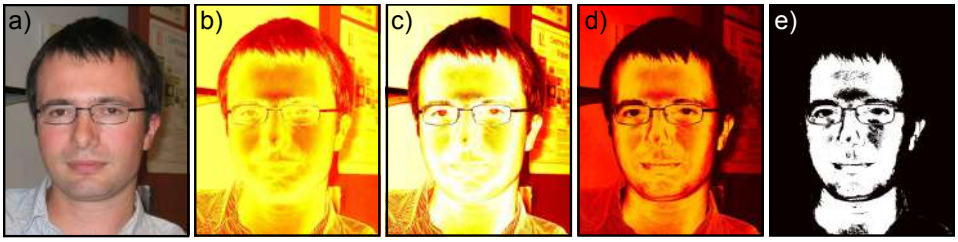

Figure 6.6 Skin detection. For each pixel we aim to infer a label $w \in \{0,1\}$ denoting the absence or presence of skin based on the RGB triple $\mathbf{x}$. Here we modeled the class conditional density functions $Pr(\mathbf{x}|w)$ as normal distributions. a) Original image. b) Log likelihood (log of data assessed under class-conditional density function) for non-skin. c) Log likelihood for skin. d) Posterior probability of belonging to skin class. e) Thresholded posterior probability $Pr(w|\mathbf{x}) > 0.5$ gives estimate of w.

It is notable that generative models are more common in vision applications. Consequently, most of the chapters in the rest of the book concern generative models.

6.6 Applications

The focus of this chapter, and indeed most of the chapters of this book, is on the models themselves and the learning and inference algorithms. From this point forward, we will devote a section at the end of each chapter to discussing practical applications of the relevant models in computer vision. For this chapter, only one of the models can actually be implemented based on the information presented so far. This is the generative classification model from section 6.4.2. Consequently, we will focus the applications on variations of this model and return to the other models in subsequent chapters.

6.6.1 Skin detection

The goal of skin-detection algorithms is to infer a label $w \in \{0,1\}$ denoting the presence or absence of skin at a pixel, based on the RGB measurements $\mathbf{x} = [x^R, x^G, x^B]$ at that pixel. This is a useful precursor to segmenting a face or hand, or it may be used as the basis of a crude method for detecting prurient content in web images. Taking a generative approach, we describe the likelihoods as

$$Pr(\mathbf{x}|w=k) \quad = \quad \text{Norm}_{\mathbf{x}}[\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k] \tag{6.12}$$

and the prior probability over states as

$$Pr(w) = \text{Bern}_w[\lambda]. \tag{6.13}$$

In the learning algorithm, we estimate the parameters $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_0, \boldsymbol{\Sigma}_1$ from training data pairs $\{w_i, \mathbf{x}_i\}_{i=1}^I$ where the pixels have been labeled by hand. In particular we learn $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ from the subset of the training data where $w_i = 0$ and $\boldsymbol{\mu}_1$ and $\boldsymbol{\Sigma}_1$ from the subset where $w_i = 1$. The prior parameter is learned from the world states $\{w_i\}_{i=1}^I$ alone. In each case, this involves fitting a probability distribution to data using one of the techniques discussed in chapter 4.

To classify a new data point $\mathbf{x}$ as skin or non-skin we apply Bayes' rule

$$Pr(w=1|\mathbf{x}) = \frac{Pr(\mathbf{x}|w=1)Pr(w=1)}{\sum_{k=0}^{1} Pr(\mathbf{x}|w=k)Pr(w=k)}, \tag{6.14}$$

and denote this pixel as skin if $Pr(w=1|\mathbf{x}) > 0.5$. Figure 6.6 shows the result of applying this model at each pixel independently in the image. Note that the classification is not perfect: there is genuinely an overlap between the skin- and non-skin distributions and this inevitably results in misclassified pixels. The results could be improved by exploiting the fact that skin areas tend to be contiguous regions without small holes. To do this, we must somehow connect together all of the per-pixel classifiers. This is the topic of chapters 11 and 12.

We briefly note that the RGB data are naturally discrete with $x^R, x^G, x^B \in \{0, 1, \ldots, 255\}$, and we could alternatively have based our skin detection model on this assumption. For example, modeling the three color channels independently, the likelihoods become

$$Pr(\mathbf{x}|w=k) = \text{Cat}_{x^R}[\boldsymbol{\lambda}_k^R]\text{Cat}_{x^G}[\boldsymbol{\lambda}_k^G]\text{Cat}_{x^B}[\boldsymbol{\lambda}_k^B]. \tag{6.15}$$

We refer to the assumption that the elements of the data vector are independent as *naïve Bayes*. Of course, it is not necessarily valid in the real world. To model the joint distribution of the R,G, and B components, we might combine them to form one variable with 256^3 entries and model this with a single categorical distribution. Unfortunately, this means we must learn 256^3 parameters for each categorical distribution, and so it is more practical to quantize each channel to fewer levels (say 8) before combining them together.

