Chapter 8

Regression models

This chapter concerns regression problems: the goal is to estimate a univariate world state w based on observed measurements $\mathbf{x}$. The discussion is limited to discriminative methods in which the distribution $Pr(w|\mathbf{x})$ of the world state is directly modeled. This contrasts with chapter 7 where the focus was on generative models in which the likelihood $Pr(\mathbf{x}|w)$ of the observations is modeled.

To motivate the regression problem, consider *body pose estimation*: here the goal is to estimate the joint angles of a human body, based on an observed image of the person in an unknown pose (figure 8.1). Such an analysis could form the first step toward activity recognition.

We assume that the image has already been preprocessed and a low dimensional vector $\mathbf{x}$ that represents the shape of the contour has been extracted. Our goal is to use this data vector to predict a second vector containing the joint angles for each of the major body joints. In practice, we will estimate each joint angle separately; we can hence concentrate our discussion on how to estimate a univariate quantity w from continuous observed data $\mathbf{x}$. We begin by assuming that the relation between the world and the data is linear and that the uncertainty around this prediction is normally distributed with constant variance. This is the linear regression model.

8.1 Linear regression

The goal of linear regression is to predict the posterior distribution $Pr(w|\mathbf{x})$ over the world state w based on observed data $\mathbf{x}$. Since this is a discriminative model, we proceed by choosing a probability distribution over the world w and making the parameters dependent on the data $\mathbf{x}$. The world state w is univariate and continuous and so a suitable distribution is the univariate normal. In linear regression (figure 8.2), we make the mean μ of this normal distribution a linear function $\phi_0 + \boldsymbol{\phi}^T\mathbf{x}_i$ of the data and treat the variance σ^2 as a constant so that

$$Pr(w_i|\mathbf{x}_i, \boldsymbol{\theta}) = \text{Norm}_{w_i}\left[\phi_0 + \boldsymbol{\phi}^T\mathbf{x}_i, \sigma^2\right], \tag{8.1}$$

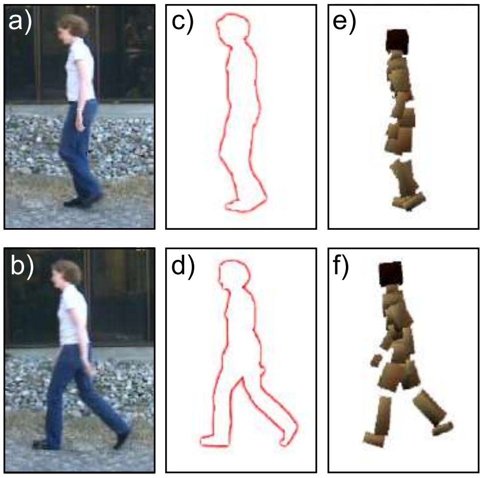

Figure 8.1 Body pose estimation. a–b) Human beings in unknown poses. c–d) The silhouette is found by segmenting the image and the contour extracted by tracing around the edge of the silhouette. A 100D $\mathbf{x}$ is extracted that describes the contour shape based on the shape context descriptor (see section 13.3.5). e–f) The goal is to estimate the vector $\mathbf{w}$ containing the major joint angles of the body. This is a regression problem as each element of the world state $\mathbf{w}$ is continuous. Adapted from Agarwal & Triggs (2006).

where $\boldsymbol{\theta} = \{\phi_0, \boldsymbol{\phi}, \sigma^2\}$ are the model parameters. The term ϕ_0 can be interpreted as the y-intercept of a hyperplane and the entries of $\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_D]^T$ are its gradients with respect to each of the D data dimensions.

It is cumbersome to treat the y-intercept separately from the gradients, so we apply a trick that allows us to simplify the subsequent notation. We attach a 1 to the start of every data vector $\mathbf{x}_i \leftarrow [1 \quad \mathbf{x}_i^T]^T$ and attach the y-intercept ϕ_0 to the start of the gradient vector $\boldsymbol{\phi} \leftarrow [\phi_0 \quad \boldsymbol{\phi}^T]^T$ so that we can now equivalently write

$$Pr(w_i|\mathbf{x}_i, \boldsymbol{\theta}) = \text{Norm}_{w_i}\left[\boldsymbol{\phi}^T\mathbf{x}_i, \sigma^2\right]. \tag{8.2}$$

In fact, since each training data example is considered independent, we can write the probability $Pr(\mathbf{w}|\mathbf{X})$ of the entire training set as a single normal distribution with a diagonal covariance so that

$$Pr(\mathbf{w}|\mathbf{X}) = \text{Norm}_{\mathbf{w}}[\mathbf{X}^T\boldsymbol{\phi}, \sigma^2\mathbf{I}], \tag{8.3}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I]$ and $\mathbf{w} = [w_1, w_2, \ldots, w_I]^T$.

Inference for this model is very simple: for a new datum $\mathbf{x}^*$ we simply evaluate equation 8.2 to find the posterior distribution $Pr(w^*|\mathbf{x}^*)$ over the world state w^*. Hence we turn our main focus to learning.

8.1.1 Learning

Algorithm 7.1

The learning algorithm estimates the model parameters $\boldsymbol{\theta} = \{\boldsymbol{\phi}, \sigma^2\}$ from paired training examples $\{\mathbf{x}_i, w_i\}_{i=1}^I$. In the maximum likelihood approach we seek

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\text{argmax}}\left[Pr(\mathbf{w}|\mathbf{X}, \boldsymbol{\theta})\right] \\ &= \underset{\boldsymbol{\theta}}{\text{argmax}}\left[\log[Pr(\mathbf{w}|\mathbf{X}, \boldsymbol{\theta})]\right],\end{aligned} \tag{8.4}$$

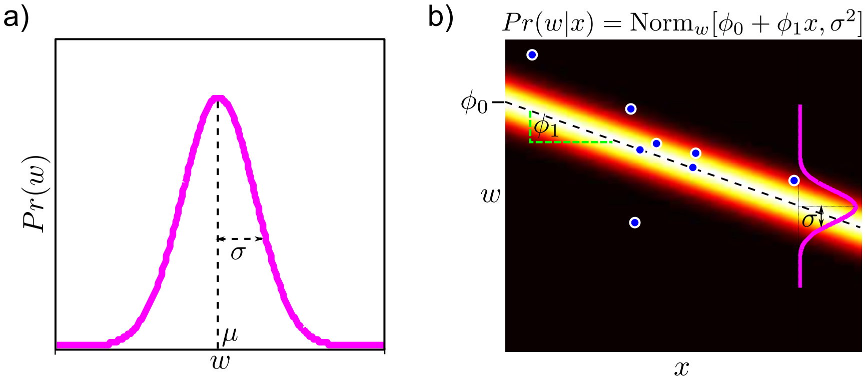

Figure 8.2 Linear regression model with univariate data x. a) We choose a univariate normal distribution over the world state w. b) The parameters of this distribution are now made to depend on the data x: the mean μ is a linear function $\phi_0 + \phi_1 x$ of the data and the variance σ^2 is constant. The parameters ϕ_0 and ϕ_1 represent the intercept and slope of the linear function, respectively.

where as usual we have taken the logarithm of the criterion. The logarithm is a monotonic transformation, and so it does not change the position of the maximum, but the resulting cost function is easier to optimize. Substituting in we find that

$$\hat{\boldsymbol{\phi}}, \hat{\sigma}^2 = \underset{\boldsymbol{\phi},\sigma^2}{\operatorname{argmax}} \left[-\frac{I\log[2\pi]}{2} - \frac{I\log[\sigma^2]}{2} - \frac{(\mathbf{w}-\mathbf{X}^T\boldsymbol{\phi})^T(\mathbf{w}-\mathbf{X}^T\boldsymbol{\phi})}{2\sigma^2} \right]. \tag{8.5}$$

We now take the derivatives with respect to $\boldsymbol{\phi}$ and σ^2, equate the resulting expressions to zero and solve to find

Problem 8.3

$$\begin{aligned} \hat{\boldsymbol{\phi}} &= (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{w} \\ \hat{\sigma}^2 &= \frac{(\mathbf{w}-\mathbf{X}^T\boldsymbol{\phi})^T(\mathbf{w}-\mathbf{X}^T\boldsymbol{\phi})}{I}. \end{aligned} \tag{8.6}$$

Figure 8.2b shows an example fit with univariate data x. In this case, the model describes the data reasonably well.

8.1.2 Problems with the linear regression model

There are three main limitations of the linear regression model.

- The predictions of the model are overconfident; for example, small changes in the estimated slope ϕ_1 make increasingly large changes in the predictions as we move further from the y-intercept ϕ_0. However, this is not reflected in the posterior distribution.

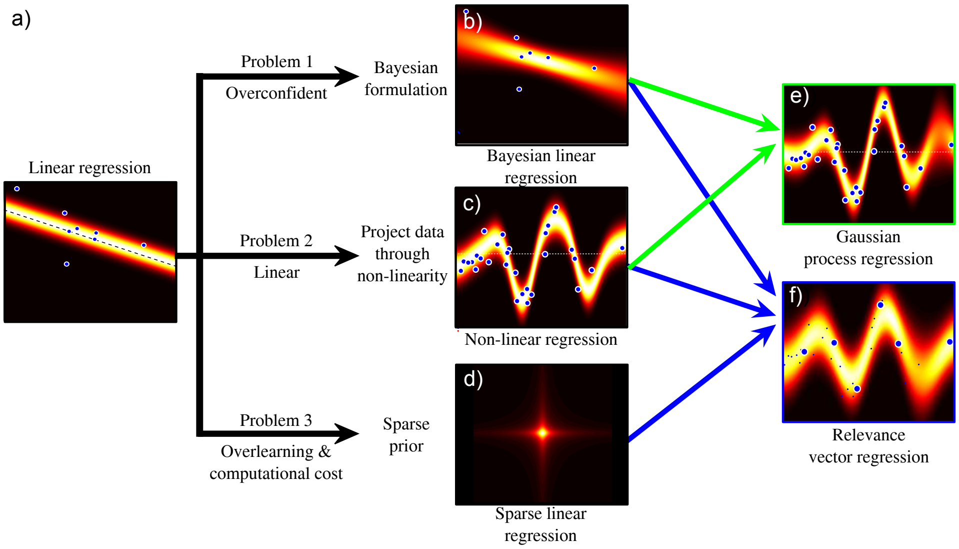

Figure 8.3 Family of regression models. There are several limitations to linear regression which we deal with in subsequent sections. The linear regression model with maximum likelihood learning is overconfident, and hence we develop a Bayesian version. It is unrealistic to always assume a linear relationship between the data and the world and to this end, we introduce a nonlinear version. The linear regression model has many parameters when the data dimension is high, and hence we consider a sparse version of the model. The ideas of Bayesian estimation, nonlinear functions and sparsity are variously combined to form the Gaussian process regression and relevance vector regression model.

