Excerpts from "Bishop's Pattern Recognition and Machine Learning" used for educational purpose. do not republish...

1.2.6 Bayesian curve fitting

Note: the Gaussian curve problem demonstrated, can be/is solved analytically

In a fully Bayesian approach, we should consistently apply the sum and product rules of probability, which requires, as we shall see shortly, that we integrate over all val- ues of w. Such marginalizations lie at the heart of Bayesian methods for pattern recognition.

given the training data $\mathbf{x}$ and $\mathbf{t}$, along with a new test point x, and our goal is to predict the value of t.

we assume that the parameters α and β are fixed and known in advance (in later chapters we shall discuss how such parameters can be inferred from data in a Bayesian setting).

predictive distribution

$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w})p(\mathbf{w}|\mathbf{x}, \mathbf{t})\,\mathrm{d}\mathbf{w}. \tag{1.68}$$

is the posterior distribution

can be found by normalizing the right-hand side of (1.66).

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha). \tag{1.66}$$

posterior ∝ likelihood × prior

omitted the dependence on α and β to simplify the notation

target values $\mathbf{t} = (t_1, \ldots, t_N)^{\mathrm{T}}$. *(sometimes t is labels)*

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1.1}$$

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right) \tag{1.60}$$

β corresponding to the inverse variance of the distribution.

N input values $\mathbf{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$ *where x are observations or predictions*

1.2.6 Bayesian curve fitting (cont.)

solving 1.60:

- solve for $\mathbf{w}_{\mathrm{ML}}$ by minimizing the negative log-likelihood:

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi). \qquad (1.62)$$

$$\frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 = 0$$

where $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$

One can use regularization to avoid overfitting by a polynomial order that is too high:

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where $\|\mathbf{w}\|^2 \equiv \mathbf{w}^{\mathrm{T}}\mathbf{w} = w_0^2 + w_1^2 + \ldots + w_M^2$, and the coefficient λ governs the relative importance of the regularization term compared with the sum-of-squares error term. Note that often the coefficient w_0 is omitted from the regularizer because its

- solve for β_{ML}

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}_{\mathrm{ML}}) - t_n\}^2. \qquad (1.63)$$

1.2.6 Bayesian curve fitting (cont.)

then 1.60 becomes 1.64:

$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}\left(t|y(x, \mathbf{w}), \beta^{-1}\right) \qquad (1.60)$$

$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}\left(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1}\right). \qquad (1.64)$$

for the prior of 1.66 consider a Gaussian over the polynomial coefficients of w:

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\} \qquad (1.65)$$

where α is the precision of the distribution, and $M+1$ is the total number of elements in the vector $\mathbf{w}$ for an M^{th} order polynomial. Variables such as α, which control

- solve for $\mathbf{w}$ by minimizing the negative log-likelihood of 1.66:

$$\frac{\beta}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}. \qquad (1.67)$$

Thus we see that maximizing the posterior distribution is equivalent to minimizing the regularized sum-of-squares error function encountered earlier in the form (1.4), with a regularization parameter given by $\lambda = \alpha/\beta$.

Integration of 1.68 can be solved analytically:

predictive distribution

$$p(t|x,\mathbf{x},\mathbf{t}) = \int p(t|x,\mathbf{w})p(\mathbf{w}|\mathbf{x},\mathbf{t})\,\mathrm{d}\mathbf{w}. \tag{1.68}$$

$$p(t|x,\mathbf{x},\mathbf{t}) = \mathcal{N}\left(t|m(x),s^2(x)\right) \tag{1.69}$$

where the mean and variance are given by

$$m(x) = \beta\boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)t_n \tag{1.70}$$

$$s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^{\mathrm{T}}\mathbf{S}\boldsymbol{\phi}(x). \tag{1.71}$$

Here the matrix $\mathbf{S}$ is given by

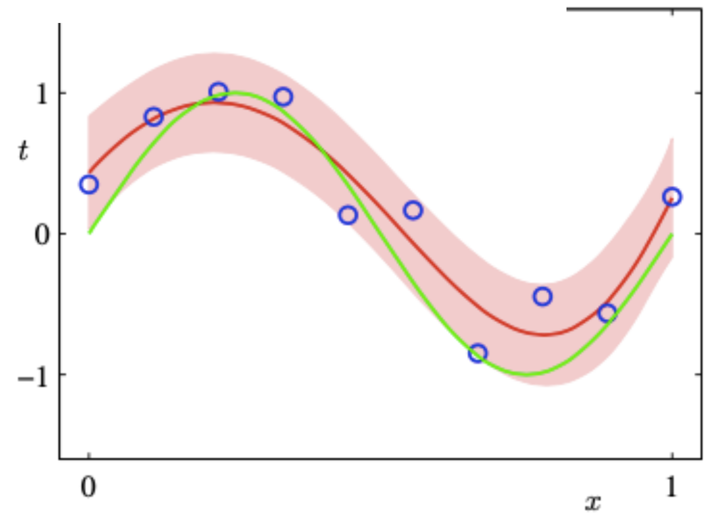
$$\mathbf{S}^{-1} = \alpha\mathbf{I} + \beta\sum_{n=1}^{N}\boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x)^{\mathrm{T}} \tag{1.72}$$

where $\mathbf{I}$ is the unit matrix, and we have defined the vector $\boldsymbol{\phi}(x)$ with elements $\phi_i(x) = x^i$ for $i = 0, \ldots, M$.