6.6.2 Background subtraction

Problem 6.10

A second application of the generative classification model is for background subtraction. Here, the goal is to infer a binary label $w_n \in \{0, 1\}$ which indicates whether the n^{th} pixel in the image is part of a known background ($w = 0$) or whether a foreground object is occluding it ($w = 1$). As for the skin detection model, this is based on its RGB pixel data $\mathbf{x}_n$ at that pixel.

It is usual to have training data $\{\mathbf{x}_{in}\}_{i=1,n=1}^{I,N}$ that consists of a number of empty scenes where all pixels are known to be background. However, it is not typical to have examples of the foreground objects which are highly variable in appearance.

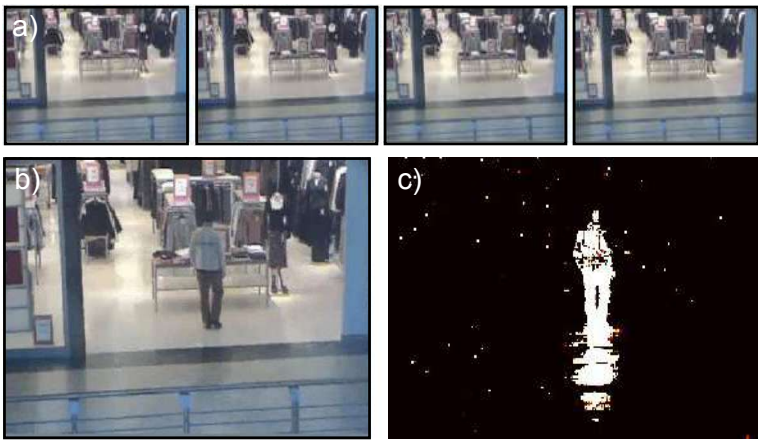

Figure 6.7 Background subtraction. For each pixel we aim to infer a label $w \in \{0, 1\}$ denoting the absence or presence of a foreground object. a) We learn a class conditional density model $Pr(\mathbf{x}|w)$ for the background from training examples of an empty scene. The foreground model is treated as uniform. b) For a new image, we then compute the posterior distribution using Bayes' rule. c) Posterior probability of being foreground $Pr(w = 1|\mathbf{x})$. Images from CAVIAR database.

For this reason, we model the class conditional distribution of the background as a normal distribution

$$Pr(\mathbf{x}_n|w = 0) = \text{Norm}_{\mathbf{x}_n}[\boldsymbol{\mu}_{n0}, \boldsymbol{\Sigma}_{n0}], \tag{6.16}$$

but model the foreground class as a uniform distribution

$$Pr(\mathbf{x}_n|w = 1) = \begin{cases} 1/255^3 & 0 < x_n^R, x_n^G, x_n^B < 255 \\ 0 & \text{otherwise} \end{cases}, \tag{6.17}$$

and again model the prior as a Bernoulli variable.

To compute the posterior distribution we once more apply Bayes' rule. Typical results are shown in figure 6.7, which illustrates a common problem with this method: shadows are often misclassified as foreground. A simple way to remedy this is to classify pixels based on the hue alone.

In some situations we need a more complex distribution to describe the background. For example, consider an outdoor scene in which trees are blowing in the wind (figure 6.8). Certain pixels may have bimodal distributions where one part of the foliage intermittently moves in front of another. It is clear that the unimodal normal likelihood cannot provide a good description of this data, and the resulting background segmentation result will be poor. We devote part of the next chapter to methods for describing more complex probability distributions of this type.

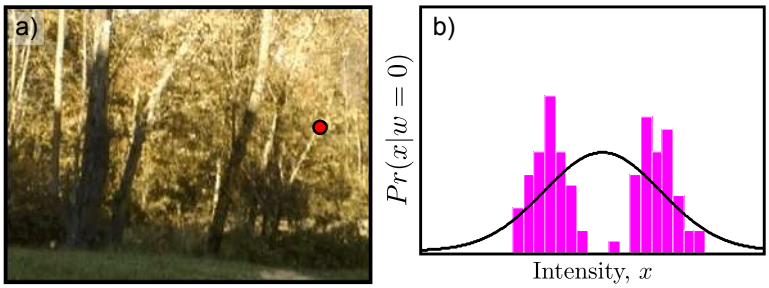

Figure 6.8 Background subtraction in deforming scene. a) The foliage is blowing in the wind in the training images. b) The distribution of RGB values at the pixel indicated by the circle in (a) is now bimodal and not well described by a normal density function (red channel only shown). Images from video by Terry Boult.