- We are limited to linear functions and usually there is no particular reason that the visual data and world state should be linearly related.

- When the observed data $\mathbf{x}$ is high=dimensional, it may be that many elements of this variable aren't useful for predicting the state of the world, and so the resulting model is unnecessarily complex.

We tackle each of these problems in turn. In the following section we address the overconfidence of the model by developing a Bayesian approach to the same problem. In section 8.3, we generalize this model to fit nonlinear functions. In section 8.6, we introduce a sparse version of the regression model where most of the weighting coefficients $\boldsymbol{\phi}$ are encouraged to be zero. The relationships between the models in this chapter are indicated in figure 8.3.

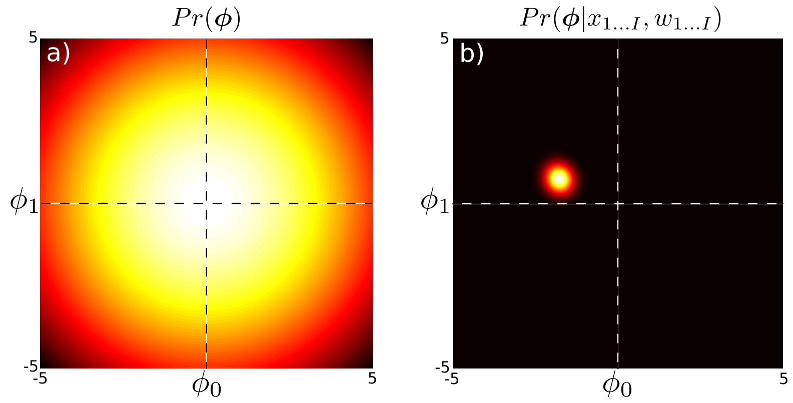

Figure 8.4 Bayesian linear regression. a) Prior $Pr(\boldsymbol{\phi})$ over intercept ϕ_0 and slope ϕ_1 parameters. This represents our knowledge about the parameters before we observe the data. b) Posterior distribution $Pr(\boldsymbol{\phi}|\mathbf{X}, \mathbf{w})$ over intercept and slope parameters. This represents our knowledge about the parameters after observing the data from figure 8.2b: we are considerably more certain but there remain a range of possible parameter values.

8.2 Bayesian linear regression

In the Bayesian approach, we compute a probability distribution over possible values of the parameters $\boldsymbol{\phi}$ (we will assume for now that σ^2 is known, see section 8.2.2). When we evaluate the probability of new data, we take a weighted average of the predictions induced by the different possible values.

Algorithm 7.2

Since the gradient vector $\boldsymbol{\phi}$ is multivariate and continuous, we model the prior $Pr(\boldsymbol{\phi})$ as normal with zero mean and spherical covariance,

$$Pr(\boldsymbol{\phi}) = \text{Norm}_{\boldsymbol{\phi}}[\mathbf{0}, \sigma_p^2\mathbf{I}], \tag{8.7}$$

where σ_p^2 scales the prior covariance and $\mathbf{I}$ is the identity matrix. Typically σ_p^2 is set to a large value to reflect the fact that our prior knowledge is weak.

Given paired training examples $\{\mathbf{x}_i, w_i\}_{i=1}^I$, the posterior distribution over the parameters can be computed using Bayes' rule

$$Pr(\boldsymbol{\phi}|\mathbf{X}, \mathbf{w}) = \frac{Pr(\mathbf{w}|\mathbf{X}, \boldsymbol{\phi})Pr(\boldsymbol{\phi})}{Pr(\mathbf{w}|\mathbf{X})}, \tag{8.8}$$

where, as before, the likelihood is given by

$$Pr(\mathbf{w}|\mathbf{X}, \boldsymbol{\theta}) \quad = \quad \text{Norm}_{\mathbf{w}}\left[\mathbf{X}^T\boldsymbol{\phi}, \sigma^2\mathbf{I}\right]. \tag{8.9}$$

The posterior distribution can be computed in closed form (using the relations in sections 5.7 and 5.6) and is given by the expression:

Problem 8.4

$$Pr(\boldsymbol{\phi}|\mathbf{X},\mathbf{w}) = \text{Norm}_{\boldsymbol{\phi}}\left[\frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{w}, \mathbf{A}^{-1}\right], \tag{8.10}$$

where

$$\mathbf{A} = \frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \frac{1}{\sigma_p^2}\mathbf{I}. \tag{8.11}$$

Note that the posterior distribution $Pr(\boldsymbol{\phi}|\mathbf{X},\mathbf{w})$ is always narrower than the prior distribution $Pr(\boldsymbol{\phi})$ (figure 8.4); the data provides information that refines our knowledge of the parameter values.

Problem 8.5

We now turn to the problem of computing the predictive distribution over the world state w^* for a new observed data vector $\mathbf{x}^*$. We take an infinite weighted sum (i.e., an integral) over the predictions $Pr(w^*|\mathbf{x}^*,\boldsymbol{\phi})$ implied by each possible $\boldsymbol{\phi}$ where the weights are given by the posterior distribution $Pr(\boldsymbol{\phi}|\mathbf{X},\mathbf{w})$.

$$\begin{aligned} Pr(w^*|\mathbf{x}^*,\mathbf{X},\mathbf{w}) &= \int Pr(w^*|\mathbf{x}^*,\boldsymbol{\phi})Pr(\boldsymbol{\phi}|\mathbf{X},\mathbf{w})d\boldsymbol{\phi} \\ &= \int \text{Norm}_{w^*}[\boldsymbol{\phi}^T\mathbf{x}^*, \sigma^2]\text{Norm}_{\boldsymbol{\phi}}\left[\frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{w}, \mathbf{A}^{-1}\right]d\boldsymbol{\phi} \\ &= \text{Norm}_{w^*}\left[\frac{1}{\sigma^2}\mathbf{x}^{*T}\mathbf{A}^{-1}\mathbf{X}\mathbf{w}, \mathbf{x}^{*T}\mathbf{A}^{-1}\mathbf{x}^* + \sigma^2\right]. \end{aligned} \tag{8.12}$$

To compute this, we reformulated the integrand using the relations from sections 5.7 and 5.6 as the product of a normal distribution in $\boldsymbol{\phi}$ and a constant with respect to $\boldsymbol{\phi}$. The integral of the normal distribution must be one, and so the final result is just the constant. This constant is itself a normal distribution in w^*.

This Bayesian formulation of linear regression (figure 8.5) is less confident about its predictions, and the confidence decreases as the test data $\mathbf{x}^*$ departs from the mean $\overline{\mathbf{x}}$ of the observed data. This is because uncertainty in the gradient causes increasing uncertainty in the predictions as we move further away from the bulk of the data. This agrees with our intuitions: predictions ought to become less confident as we extrapolate further from the data.

8.2.1 Practical concerns

To implement this model we must compute the $D \times D$ matrix inverse $\mathbf{A}^{-1}$ (equation 8.12). If the dimension D of the original data is large, then it will be difficult to compute this inverse directly.

Fortunately, the structure of $\mathbf{A}$ is such that it can be inverted far more efficiently. We exploit the Woodbury identity (see appendix C.8.4), to rewrite $\mathbf{A}^{-1}$ as

$$\mathbf{A}^{-1} = \left(\frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \frac{1}{\sigma_p^2}\mathbf{I}_D\right)^{-1} = \sigma_p^2\mathbf{I}_D - \sigma_p^2\mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \frac{\sigma^2}{\sigma_p^2}\mathbf{I}_I\right)^{-1}\mathbf{X}^T, \tag{8.13}$$

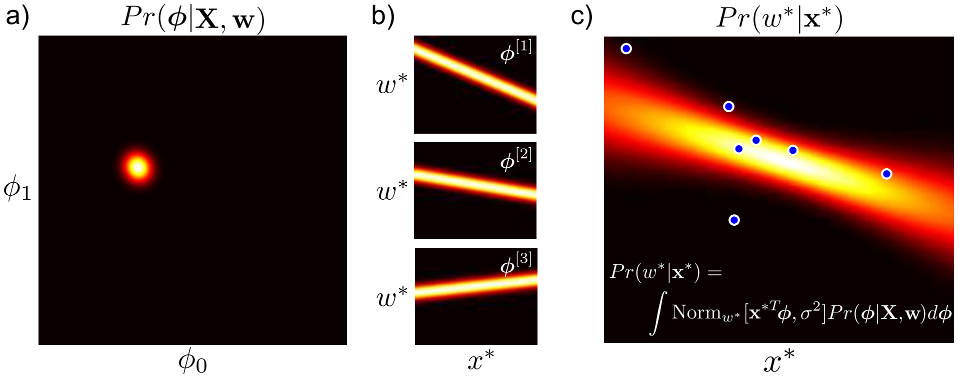

Figure 8.5 Bayesian linear regression. a) In learning we compute the posterior distribution $Pr(\boldsymbol{\phi}|\mathbf{X},\mathbf{w})$ over the intercept and slope parameters: there is a family of parameter settings that are compatible with the data. b) Three samples from the posterior, each of which corresponds to a different regression line. c) To form the predictive distribution we take an infinite weighted sum (integral) of the predictions from all of the possible parameter settings, where the weight is given by the posterior probability. The individual predictions vary more as we move from the centroid $\overline{\mathbf{x}}$ and this is reflected in the fact that the certainty is lower on either side of the plot.

where we have explicitly noted the dimensionality of each of the identity matrices $\mathbf{I}$ as a subscript. The expression still includes an inversion, but now it is of size $I \times I$ where I is the number of examples. If the number of examples I is fewer than the number of data dimensions D, then this formulation is more practical. This formulation also demonstrates clearly that the posterior covariance is less than the prior; the posterior covariance is the prior covariance σ_p^2 with a data-dependent term subtracted from it.