x = 10 data points drawn randomly from 0 to 1.
y = the values generated from sin(2πx)
or use those from figures:
x = (0., 6./60, 13./60, 21./60, 27./60, 34./60, 40./60, 47./60, 54./60, 60./60)
y = (3.5/9, 7.5/9, 9./9, 8.5/9, -1./9, -1.5/9, -8./9, -4./9, -5./9, 2.5.9)
alpha = 5E-3
beta = 11.1

Figure 1.17

The predictive distribution resulting from a Bayesian treatment of polynomial curve fitting using an $M = 9$ polynomial, with the fixed parameters $\alpha = 5 \times 10^{-3}$ and $\beta = 11.1$ (corresponding to the known noise variance), in which the red curve denotes the mean of the predictive distribution and the red region corresponds to ± 1 standard deviation around the mean.



where $\mathbf{I}$ is the unit matrix, and we have defined the vector $\boldsymbol{\phi}(x)$ with elements $\phi_i(x) = x^i$ for $i = 0, \ldots, M$.

ϕ_i(x) = x^i. let x=x_0=0. then ϕ_i(x_0) = 0^i. ϕ(x_0)=(0^0,0^1,0^2,…0^9)

$$\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^{N} \boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x)^{\mathrm{T}} \tag{1.72}$$

x = 10 data points drawn randomly from 0 to 1.
y = the values generated from sin(2πx)
or use those from figures:
x = (0., 6./60, 13./60, 21./60, 27./60, 34./60, 40./60, 47./60, 54./60, 60./60)
y = (3.5/9, 7.5/9, 9./9, 8.5/9, -1./9, -1.5/9, -8./9, -4./9, -5./9, 2.5/9)
alpha = 5E-3
beta = 11.1
M = order of fit = 9 for this example

ϕ_i(x) = x^i. let x=x_0=0. then ϕ_i(x_0) = 0^i. ϕ(x_0)=(0^0,0^1,0^2,…0^9)
1 to N are data element numbers from 1 to 10 (or 0-9)
0 to M are the order (power exponents)

ϕ_0(x) = | 0.^0 |
[Nx1]
| 6./60^0 |
| 13./60^0 |
| 21./60^0 |
| 27./60^0 |
| 34./60^0 |
| 40./60^0 |
| 47./60^0 |
| 54./60^0 |
| 60./60^0 |

ϕ(x) = | 0.^0, … 0.^9 |
[NxM]
| 6./60^0, … 6./60^9 |
| 13./60^0, … 13./60^9 |
| 21./60^0, … 21./60^9 |
| 27./60^0, … 27./60^9 |
| 34./60^0, … 34./60^9 |
| 40./60^0, … 40./60^9 |
| 47./60^0, … 47./60^9 |
| 54./60^0, … 54./60^9 |
| 60./60^0, … 60./60^9 |
…

ϕ(x_0)^T=(0^0,0^1,0^2,…0^9)
ϕ(x_1)^T=((6./60)^0,(6./60)^1,…(6./60)^9)
ϕ(x_9)^T=((60./60)^0,(60./60)^1,…(60./60)^9)

each is **[1XM]**

search literature. Vectors are denoted by lower case bold Roman letters such as $\mathbf{x}$, and all vectors are assumed to be column vectors. A superscript T denotes the transpose of a matrix or vector, so that $\mathbf{x}^{\mathrm{T}}$ will be a row vector. Uppercase bold roman letters, such as $\mathbf{M}$, denote matrices. The notation $(w_1, \ldots, w_M)$ denotes a row vector with M elements, while the corresponding column vector is written as $\mathbf{w} = (w_1, \ldots, w_M)^{\mathrm{T}}$.

If we have N values $\mathbf{x}_1, \ldots, \mathbf{x}_N$ of a D-dimensional vector $\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$, we can combine the observations into a data matrix $\mathbf{X}$ in which the n^{th} row of $\mathbf{X}$ corresponds to the row vector $\mathbf{x}_n^{\mathrm{T}}$. Thus the n, i element of $\mathbf{X}$ corresponds to the i^{th} element of the n^{th} observation $\mathbf{x}_n$. For the case of one-dimensional variables we shall denote such a matrix by x, which is a column vector whose n^{th} element is x_n. Note that x (which has dimensionality N) uses a different typeface to distinguish it from $\mathbf{x}$ (which has dimensionality D).

here, x is a row vector

training data comprising N input values $\mathsf{x} = (x_1, \ldots, x_N)^{\mathrm{T}}$

original data set consists of N data points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

ϕ(x)ᵀ∗ϕ(x)

$$\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^{N} \boldsymbol{\phi}(x_n)\boldsymbol{\phi}(x_n)^{\mathrm{T}} \quad \textbf{[MXM]} \tag{1.72}$$

$$m(x) = \beta \cancel{\boldsymbol{\phi}(x)}^{\mathrm{T}} \mathbf{S} \sum_{n=1}^{N} \boldsymbol{\phi}(x_n) t_n \tag{1.70}$$

~~[MXN]~~[MXM][MX1]=[MX1] error in eqn. see ctgk PRML

$$s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^{\mathrm{T}} \mathbf{S} \boldsymbol{\phi}(x). \tag{1.71}$$

[][MXM][] = [MXM] error in eqn. see ctgk PRML github

$$p(t|x, \mathsf{x}, \mathsf{t}) = \mathcal{N}\left(t|m(x), s^2(x)\right) \tag{1.69}$$