Summary

In this chapter, we have provided an overview of how abstract vision problems can be solved using machine learning techniques. We have illustrated these ideas with some simple examples. We did not provide the implementation level details of the learning and inference algorithms; these are presented in subsequent chapters.

	Model $Pr(w\|x)$	Model $Pr(x\|w)$
Regression $x \in [-\infty, \infty], w \in [-\infty, \infty]$	Linear regression	Linear regression
Classification $x \in [-\infty, \infty], w \in \{0, 1\}$	Logistic regression	Probability density function

Table 6.1: Example models in this chapter. These can be categorized into those that are based on modeling probability density functions, those that are based on linear regression, and those that are based on logistic regression.

The examples in this chapter are summarized in table 6.1, where it can be seen that there are three distinct types of model. First, there are those that depend on building probability density functions (describing the class conditional density functions $Pr(x|w = k)$). In the following chapter, we investigate building complex probability density models. The second type of model is based on linear regression, and chapter 8 investigates a family of related algorithms. Finally, the third type of model discussed in this chapter was logistic regression. We will elaborate on the logistic regression model in chapter 9.

Notes

The goal of this chapter was to give a very compact view of learning and inference in vision. Alternative views of this material which are not particularly aimed at vision can be found in Bishop (2006) and Duda *et al.* (2001) and many other texts.

Skin Detection: Reviews of skin detection can be found in Kakumanu *et al.* (2007) and Vezhnevets *et al.* (2003). Pixel-based skin-segmentation algorithms have been variously used as the basis for face detection (Hsu *et al.* 2002), hand gesture analysis (Zhu *et al.* 2000) and filtering of pornographic images (Jones & Rehg 2002).

There are two main issues that affect the quality of the final results: the representation of the pixel color and the classification algorithm. With regard to the latter issue, various generative approaches have been investigated, including methods based on normal distributions (Hsu *et al.* 2002), mixtures of normal distributions (Jones & Rehg 2002) and categorical distributions (Jones & Rehg 2002) as well as discriminative methods such as the multi-layer perceptron (Phung *et al.* 2005). There are several detailed empirical studies that compare the efficacy of the color representation and classification algorithm (Phung *et al.* 2005; Brand & Mason 2000; Schmugge *et al.* 2007)

Background Subtraction: Reviews of background subtraction techniques can be found in Piccardi (2004), Bouwmans *et al.* (2010), and Elgammal (2011). Background subtraction is a common first step in many vision systems as it quickly identifies regions of the image that are of interest. Generative classification systems have been built based on normal distributions (Wren *et al.* 1997), mixtures of normal distribution (Stauffer & Grimson 1999), and kernel density functions (Elgammal *et al.* 2000). Several systems (Friedman & Russell 1997; Horprasert *et al.* 2000) have incorporated an explicit label in the model to identify shadows.

Most recent research in this area has addressed maintenance of the background model in changing environments. Many systems such as that of Stauffer & Grimson (1999) are adaptive and can incorporate new objects into the background model when the background changes. Other models compensate for lighting changes by exploiting the fact that all of the background pixels change together and describing this covariance with a subspace model (Oliver *et al.* 2000). It is also common now to abandon the per-pixel approach and to estimate the whole label field simultaneously using a technique based on Markov random fields (e.g., Sun *et al.* 2006).

Problems

Problem 6.1 Consider the following problems.

i Determining the gender of an image of a face.
ii Determining the pose of the human body given an image of the body.
iii Determining which suit a playing card belongs to based on an image of that card.
iv Determining whether two images of faces match (face verification).
v Determining the 3D position of a point given the positions to which it projects in two cameras at different positions in the world (stereo reconstruction).

For each case, try to describe the contents of the world state $\mathbf{w}$ and the data $\mathbf{x}$. Is each discrete or continuous? If discrete, then how many values can it take? Which are regression problems and which are classification problems?

Problem 6.2 Describe a classifier that relates univariate discrete data $x \in \{1 \ldots K\}$ to a univariate discrete world state $w \in \{1 \ldots M\}$ for both discriminative and generative model types.

Problem 6.3 Describe a regression model that relates univariate binary discrete data $x \in \{0,1\}$ to a univariate continuous world state $w \in [-\infty, \infty]$. Use a generative formulation in which $Pr(x|w)$ and $Pr(w)$ are modeled.

Problem 6.4 Describe a discriminative regression model that relates a continuous world state $w \in [0,1]$ to univariate continuous data $x \in [-\infty, \infty]$. Hint: Base your classifier on the Beta distribution. Ensure that the constraints on the parameters are obeyed.