Substituting the new expression for $\mathbf{A}^{-1}$ into equation 8.12, we derive a new expression for the predictive distribution,

$$\begin{aligned} Pr(w^*|\mathbf{x}^*,\mathbf{X},\mathbf{w}) = & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (8.14)\\ & \text{Norm}_{w^*}\left[\frac{\sigma_p^2}{\sigma^2}\mathbf{x}^{*T}\mathbf{X}\mathbf{w} - \frac{\sigma_p^2}{\sigma^2}\mathbf{x}^{*T}\mathbf{X}\left(\mathbf{X}^T\mathbf{X}+\frac{\sigma^2}{\sigma_p^2}\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w},\right.\\ & \qquad\left.\sigma_p^2\mathbf{x}^{*T}\mathbf{x}^* - \sigma_p^2\mathbf{x}^{*T}\mathbf{X}\left(\mathbf{X}^T\mathbf{X}+\frac{\sigma^2}{\sigma_p^2}\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{x}^* + \sigma^2\right]. \end{aligned}$$

It is notable that only inner products of the data vectors (e.g., in the terms $\mathbf{X}^T\mathbf{x}^*$, or $\mathbf{X}^T\mathbf{X}$) are required to compute this expression. We will exploit this fact when we generalize these ideas to nonlinear regression (section 8.3).

8.2.2 Fitting the variance

Problem 8.7

The previous analysis has concentrated exclusively on the slope parameters $\boldsymbol{\phi}$. In principle, we could have taken a Bayesian approach to estimating the variance parameter σ^2 as well. However, for simplicity we will compute a point estimate of σ^2 using the maximum likelihood approach. To this end, we optimize the *marginal likelihood*, which is the likelihood after marginalizing out $\boldsymbol{\phi}$ and is given by

$$\begin{aligned} Pr(\mathbf{w}|\mathbf{X},\sigma^2) &= \int Pr(\mathbf{w}|\mathbf{X},\boldsymbol{\phi},\sigma^2)Pr(\boldsymbol{\phi})\,d\boldsymbol{\phi} \\ &= \int \text{Norm}_{\mathbf{w}}[\mathbf{X}^T\boldsymbol{\phi},\sigma^2\mathbf{I}]\text{Norm}_{\boldsymbol{\phi}}[\mathbf{0},\sigma_p^2\mathbf{I}]\,d\boldsymbol{\phi} \\ &= \text{Norm}_{\mathbf{w}}[\mathbf{0},\sigma_p^2\mathbf{X}^T\mathbf{X}+\sigma^2\mathbf{I}] \qquad (8.15) \end{aligned}$$

where the integral was solved using the same technique as for equation 8.12.

To estimate the variance, we maximize the log of this expression with respect to σ^2. Since the unknown is a scalar it is straightforward to optimize this function by simply evaluating the function over a range of values and choosing the maximum. Alternatively, we could use a general purpose nonlinear optimization technique (see appendix B).

8.3 Non-linear regression

Problem 8.9

It is unrealistic to assume that there is always a linear relationship between the world state w and the input data $\mathbf{x}$. In developing an approach to nonlinear regression, we would like to retain the mathematical convenience of the linear model while extending the class of functions that can be described.

Consequently, the approach that we describe is extremely simple: we first pass each data example through a nonlinear transformation

$$\mathbf{z}_i = \mathbf{f}[\mathbf{x}_i], \qquad (8.16)$$

to create a new data vector $\mathbf{z}_i$ which is usually higher dimensional than the original data. Then we proceed as before: we describe the mean of the posterior distribution $Pr(w_i|\mathbf{x}_i)$ as a linear function $\boldsymbol{\phi}^T\mathbf{z}_i$ of the transformed measurements so that

$$Pr(w_i|\mathbf{x}_i,\boldsymbol{\theta}) = \text{Norm}_{w_i}[\boldsymbol{\phi}^T\mathbf{z}_i,\sigma^2]. \qquad (8.17)$$

For example, consider the case of 1D polynomial regression:

$$Pr(w_i|x_i) = \text{Norm}_{w_i}[\phi_0+\phi_1x_i+\phi_2x_i^2+\phi_3x_i^3,\sigma^2]. \qquad (8.18)$$

This model can be considered as computing the nonlinear transformation

$$\mathbf{z}_i = \begin{bmatrix} 1 \\ x_i \\ x_i^2 \\ x_i^3 \end{bmatrix}, \tag{8.19}$$

and so it has the general form of equation 8.17.

8.3.1 Maximum likelihood

To find the maximum likelihood solution for the gradient vector $\boldsymbol{\phi}$ we first combine all of the transformed training data relations (equation 8.17) into a single expression:

$$Pr(\mathbf{w}|\mathbf{X}) = \text{Norm}_{\mathbf{w}}[\mathbf{Z}^T\boldsymbol{\phi}, \sigma^2\mathbf{I}]. \tag{8.20}$$

The optimal weights can now be computed as

$$\begin{aligned} \hat{\boldsymbol{\phi}} &= (\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{w} \\ \hat{\sigma}^2 &= \frac{(\mathbf{w}-\mathbf{Z}^T\boldsymbol{\phi})^T(\mathbf{w}-\mathbf{Z}^T\boldsymbol{\phi})}{I}, \end{aligned} \tag{8.21}$$

where the matrix $\mathbf{Z}$ contains the transformed vectors $\{\mathbf{z}_i\}_{i=1}^I$ in its columns. These equations were derived by replacing the original data term $\mathbf{X}$ by the transformed data $\mathbf{Z}$ in the equivalent linear expressions (equation 8.6). For a new observed data example $\mathbf{x}^*$ we compute the vector $\mathbf{z}^*$ and then evaluate equation 8.17.

Figures 8.6 and 8.7 provide two more examples of this approach. In figure 8.6, the new vector $\mathbf{z}$ is computed by evaluating the data $\mathbf{x}$ under a set of radial basis functions:

$$\mathbf{z}_i = \begin{bmatrix} 1 \\ \exp\left[-(x_i-\alpha_1)^2/\lambda\right] \\ \exp\left[-(x_i-\alpha_2)^2/\lambda\right] \\ \exp\left[-(x_i-\alpha_3)^2/\lambda\right] \\ \exp\left[-(x_i-\alpha_4)^2/\lambda\right] \\ \exp\left[-(x_i-\alpha_5)^2/\lambda\right] \\ \exp\left[-(x_i-\alpha_6)^2/\lambda\right] \end{bmatrix}. \tag{8.22}$$

The term *radial basis functions* can be used to denote any spherically symmetric function, and here we have used the Gaussian. The parameters $\{\alpha_k\}_{k=1}^K$ are the centers of the functions, and λ is a scaling factor that determines their width. The functions themselves are shown in figure 8.6b. Because they are spatially localized, each one accounts for a part of the original data space. We can approximate a function by taking weighted sums $\boldsymbol{\phi}^T\mathbf{z}$ of these functions. For example, when they are weighted as in figure 8.6c, they create the function in figure 8.6a.

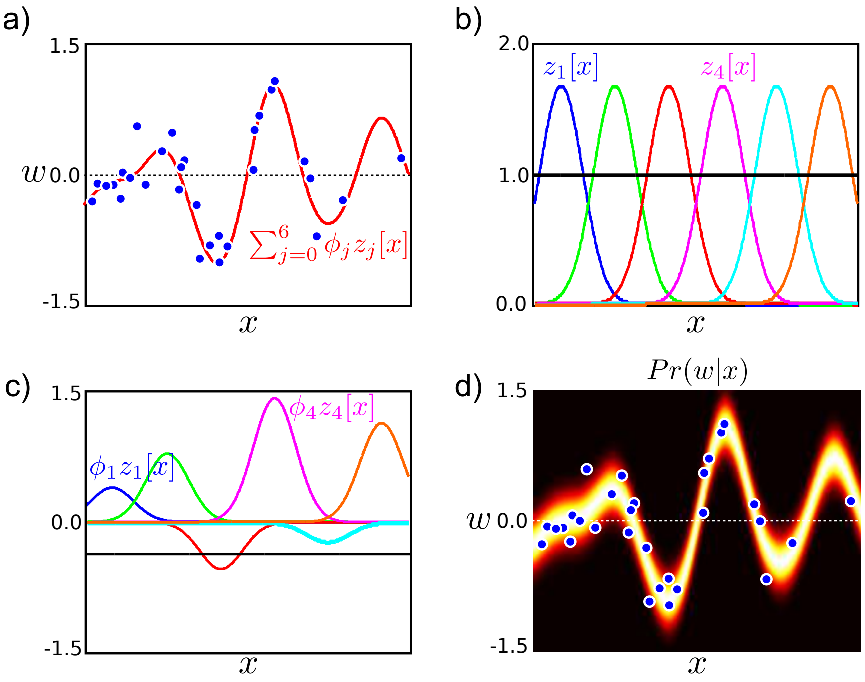

Figure 8.6 Non-linear regression using radial basis functions. a) The relationship between the data x and world w is clearly not linear. b) We compute a new seven dimensional vector $\mathbf{z}$ by evaluating the original observation x against each of six radial basis functions (Gaussians) and a constant function (black line). c) The mean of the predictive distribution (red line in (a)) can be formed by taking a linear sum $\boldsymbol{\phi}^T\mathbf{z}$ of these seven functions where the weights are as shown. The weights are estimated by maximum likelihood estimation of the linear regression model using the nonlinearly transformed data $\mathbf{z}$ instead of the original data $\mathbf{x}$. d) The final distribution $Pr(w|x)$ has a mean that is a sum of these functions and constant variance σ^2.

In figure 8.7 we compute a different nonlinear transformation and regress against the same data. This time, the transformation is based on arc tangent functions so that

$$\mathbf{z}_i = \begin{bmatrix} \arctan[\lambda x_i - \alpha_1] \\ \arctan[\lambda x_i - \alpha_2] \\ \arctan[\lambda x_i - \alpha_3] \\ \arctan[\lambda x_i - \alpha_4] \\ \arctan[\lambda x_i - \alpha_5] \\ \arctan[\lambda x_i - \alpha_6] \\ \arctan[\lambda x_i - \alpha_7] \end{bmatrix}. \tag{8.23}$$

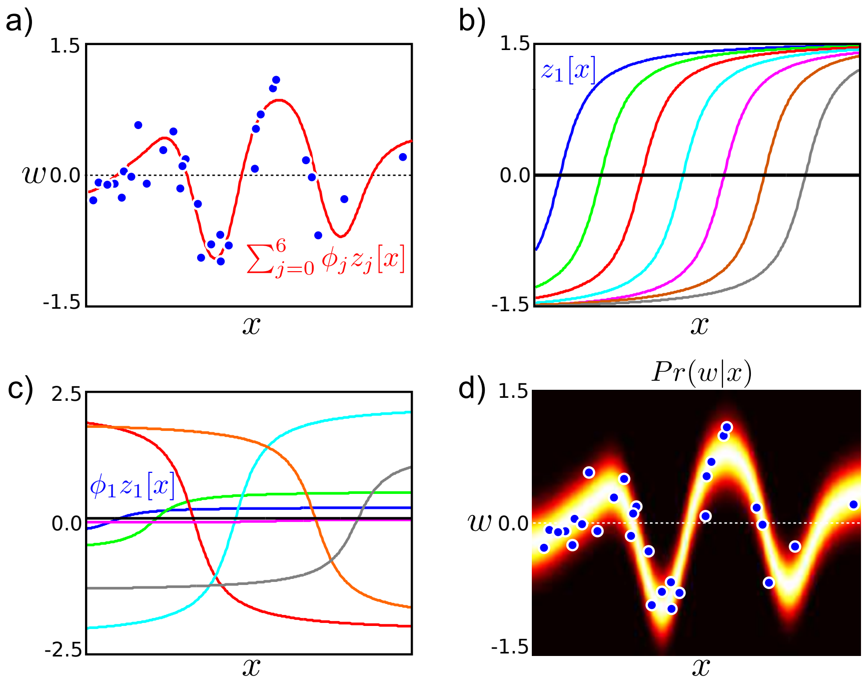

Figure 8.7 Non-linear regression using arc tangent functions. a) The relationship between the data x and world w is not linear. b) We compute a new seven dimensional vector $\mathbf{z}$ by evaluating the original observation x against each of seven arc tangent functions. c) The mean of the predictive distribution (red line in (a)) can be formed by taking a linear sum of these seven functions weighted as shown. The optimal weights were established using the maximum likelihood approach. d) The final distribution $Pr(w|x)$ has a mean that is a sum of these weighted functions and constant variance.

Here, the parameter λ controls the speed with which the function changes, and the parameters $\{\alpha_m\}_{m=1}^{7}$ determine the horizontal offsets of the arc tangent functions.

In this case, it is harder to understand the role of each weighted arc tangent function in the final regression, but nonetheless they collectively approximate the function well.

8.3.2 Bayesian nonlinear regression

In the Bayesian solution, the weights $\boldsymbol{\phi}$ of the nonlinear basis functions are treated as uncertain: in learning we compute the posterior distribution over these weights. For a new observation $\mathbf{x}^*$, we compute the transformed vector $\mathbf{z}^*$ and compute an infinite weighted sum over the predictions due to the possible parameter values (figure 8.8). The new expression for the predictive distribution is

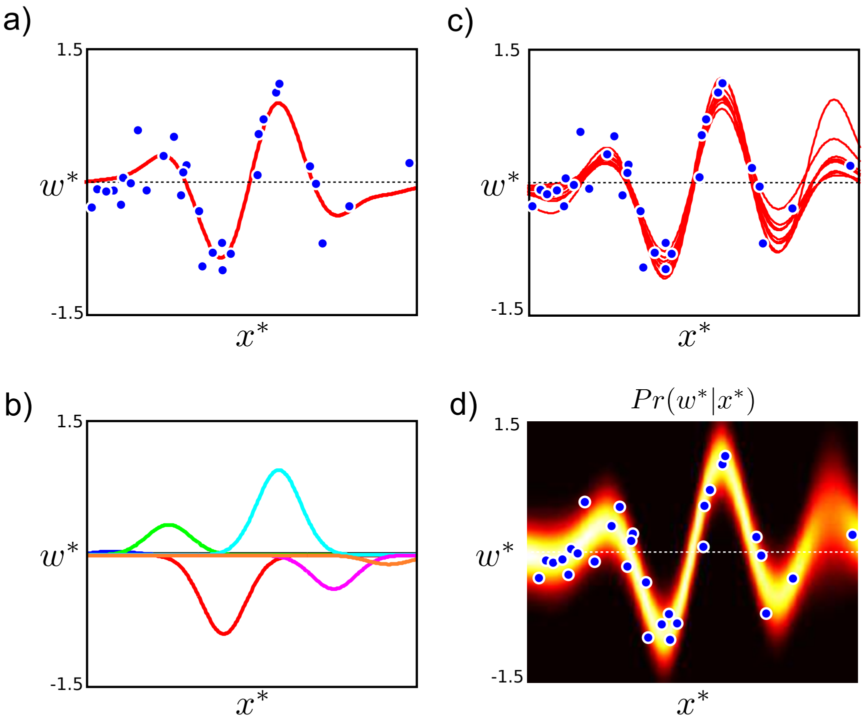

Figure 8.8 Bayesian nonlinear regression using radial basis functions. a) The relationship between the data and measurements is nonlinear. b) As in figure 8.6, the mean of the predictive distribution is constructed as a weighted linear sum of radial basis functions. However in the Bayesian approach we compute the posterior distribution over the weights $\boldsymbol{\phi}$ of these basis functions. c) Different draws from this distribution of weight parameters result in different predictions. d) The final predictive distribution is formed from an infinite weighted average of these predictions where the weight is given by the posterior probability. The variance of the predictive distribution depends on both the mutual agreement of these predictions and the noise σ^2. The uncertainty is greatest in the region on the right where there is little data and so the individual predictions vary widely.

$$\begin{aligned}Pr(w^*|\mathbf{z}^*,\mathbf{X},\mathbf{w}) &= \\ &\text{Norm}_w\left[\frac{\sigma_p^2}{\sigma^2}\mathbf{z}^{*T}\mathbf{Z}\mathbf{w}-\frac{\sigma_p^2}{\sigma^2}\mathbf{z}^{*T}\mathbf{Z}\left(\mathbf{Z}^T\mathbf{Z}+\frac{\sigma^2}{\sigma_p^2}\mathbf{I}\right)^{-1}\mathbf{Z}^T\mathbf{Z}\mathbf{w},\right.\\ &\left.\sigma_p^2\mathbf{z}^{*T}\mathbf{z}^*-\sigma_p^2\mathbf{z}^{*T}\mathbf{Z}\left(\mathbf{Z}^T\mathbf{Z}+\frac{\sigma^2}{\sigma_p^2}\mathbf{I}\right)^{-1}\mathbf{Z}^T\mathbf{z}^*+\sigma^2\right],\end{aligned} \tag{8.24}$$

where we have simply substituted the transformed vectors $\mathbf{z}$ for the original data

$\mathbf{x}$ in equation 8.14. The prediction variance depends on both the uncertainty in $\boldsymbol{\phi}$ and the additive variance σ^2. The Bayesian solution is less confident than the maximum likelihood solution (compare figures 8.8d and 8.7d), especially in regions where the data are sparse.

To compute the additive variance σ^2 we again optimize the marginal likelihood. The expression for this can be found by substituting $\mathbf{Z}$ for $\mathbf{X}$ in equation 8.15.

8.4 Kernels and the kernel trick

The Bayesian approach to nonlinear regression described in the previous section is rarely used directly in practice: the final expression for the predictive distribution (equation 8.24) relies on computing inner products $\mathbf{z}_i^T\mathbf{z}_j$. However, when the transformed space is high-dimensional it may be costly to compute the vectors $\mathbf{z}_i = \mathbf{f}[\mathbf{x}_i]$ and $\mathbf{z}_j = \mathbf{f}[\mathbf{x}_j]$ explicitly and then compute the inner product $\mathbf{z}_i^T\mathbf{z}_j$.

An alternative approach is to use *kernel substitution* in which we directly define a single *kernel function* $\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j]$ as a replacement for the operation $\mathbf{f}[\mathbf{x}_i]^T\mathbf{f}[\mathbf{x}_j]$. For many transformations $\mathbf{f}[\bullet]$ it is more efficient to evaluate the kernel function directly than to transform the variables separately and then compute the dot product.

Taking this idea one step further, it is possible to choose a kernel function $\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j]$ with no knowledge of what transformation $\mathbf{f}[\bullet]$ that it corresponds to. When we use kernel functions, we no longer explicitly compute the transformed vector $\mathbf{z}$. One advantage of this is we can define kernel functions that correspond to projecting the data into very high dimensional or even infinite spaces. This is sometimes called the *kernel trick*.

Clearly, the kernel function must be carefully chosen so that it does in fact correspond to computing some function $\mathbf{z} = \mathbf{f}[\mathbf{x}]$ for each data vector and taking the inner product of the resulting values: for example, since $\mathbf{z}_i^T\mathbf{z}_j = \mathbf{z}_j^T\mathbf{z}_i$ the kernel function must treat its arguments symmetrically so that $\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j] = \mathrm{k}[\mathbf{x}_j, \mathbf{x}_i]$.

More precisely, *Mercer's theorem* states that a kernel function is valid when the kernel's arguments are in a measurable space, and the kernel is positive semi-definite so that

$$\sum_{ij} \mathrm{k}[\mathbf{x}_i, \mathbf{x}_j] a_i a_j \geq 0 \tag{8.25}$$

for any finite subset $\{\mathbf{x}_n\}_{n=1}^N$ of vectors in the space and any real numbers $\{a_n\}_{n=1}^N$. Examples of valid kernels include:

- linear
$$\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j] = \mathbf{x}_i^T\mathbf{x}_j, \tag{8.26}$$

- degree p polynomial
$$\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j] = (\mathbf{x}_i^T\mathbf{x}_j + 1)^p, \tag{8.27}$$

- radial basis function (RBF) or Gaussian

$$\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j] = \exp\left[-0.5\left(\frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}{\lambda^2}\right)\right]. \tag{8.28}$$

The last of these is particularly interesting. It can be shown that this kernel function corresponds to computing *infinite* length vectors $\mathbf{z}$ and taking their dot product. The entries of $\mathbf{z}$ correspond to evaluating a radial basis function (figure 8.6b) at every possible point in the space of $\mathbf{x}$.

It is also possible to create new kernels by combining two or more existing kernels. For example, sums and products of valid kernels are guaranteed to be positive semi-definite and so are also valid kernels.