Problem 6.5 Find expressions for the maximum likelihood estimates of the parameters in the discriminative linear regression model (section 6.3.1). In other words find the parameters $\{\phi_0, \phi_1, \sigma^2\}$ that satisfy

$$\begin{aligned}\hat{\phi}_0, \hat{\phi}_1, \hat{\sigma}^2 &= \underset{\phi_0,\phi_1,\sigma^2}{\text{argmax}}\left[\prod_{i=1}^{I} Pr(w_i|x_i, \phi_0, \phi_1, \sigma^2)\right] \\ &= \underset{\phi_0,\phi_1,\sigma^2}{\text{argmax}}\left[\sum_{i=1}^{I} \log\left[Pr(w_i|x_i, \phi_0, \phi_1, \sigma^2)\right]\right] \\ &= \underset{\phi_0,\phi_1,\sigma^2}{\text{argmax}}\left[\sum_{i=1}^{I} \log\left[\text{Norm}_w\left[\phi_0 + \phi_1 x, \sigma^2\right]\right]\right],\end{aligned}$$

where $\{w_i, x_i\}_{i=1}^{I}$ are paired training examples.

Problem 6.6 Consider a regression model that models the joint probability $Pr(x, w)$ between the world w and the data x as

$$Pr\left(\begin{bmatrix} w_i \\ x_i \end{bmatrix}\right) = \text{Norm}_{[w_i, x_i]^T}\left[\begin{bmatrix} \mu_w \\ \mu_x \end{bmatrix}, \begin{bmatrix} \sigma_{ww}^2 & \sigma_{xw}^2 \\ \sigma_{xw}^2 & \sigma_{xx} \end{bmatrix}\right].$$

Use the relation in section 5.5 to compute the posterior distribution $Pr(w_i|x_i)$. Show that it has the form

$$Pr(w_i|x_i) = \text{Norm}_{w_i}[\phi_0 + \phi_1 x, \sigma^2],$$

and compute expressions for ϕ_0 and ϕ_1 in terms of the training data $\{w_i, x_i\}_{i=1}^{I}$ by substituting in explicit maximum likelihood estimates of the parameters $\{\mu_w, \mu_x, \sigma_{ww}^2, \sigma_{xw}^2, \sigma_{xx}^2\}$.

Problem 6.7 For a two-class problem, the *decision boundary* is the locus of world values w where the posterior probability $Pr(w = 1|x)$ is equal to 0.5. In other words, it represents the boundary between regions that would be classified as $w = 0$ and $w = 1$. Consider the generative classifier from section 6.4.2. Show that with equal priors $Pr(w = 0) = Pr(w = 1) = 0.5$ points on the decision boundary (the locus of points where $Pr(w = 0|x) = Pr(w = 1|x)$) obey a constraint of the form

$$ax^2 + bx + c = 0,$$

where and $\{a, b, c\}$ are scalars. Does the shape of the decision boundary for the logistic regression model from section 6.4.1 have the same form?

Problem 6.8 Consider a generative classification model for 1D data with likelihood terms

$$\begin{aligned}Pr(x_i|w_i = 0) &= \text{Norm}_{x_i}\left[0, \sigma^2\right] \\ Pr(x_i|w_i = 1) &= \text{Norm}_{x_i}\left[0, 1.5\sigma^2\right].\end{aligned}$$

What is the decision boundary for this classifier with equal priors $Pr(w=0) = Pr(w=1) = 0.5$? Develop a discriminative classifier that can produce the same decision boundary. Hint: base your classifier on a quadratic rather than a linear function.

Problem 6.9 Consider a generative binary classifier for multivariate data based on multivariate normal likelihood terms

$$\begin{aligned} Pr(\mathbf{x}_i|w_i=0) &= \text{Norm}_{\mathbf{x}_i}\left[\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right] \\ Pr(\mathbf{x}_i|w_i=1) &= \text{Norm}_{\mathbf{x}_i}\left[\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right] \end{aligned}$$

and a discriminative classifier based on logistic regression for the same data

$$Pr(w_i|\mathbf{x}_i) = \text{Bern}_{w_i}\left[\text{sig}[\phi_0 + \boldsymbol{\phi}^T\mathbf{x}_i]\right].$$

where there is one entry in the gradient vector $\boldsymbol{\phi}$ for each entry of $\mathbf{x}_i$.

How many parameters does each model have as a function of the dimensionality of $\mathbf{x}_i$? What are the relative advantages and disadvantages of each model as the dimensionality increases?

Problem 6.10 One of the problems with the background subtraction method described is that it erroneously classifies shadows as foreground. Describe a model that could be used to classify pixels into three categories (foreground, background, and shadow).