8.5 Gaussian process regression

Algorithm 8.3

We now replace the inner products $\mathbf{z}_i^T\mathbf{z}_j$ in the nonlinear regression algorithm (equation 8.24) with kernel functions. The resulting model is termed *Gaussian process regression.* The predictive distribution for a new datum $\mathbf{x}^*$ is

$$\begin{aligned}
&Pr(w^*|\mathbf{x}^*, \mathbf{X}, \mathbf{w}) = \qquad\qquad (8.29)\\
&\mathrm{Norm}_{w^*}\Bigg[\frac{\sigma_p^2}{\sigma^2}\mathbf{K}[\mathbf{x}^*, \mathbf{X}]\mathbf{w} - \frac{\sigma_p^2}{\sigma^2}\mathbf{K}[\mathbf{x}^*, \mathbf{X}]\left(\mathbf{K}[\mathbf{X}, \mathbf{X}] + \frac{\sigma^2}{\sigma_p^2}\mathbf{I}\right)^{-1}\mathbf{K}[\mathbf{X}, \mathbf{X}]\mathbf{w},\\
&\qquad \sigma_p^2\mathbf{K}[\mathbf{x}^*, \mathbf{x}^*] - \sigma_p^2\mathbf{K}[\mathbf{x}^*, \mathbf{X}]\left(\mathbf{K}[\mathbf{X}, \mathbf{X}] + \frac{\sigma^2}{\sigma_p^2}\mathbf{I}\right)^{-1}\mathbf{K}[\mathbf{X}, \mathbf{x}^*] + \sigma^2\Bigg].
\end{aligned}$$

where the notation $\mathbf{K}[\mathbf{X}, \mathbf{X}]$ represents a matrix of dot products where element (i, j) is given by $\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j]$.

Note that kernel functions may also contain parameters. For example, the RBF kernel (equation 8.28) takes the parameter λ, which determines the width of the underlying RBF functions and hence the smoothness of the function (figure 8.9). Kernel parameters such as λ can be learned by maximizing the marginal likelihood:

$$\begin{aligned}
\hat{\lambda} &= \underset{\lambda}{\mathrm{argmax}}\left[Pr(\mathbf{w}|\mathbf{X}, \sigma^2)\right]\\
&= \underset{\lambda}{\mathrm{argmax}}\left[\int Pr(\mathbf{w}|\mathbf{X}, \boldsymbol{\phi}, \sigma^2)Pr(\boldsymbol{\phi})d\boldsymbol{\phi}\right]\\
&= \underset{\lambda}{\mathrm{argmax}}\left[\mathrm{Norm}_{\mathbf{w}}[\mathbf{0}, \sigma_p^2\mathbf{K}[\mathbf{X}, \mathbf{X}] + \sigma^2\mathbf{I}]\right]. \qquad (8.30)
\end{aligned}$$

This typically requires a nonlinear optimization procedure.

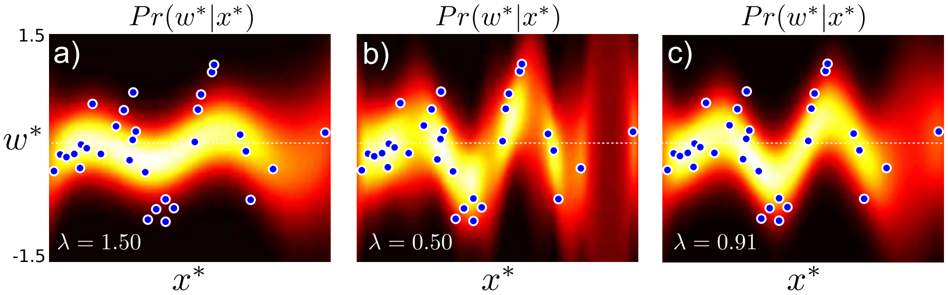

Figure 8.9 Gaussian process regression using an RBF kernel. a) When the length scale parameter λ is large, the function is too smooth. b) For small values of the length parameter the model does not successfully interpolate between the examples. c) The regression using the maximum likelihood length scale parameter is neither too smooth nor disjointed.

8.6 Sparse linear regression

Algorithm 8.4

We now turn our attention to the third potential disadvantage of linear regression. It is often the case that only a small subset of the dimensions of $\mathbf{x}$ are useful for predicting w. However, without modification, the linear regression algorithm will assign non-zero values to the gradient $\boldsymbol{\phi}$ in these directions. The goal of *sparse* linear regression is to adapt the algorithm to find a gradient vector $\boldsymbol{\phi}$ where most of the entries are zero. The resulting classifier will be faster, since we no longer even have to make all of the measurements. Furthermore, simpler models are preferable to complex ones; they capture the main trends in the data without over-fitting to peculiarities of the training set and generalize better to new test examples.

To encourage sparse solutions, we impose a penalty for every non-zero weighted data dimension. We replace the normal prior over the gradient parameters $\boldsymbol{\phi} = [\phi_1, \ \phi_2, \ldots, \phi_D]^T$ with a product of one-dimensional t-distributions so that

$$\begin{aligned} Pr(\boldsymbol{\phi}) &= \prod_{d=1}^{D} \text{Stud}_{\phi_d}[0,1,\nu] \\ &= \prod_{d=1}^{D} \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{\phi_d^2}{\nu}\right)^{-(\nu+1)/2}. \end{aligned} \tag{8.31}$$

The product of univariate t-distributions has ridges of high probability along the coordinate axes, which encourages sparseness (see figure 8.10). We expect the final solution to be a trade-off between fitting the training data accurately and the sparseness of $\boldsymbol{\phi}$ (and hence the number of training data dimensions that contribute to the solution).

Adopting the Bayesian approach, our aim is to compute the posterior distribution $Pr(\boldsymbol{\phi}|\mathbf{X}, \mathbf{w}, \sigma^2)$ over the possible values of the gradient variable $\boldsymbol{\phi}$ using this new prior so that

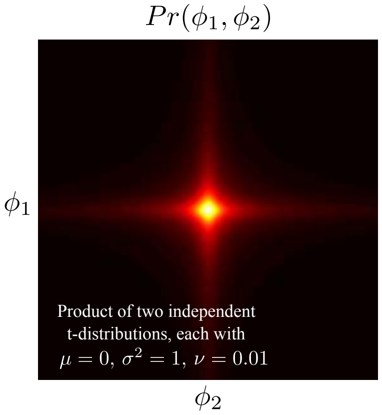

Figure 8.10 A product of two 1D t-distributions where each has small degrees of freedom ν. This 2D distribution favors sparseness (where one or both variables are close to zero). In higher dimensions, the product of t-distributions encourages solutions where *most* variables are set to zero. Note that the product of 1D distributions is *not* the same as a multivariate t-distribution with a spherical covariance matrix, which looks like a multivariate normal distribution but with longer tails.

$$Pr(\boldsymbol{\phi}|\mathbf{X},\mathbf{w},\sigma^2) = \frac{Pr(\mathbf{w}|\mathbf{X},\boldsymbol{\phi},\sigma^2)Pr(\boldsymbol{\phi})}{Pr(\mathbf{w}|\mathbf{X},\sigma^2)}. \tag{8.32}$$

Unfortunately, there is no simple closed form expression for the posterior on the left hand side. The prior is no longer normal and the conjugacy relationship is lost.

To make progress, we re-express each t-distribution as an infinite weighted sum of normal distributions where a hidden variable h_d determines the variance (section 7.5), so that

$$\begin{aligned} Pr(\boldsymbol{\phi}) &= \prod_{d=1}^{D}\int \text{Norm}_{\phi_d}[0,1/h_d]\text{Gam}_{h_d}[\nu/2,\nu/2]\ dh_d \\ &= \int \text{Norm}_{\boldsymbol{\phi}}[0,\mathbf{H}^{-1}]\prod_{d=1}^{D}\text{Gam}_{h_d}[\nu/2,\nu/2]\ d\mathbf{H}, \end{aligned} \tag{8.33}$$

where the matrix $\mathbf{H}$ contains the hidden variables $\{h_d\}_{d=1}^{D}$ on its diagonal and zeros elsewhere. We now write out the expression for the marginal likelihood (likelihood after integrating over the gradient parameters $\boldsymbol{\phi}$) as

$$\begin{aligned} Pr(\mathbf{w}|\mathbf{X},\sigma^2) &\propto \int Pr(\mathbf{w},\boldsymbol{\phi}|\mathbf{X},\sigma^2)\ d\boldsymbol{\phi} \\ &= \int Pr(\mathbf{w}|\mathbf{X},\boldsymbol{\phi},\sigma^2)Pr(\boldsymbol{\phi})\ d\boldsymbol{\phi} \\ &= \int \text{Norm}_{\mathbf{w}}[\mathbf{X}^T\boldsymbol{\phi},\sigma^2\mathbf{I}]\int \text{Norm}_{\boldsymbol{\phi}}[0,\mathbf{H}^{-1}]\prod_{d=1}^{D}\text{Gam}_{h_d}[\nu/2,\nu/2]\ d\mathbf{H}d\boldsymbol{\phi} \\ &= \iint \text{Norm}_{\mathbf{w}}[\mathbf{X}^T\boldsymbol{\phi},\sigma^2\mathbf{I}]\text{Norm}_{\boldsymbol{\phi}}[0,\mathbf{H}^{-1}]\prod_{d=1}^{D}\text{Gam}_{h_d}[\nu/2,\nu/2]\ d\mathbf{H}d\boldsymbol{\phi} \\ &= \int \text{Norm}_{\mathbf{w}}[\mathbf{0},\mathbf{X}^T\mathbf{H}^{-1}\mathbf{X}+\sigma^2\mathbf{I}]\prod_{d=1}^{D}\text{Gam}_{h_d}[\nu/2,\nu/2]\ d\mathbf{H}. \end{aligned} \tag{8.34}$$

where we have computed the integral over $\boldsymbol{\phi}$ using the same method as in equation 8.12.

Unfortunately, we still cannot compute the remaining integral in closed form, so we instead take the approach of maximizing over hidden variables to give an approximate expression for the marginal likelihood

$$Pr(\mathbf{w}|\mathbf{X},\sigma^2) \approx \max_{\mathbf{H}} \left[\text{Norm}_{\mathbf{w}}[\mathbf{0}, \mathbf{X}^T\mathbf{H}^{-1}\mathbf{X} + \sigma^2\mathbf{I}] \prod_{d=1}^{D} \text{Gam}_{h_d}[\nu/2, \nu/2] \right]. \tag{8.35}$$

As long as the true distribution over the hidden variables is concentrated tightly around the mode, this will be a reasonable approximation. When h_d takes a large value, the prior has a small variance $(1/h_d)$, and the associated coefficient ϕ_d will be forced to be close to zero: in effect, this means that the d^{th} dimension of $\mathbf{x}$ does not contribute to the solution and can be dropped from the equations.

The general approach to fitting the model is now clear. There are two unknown quantities – the variance σ^2 and the hidden variables $\mathbf{h}$ and we alternately update each to maximize the log marginal likelihood.[1]

- To update the hidden variables, we take the derivative of the log of this expression with respect to $\mathbf{H}$, equate the result to zero, and re-arrange to get the iteration

$$h_d^{new} = \frac{1 - h_d\Sigma_{dd} + \nu}{\mu_d^2 + \nu}, \tag{8.36}$$

  where μ_d is the d^{th} element of the mean $\boldsymbol{\mu}$ of the posterior distribution over the weights $\boldsymbol{\phi}$ and Σ_{dd} is the d^{th} element of the diagonal of the covariance $\boldsymbol{\Sigma}$ of the posterior distribution over the weights (equation 8.10) so that

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{w} \\ \boldsymbol{\Sigma} &= \mathbf{A}^{-1}, \end{aligned} \tag{8.37}$$

  and $\mathbf{A}$ is defined as

$$\mathbf{A} = \frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \mathbf{H}. \tag{8.38}$$

- To update the variance, we take the derivative of the log of this expression with respect to σ^2, equate the result to zero, and simplify to get

$$(\sigma^2)^{new} = \frac{1}{D - \sum_d (1 - h_d\Sigma_{dd})} (\mathbf{w} - \mathbf{X}\boldsymbol{\mu})^T (\mathbf{w} - \mathbf{X}\boldsymbol{\mu}). \tag{8.39}$$

[1] More details about how these (non-obvious) update equations were generated can be found in section 3.5 of Bishop (2006) and Tipping (2001).

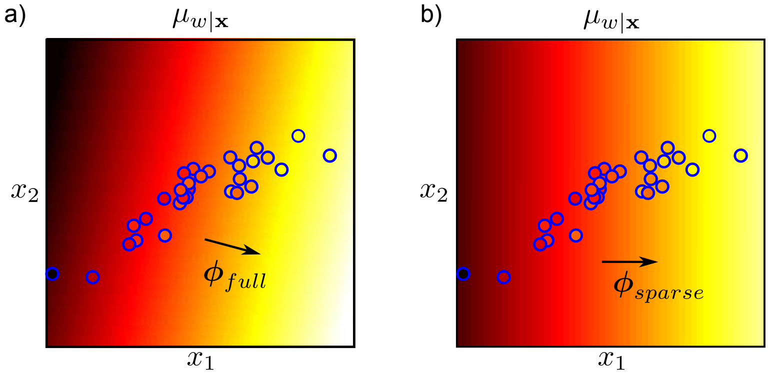

Figure 8.11 Sparse linear regression. a) Bayesian linear regression from two-dimensional data. The background color represents the mean $\mu_{w|\mathbf{x}}$ of the Gaussian prediction $Pr(w|\mathbf{x})$ for w. The variance of $Pr(w|\mathbf{x})$ is not shown. The color of the data points indicates the training value w, so for a perfect regression fit this should match exactly the surrounding color. Here the elements of $\boldsymbol{\phi}$ take arbitrary values and so the gradient of the function points in an arbitrary direction. b) Sparse linear regression. Here, the elements of $\boldsymbol{\phi}$ are encouraged to be zero where they are not necessary to explain the data. The algorithm has found a good fit where the second element of $\boldsymbol{\phi}$ is zero and so there is no dependence on the vertical axis.

In between each of these updates, the posterior mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ should be recalculated.

In practice, we choose a very small value for the degrees of freedom ($\nu < 10^{-3}$) to encourage sparseness. We may also restrict the maximum possible values of the hidden variables h_i to ensure numerical stability.

At the end of the training, all dimensions of $\boldsymbol{\phi}$ where the hidden variable h_d is large (say > 1000) are discarded. Figure 8.11 shows an example fit to some two-dimensional data. The sparse solution depends only on one of the two possible directions and so is twice as efficient.

In principle, a nonlinear version of this algorithm can be generated by transforming the input data $\mathbf{x}$ to create the vector $\mathbf{z} = \mathbf{f}[\mathbf{x}]$. However, if the transformed data $\mathbf{z}$ is very high-dimensional, we will need correspondingly more hidden variables h_d to cope with these dimensions. Obviously, this idea will not transfer to kernel functions where the dimensionality of the transformed data could be infinite.

To resolve this problem, we will develop the *relevance vector machine*. This model also imposes sparsity, but it does so in a way that makes the final prediction depend only on a sparse subset of the training data, rather than a sparse subset of the observed dimensions. Before we can investigate this model, we must develop a version of linear regression where there is one parameter per data example rather than one per observed dimension. This model is known as *dual linear regression*.

8.7 Dual linear regression

Problem 8.10

In the standard linear regression model the parameter vector $\boldsymbol{\phi}$ contains D entries corresponding to each of the D dimensions of the (possibly transformed) input data. In the dual formulation, we re-parameterize the model in terms of a vector $\boldsymbol{\psi}$ which has I entries where I is the number of training examples. This is more efficient in situations where we are training a model where the input data are high dimensional, but the number of examples is small ($I < D$), and leads to other interesting models, such as relevance vector regression.

8.7.1 Dual model

In the dual model, we retain the original linear dependence of the prediction w on the input data $\mathbf{x}$ so that

$$Pr(w_i|\mathbf{x}_i) = \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\phi}^T\mathbf{x}_i, \sigma^2]. \tag{8.40}$$

However, we now represent the slope parameters $\boldsymbol{\phi}$ as a weighted sum of the observed data points so that

$$\boldsymbol{\phi} = \mathbf{X}\boldsymbol{\psi}, \tag{8.41}$$

where $\boldsymbol{\psi}$ is a $I \times 1$ vector representing the weights (figure 8.12). We term this the *dual parameterization*. Notice that if there are fewer data examples than data dimensions, then there will be fewer unknowns here than in the standard linear regression model and hence learning and inference will be more efficient. Note that the term ‘dual’ is heavily overloaded in computer science, and the reader should be careful not to confuse this use with its other meanings.

If the data dimensionality D is less than the number of examples I then we can find parameters $\boldsymbol{\psi}$ to represent any gradient vector $\boldsymbol{\phi}$. However, if $D > I$ (often true in vision where measurements can be high dimensional), then the vector $\mathbf{X}\boldsymbol{\psi}$ can only span a subspace of the possible gradient vectors. However, this is not a problem: if there was no variation in the data $\mathbf{X}$ in a given direction in space, then the gradient along that axis should be zero anyway since we have no information about how the world state w varies in this direction.

Making the substitution from equation 8.41, the regression model becomes

$$Pr(w_i|\mathbf{x}_i, \boldsymbol{\theta}) = \text{Norm}_{\mathbf{x}_i}[\boldsymbol{\psi}^T\mathbf{X}^T\mathbf{x}_i, \sigma^2], \tag{8.42}$$

or writing all of the data likelihoods in one term

$$Pr(\mathbf{w}|\mathbf{X}, \boldsymbol{\theta}) = \text{Norm}_{\mathbf{w}}\left[\mathbf{X}^T\mathbf{X}\boldsymbol{\psi}, \sigma^2\mathbf{I}\right], \tag{8.43}$$

where the parameters of the model are $\boldsymbol{\theta} = \{\boldsymbol{\psi}, \sigma^2\}$. We now consider how to learn this model using both the maximum likelihood and Bayesian approaches.

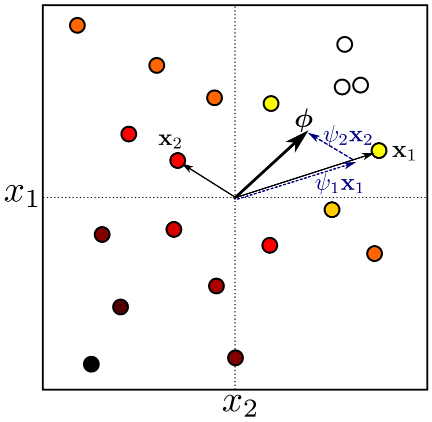

Figure 8.12 Dual variables. Two dimensional training data $\{\mathbf{x}_i\}_{i=1}^I$ and associated world state $\{w_i\}_{i=1}^I$ (indicated by marker color). The linear regression parameter $\boldsymbol{\phi}$ determines the direction in this 2D space in which w changes most quickly. We can alternately represent the gradient direction as a weighted sum of data examples. Here we show the case $\boldsymbol{\phi} = \psi_1\mathbf{x}_1 + \psi_2\mathbf{x}_2$. In practical problems the data dimensionality D is greater than the number of examples I so we take a weighted sum $\boldsymbol{\phi} = \mathbf{X}\boldsymbol{\psi}$ of all of the data points. This is the dual parameterization.

Maximum likelihood solution

Algorithm 8.5

We apply the maximum likelihood method to estimate the parameters $\boldsymbol{\psi}$ in the dual formulation. To this end we maximize the logarithm of the likelihood (equation 8.43) with respect to $\boldsymbol{\psi}$ and σ^2 so that

$$\hat{\boldsymbol{\psi}}, \hat{\sigma}^2 = \underset{\boldsymbol{\psi},\sigma^2}{\operatorname{argmax}} \left[-\frac{I\log[2\pi]}{2} - \frac{I\log[\sigma]}{2} - \frac{(\mathbf{w}-\mathbf{X}^T\mathbf{X}\boldsymbol{\psi})^T(\mathbf{w}-\mathbf{X}^T\mathbf{X}\boldsymbol{\psi})}{2\sigma^2}\right]. \quad (8.44)$$

Problem 8.11

To maximize this expression, we take derivatives with respect to $\boldsymbol{\psi}$ and σ^2, equate the resulting expressions to zero, and solve to find

$$\begin{aligned} \hat{\boldsymbol{\psi}} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{w} \\ \hat{\sigma}^2 &= \frac{(\mathbf{w}-\mathbf{X}^T\mathbf{X}\boldsymbol{\psi})^T(\mathbf{w}-\mathbf{X}^T\mathbf{X}\boldsymbol{\psi})}{I}. \end{aligned} \quad (8.45)$$

This solution is actually the same as for the original linear regression model (equations 8.6). For example, if we substitute in the definition $\boldsymbol{\phi} = \mathbf{X}\boldsymbol{\psi}$,

$$\begin{aligned} \hat{\boldsymbol{\phi}} = \mathbf{X}\hat{\boldsymbol{\psi}} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{w} \\ &= (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{w} \\ &= (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{w}, \end{aligned} \quad (8.46)$$

which was the original maximum likelihood solution for $\boldsymbol{\phi}$.

Bayesian solution

We now explore the Bayesian approach to the dual regression model. As before, we treat the dual parameters $\boldsymbol{\psi}$ as uncertain, assuming that the noise σ^2 is known. Once again, we will estimate this separately using maximum likelihood.

The goal of the Bayesian approach is to compute the posterior distribution $Pr(\boldsymbol{\psi}|\mathbf{X},\mathbf{w})$ over possible values of the parameters $\boldsymbol{\psi}$ given the training data pairs $\{\mathbf{x}_i, w_i\}_{i=1}^{I}$. We start by defining a prior $Pr(\boldsymbol{\psi})$ over the parameters. Since we have no particular prior knowledge, we choose a normal distribution with a large spherical covariance,

$$Pr(\boldsymbol{\psi}) = \text{Norm}_{\boldsymbol{\psi}}[\mathbf{0}, \sigma_p^2\mathbf{I}]. \tag{8.47}$$

We use Bayes' rule to compute the posterior distribution over the parameters

$$Pr(\boldsymbol{\psi}|\mathbf{X},\mathbf{w},\sigma^2) = \frac{Pr(\mathbf{X}|\mathbf{w},\boldsymbol{\psi},\sigma^2)Pr(\boldsymbol{\psi})}{Pr(\mathbf{X}|\mathbf{w},\sigma^2)}, \tag{8.48}$$

which can be shown to yield the closed form expression

$$Pr(\boldsymbol{\psi}|\mathbf{X},\mathbf{w},\sigma^2) \quad = \quad \text{Norm}_{\boldsymbol{\psi}}\left[\frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w}, \mathbf{A}^{-1}\right], \tag{8.49}$$

where

$$\mathbf{A} \quad = \quad \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{X} + \frac{1}{\sigma_p^2}\mathbf{I}. \tag{8.50}$$

To compute the predictive distribution $Pr(\mathbf{w}^*|\mathbf{x}^*)$, we take an infinite weighted sum over the predictions of the model associated with each possible value of the parameters $\boldsymbol{\psi}$,

$$\begin{aligned} Pr(w^*|\mathbf{x}^*,\mathbf{X},\mathbf{w}) &= \int Pr(w^*|\mathbf{x}^*,\boldsymbol{\psi})Pr(\boldsymbol{\psi}|\mathbf{X},\mathbf{w})\, d\boldsymbol{\psi} \qquad (8.51)\\ &= \text{Norm}_{w^*}\left[\frac{1}{\sigma^2}\mathbf{x}^{*T}\mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w}^*, \mathbf{x}^{*T}\mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T\mathbf{x}^* + \sigma^2\right]. \end{aligned}$$

To generalize the model to the nonlinear case, we replace the training data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I]$ with the transformed data $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_I]$ and the test data $\mathbf{x}^*$ with the transformed test data $\mathbf{z}^*$. Since the resulting expression depends only on inner products of the form $\mathbf{Z}^T\mathbf{Z}$ and $\mathbf{Z}^T\mathbf{z}^*$, it is directly amenable to kernelization. Algorithm 8.6

As for the original regression model, the variance parameter σ^2 can be estimated by maximizing the log of the marginal likelihood which is given by

$$Pr(\mathbf{w}|\mathbf{X},\sigma^2) = \text{Norm}_{\mathbf{w}}[\mathbf{0}, \sigma_p^2\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{I}]. \tag{8.52}$$

8.8 Relevance vector regression

Having developed the dual approach to linear regression, we are now in a position to develop a model that depends only sparsely on the training data. To this end, Algorithm 8.7

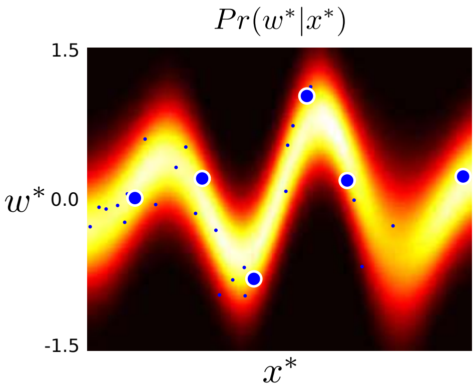

Figure 8.13 Relevance vector regression. A prior applying sparseness is applied to the dual parameters. This means that the final classifier only depends on a subset of the data points (indicated by the six larger points). The resulting regression function is considerably faster to evaluate and tends to be simpler: this means it is less likely to overfit to random statistical fluctuations in the training data and generalizes better to new data.

we impose a penalty for every non-zero weighted training example. We achieve this by replacing the normal prior over the dual parameters $\boldsymbol{\psi}$ with a product of one dimensional t-distributions so that

$$Pr(\boldsymbol{\psi}) \quad = \quad \prod_{i=1}^{I} \text{Stud}_{\boldsymbol{\psi}_i}\left[0, 1, \nu\right]. \tag{8.53}$$

This model is known as *relevance vector regression.*

This situation is exactly analogous to the sparse linear regression model (section 8.6) except that now we are working with dual variables. As for the sparse model it is not possible to marginalize over the variables $\boldsymbol{\psi}$ with the t-distributed prior. Our approach will again be to approximate the t-distributions by maximizing with respect to their hidden variables rather than marginalizing over them (equation 8.35). By analogy with section 8.6, the marginal likelihood becomes:

$$Pr(\mathbf{w}|\mathbf{X},\sigma^2) \approx \max_{\mathbf{H}}\left[\text{Norm}_{\mathbf{w}}[\mathbf{0},\mathbf{X}^T\mathbf{X}\mathbf{H}^{-1}\mathbf{X}^T\mathbf{X}+\sigma^2\mathbf{I}]\prod_{d=1}^{D}\text{Gam}_{h_d}[\nu/2,\nu/2]\right]. \tag{8.54}$$

where the matrix $\mathbf{H}$ contains the hidden variables $\{h_i\}_{i=1}^{I}$ associated with the t-distribution on its diagonal and zeros elsewhere. Notice that this expression is similar to equation 8.52 except that instead of every data point having the same prior variance σ_p^2, they now have individual variances that are determined by the hidden variables that form the elements of the diagonal matrix $\mathbf{H}$.

In relevance vector regression, we alternately (i) optimize the marginal likelihood with respect to the hidden variables and (ii) optimize the marginal likelihood with respect to the variance parameter σ^2 using

$$h_i^{new} = \frac{1 - h_i\Sigma_{ii} + \nu}{\mu_i^2 + \nu}, \tag{8.55}$$

and

$$(\sigma^2)^{new} = \frac{1}{I - \sum_i (1 - h_i \Sigma_{ii})} \left(\mathbf{w} - \mathbf{X}^T\mathbf{X}\boldsymbol{\mu}\right)^T \left(\mathbf{w} - \mathbf{X}^T\mathbf{X}\boldsymbol{\mu}\right), \tag{8.56}$$

In between each step we update the mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ of the posterior distribution

$$\begin{aligned} \boldsymbol{\mu} &= \frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}^T\mathbf{X}\mathbf{w} \\ \boldsymbol{\Sigma} &= \mathbf{A}^{-1}, \end{aligned} \tag{8.57}$$

where $\mathbf{A}$ is defined as

$$\mathbf{A} = \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{X} + \mathbf{H}. \tag{8.58}$$

At the end of the training all data examples where the hidden variable h_i is large (say > 1000) are discarded as here the coefficients $\boldsymbol{\psi}_i$ will be very small and contribute almost nothing to the solution.

Since this algorithm depends only on inner products, a nonlinear version of this algorithm can be generated by replacing the inner products with a kernel function $\mathrm{k}[\mathbf{x}_i, \mathbf{x}_j]$. If the kernel itself contains parameters, these may be also be manipulated to improve the log marginal variance during the fitting procedure. Figure 8.13 shows an example fit using the RBF kernel. The final solution now only depends on six data points but nonetheless still captures the important aspects of the data.

8.9 Regression to multivariate data

Throughout this chapter we have discussed predicting a scalar value w_i from multivariate data $\mathbf{x}_i$. In real-world situations such as the pose regression problem, the world states $\mathbf{w}_i$ are multivariate. It is trivial to extend the models in this chapter: we simply construct a separate regressor for each dimension. The exception to this rule is the relevance vector machine: here we might want to ensure that the sparse structure is common for each of these models, so the efficiency gains are retained. To this end, we modify the model so that a single set of hidden variables is shared across the model for each world state dimension.

8.10 Applications

Regression methods are used less frequently in vision than classification, but nonetheless there are many useful applications. The majority of these involve estimating the position or pose of objects, since the unknowns in such problems are naturally treated as continuous.

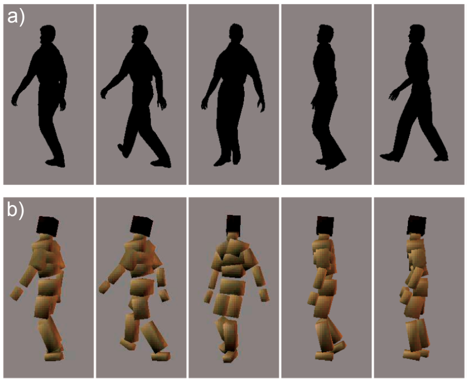

Figure 8.14 Body pose estimation results. a) Silhouettes of walking avatar. b) Estimated body pose based on silhouette using a relevance vector machine. The RVM used radial basis functions and constructed its final solution from just 156 of 2636 (6%) of the training examples. It produced a mean test error of 6.0^o averaged over the three joint angles for the 18 main body parts and the overall compass direction of the model. Adapted from Agarwal & Triggs (2006).

8.10.1 Human body pose estimation

Agarwal & Triggs (2006) developed a system based on the relevance vector machine to predict body pose $\mathbf{w}$ from silhouette data $\mathbf{x}$. To encode the silhouette, they computed a 60-dimensional shape context feature (section 13.3.5) at each of 400-500 points on the boundary of the object. To reduce the data dimensionality, they computed the similarity of each shape context feature to each of 100 different prototypes. Finally, they formed a 100-dimensional histogram containing the aggregated 100-dimensional similarities for all of the boundary points. This histogram was used as the data vector $\mathbf{x}$. The body pose was encoded by the 3 joint angles of each of the 18 major body joints and the overall azimuth (compass heading) of the body. The resulting 55-dimensional vector was used as the world state $\mathbf{w}$.

A relevance vector machine was trained using 2636 data vectors $\mathbf{x}_i$ extracted from silhouettes that were rendered using the commercial program POSER from known motion capture data $\mathbf{w}_i$. Using a radial basis function kernel, the relevance vector machine based its solution on just 6% of these training examples. The body pose angles of test data could be predicted to within an average of 6^o error (figure 8.14). They also demonstrated that the system worked reasonably well on silhouettes from real images (figure 8.1).

Silhouette information is by its nature ambiguous: it is very hard to tell which leg is in front of the other based on a single silhouette. Agarwal & Triggs (2006) partially circumvented this system by tracking the body pose $\mathbf{w}_i$ through a video sequence. Essentially, the ambiguity at a given frame is resolved by encouraging the estimated pose in adjacent frames in the sequence to be similar: information from frames where the pose vector is well defined is propagated through the sequence to resolve ambiguities in other parts (see chapter 19).

However, the ambiguity of silhouette data is an argument for *not* using this type of classifier: the regression models in this chapter are designed to give a unimodal normal output. To effectively classify single frames of data, we should use a regression method that produces a multi-modal prediction that can effectively describe the ambiguity.

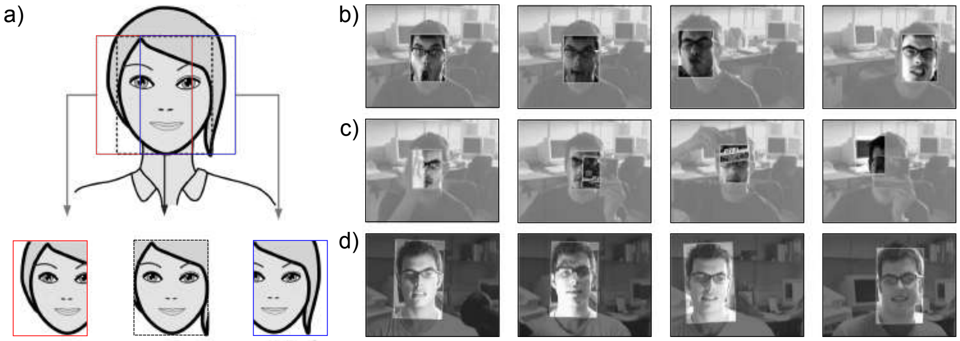

Figure 8.15 Tracking using displacement experts. The goal of the system is to predict a displacement vector indicating the motion of the object based on the pixel data at its last known position. a) The system is trained by perturbing the bounding box around the object to simulate the motion of the object. b) The system successfully tracks a face, even in the presence c) of partial occlusions. d) If the system is trained using gradient vectors rather than raw pixel values, it is also quite robust to changes in illumination. Adapted from Williams *et al.* (2005) ©2005 IEEE.

8.10.2 Displacement experts

Regression models are also used to form *displacement experts* in tracking applications. The goal is to take a region of the image $\mathbf{x}$ and return a set of numbers $\mathbf{w}$ that indicate the change in position of an object relative to the window. The world state $\mathbf{w}$ might simply contain the horizontal and vertical translation vectors, or might contain parameters of a more complex 2D transformation (chapter 15). For simplicity, we will describe the former situation.

Training data are extracted as follows. A bounding box around of the object of interest (car, face, etc.) is identified in a number of frames. For each of these frames, the bounding box is perturbed by a pre-determined set of translation vectors, to simulate the object moving in the opposite direction (figure 8.15a). In this way, we associate a translation vector $\mathbf{w}_i$ with each perturbation. The data from the perturbed bounding box are extracted, resized to a standard shape, and histogram equalized (section 13.1.2) to induce a degree of invariance to illumination changes. The resulting values are then concatenated to form the data vector $\mathbf{x}_i$.

Williams *et al.* (2005) describe a system of this kind in which the elements of $\mathbf{w}$ were learned by a set of independent relevance vector machines. They initialize the position of the object using a standard object detector (see chapter 9). In the subsequent frame, they compute a prediction for the displacement vector $\mathbf{w}$ using the relevance vector machines on the data $\mathbf{x}$ from the original position. This prediction is combined in a Kalman filter-like system (chapter 19) that imposes prior knowledge about the continuity of the motion to create a robust method for tracking known objects in scenes. Figure 8.15b-d show a series of tracking results from this system.

Discussion

The goal of this chapter was to introduce discriminative approaches to regression. These have niche applications in vision related to predicting the pose and position of objects. However, the main reason for studying these models is that the concepts involved (sparsity, dual variables, kernelization) are all important for discriminative classification methods. These are very widely used but are rather more complex and are discussed in the following chapter.

Notes

Regression methods: Rasmussen & Williams (2006) is a comprehensive resource on Gaussian processes. The relevance vector machine was first introduced by Tipping (2001). Several innovations within the vision community have extended these models. Williams *et al.* (2006) presented a semi-supervised method for Gaussian process regression in which the world state $\mathbf{w}$ is only known for a subset of examples. Ranganathan & Yang (2008) presented an efficient algorithm for online learning of Gaussian processes when the kernel matrix is sparse. Thayananthan *et al.* (2006) developed a multivariate version of the relevance vector machine.

Applications: Applications of regression in vision include head pose estimation (Williams *et al.* 2006; Ranganathan & Yang 2008; Rae & Ritter 1998), body tracking (Williams *et al.* 2006; Agarwal & Triggs 2006; Thayananthan *et al.* 2006), eye tracking (Williams *et al.* 2006), and tracking of other objects (Williams *et al.* 2005; Ranganathan & Yang 2008).

Multimodal posterior: One of the drawbacks of using the methods in this chapter is that they always produce a unimodal normally distributed posterior. For some problems (e.g., body pose estimation), the posterior probability over the world state may be genuinely multimodal – there is more than one interpretation of the data. One approach to this is to build many regressors that relate small parts of the world state to the data (Thayananthan *et al.* 2006). Alternatively, it is possible to use generative regression methods in which either the joint density is modeled directly (Navaratnam *et al.* 2007) or the likelihood and prior are modeled separately (Urtasun *et al.* 2006). In these methods, the posterior distribution over the world is multi-modal. However, the cost of this is that it is intractable to compute exactly, and so we must rely on optimization techniques to find its modes.

Problems

Problem 8.1 Consider a regression problem where the world state w is known to be positive. To cope with this we could construct a regression model in which the world state is modeled as a gamma distribution. We could constrain both parameters α, β of the gamma distribution to be the same so that $\alpha = \beta$ and make them a function of the data $\mathbf{x}$. Describe a maximum likelihood approach to fitting this model.

Problem 8.2 Consider a robust regression model based on the t-distribution rather than the normal distribution. Define this model precisely in mathematical terms and sketch out a maximum likelihood approach to fitting the parameters.

Problem 8.3 Prove that the maximum likelihood solution for the gradient in the linear regression model is

$$\hat{\boldsymbol{\phi}} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{w}.$$

Problem 8.4 For the Bayesian linear regression model (section 8.2), show that the posterior distribution over the parameters $\boldsymbol{\phi}$ is given by

$$Pr(\boldsymbol{\phi}|\mathbf{X}, \mathbf{w}) = \text{Norm}_{\boldsymbol{\phi}}\left[\frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}\mathbf{w}, \mathbf{A}^{-1}\right],$$

where

$$\mathbf{A} = \frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \frac{1}{\sigma_p^2}\mathbf{I}.$$

Problem 8.5 For the Bayesian linear regression model (section 8.2) show that the predictive distribution for a new data example $\mathbf{x}^*$ is given by

$$Pr(w^*|\mathbf{x}^*, \mathbf{X}, \mathbf{w}) = \text{Norm}_{w^*}\left[\frac{1}{\sigma^2}\mathbf{x}^{*T}\mathbf{A}^{-1}\mathbf{X}\mathbf{w}, \mathbf{x}^{*T}\mathbf{A}^{-1}\mathbf{x}^* + \sigma^2\right].$$

Problem 8.6 Use the matrix inversion lemma (appendix C.8.4) to show that

$$\mathbf{A}^{-1} = \left(\frac{1}{\sigma^2}\mathbf{X}\mathbf{X}^T + \frac{1}{\sigma_p^2}\mathbf{I}_D\right)^{-1} = \sigma_p^2\mathbf{I}_D - \sigma_p^2\mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \frac{\sigma^2}{\sigma_p^2}\mathbf{I}_I\right)^{-1}\mathbf{X}^T.$$

Problem 8.7 Compute the derivative of the marginal likelihood

$$Pr(\mathbf{w}|\mathbf{X}, \sigma^2) \quad = \quad \text{Norm}_{\mathbf{w}}[\mathbf{0}, \sigma_p^2\mathbf{X}^T\mathbf{X} + \sigma^2\mathbf{I}],$$

with respect to the variance parameter σ^2.

Problem 8.8

Compute a closed form expression for the approximated t-distribution used to impose sparseness.

$$q(h) = \max_h\left[\text{Norm}_{\boldsymbol{\phi}}[0, h^{-1}]\text{Gam}_h[\nu/2, \nu/2]\right].$$

Plot this function for $\nu = 2$. Plot the 2D function $[h_1, h_2] = q(h_1)q(h_2)$ for $\nu = 2$.

Problem 8.9 Describe maximum likelihood learning and inference algorithms for a non-linear regression model based on polynomials where

$$Pr(w|x) = \text{Norm}_w[\phi_0 + \phi_1 x + \phi_2 x^2 + \phi_3 x^3, \sigma^2].$$

Problem 8.10 I wish to learn a linear regression model in which I predict the world w from I examples of $D \times 1$ data $\mathbf{x}$ using the maximum likelihood method. If $I > D$ is it more efficient to use the dual parameterization or the original linear regression model?

Problem 8.11 Show that the maximum likelihood estimate for the parameters $\boldsymbol{\psi}$ in the dual linear regression model (section 8.7) is given by

$$\hat{\boldsymbol{\psi}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{w}.$$