1 Supervised Learning vs. Unsupervised

1.1 Unsupervised

Data does not come with labels: \{x_1, \ldots, x_n\}. Aims to predict next data point: x_{n+1}

1.2 Supervised

Data comes in pairs: \begin{pmatrix} x_1 \\ t_1 \end{pmatrix} \ldots \begin{pmatrix} x_n \\ t_n \end{pmatrix}. Aims to answer the relationship between x \rightarrow t and x_{n+1} \rightarrow t_{n+1} where t is the output of the model. Two main types of supervised learning: regression and classification.

2 Linear Models for Regression

2.1 Simplest Model: Linear Regression

\[ y(x; w) = w_0 + w_1 x_1 + \ldots + w_D x_D \]

Limited because constrained to model y as a linear function of the data x.

2.2 Linear Basis Function Models

\[ y(x; w) = w_0 + w_1^T \varphi_1(x) + w_2^T \varphi_2(x) + \ldots + w_{m-1}^T \varphi(x) \]

\[ = w^T \varphi(x) \]

Where \( m = \)number of parameters, \( w^T = [w_0 \ldots w_{m-1}]^T \), and \( \varphi(x) = [\varphi_0 \ldots \varphi_{m-1}]^T \)

\( \Rightarrow y \) can be a nonlinear function of x, if \( \varphi \) is nonlinear [analagous to \( u(x) \) in standard rep. of exponential family: \( \exp(\eta^T u(x)) \)].

3 Examples of linear basis function models

3.1 Polynomial

\[ \varphi_j(x) = x^j \Rightarrow \varphi_0(x) = 1, \varphi_1(x) = x, \ldots, \varphi_{m-1}(x) = x^{m-1} \]
\[ y(x; w) = w_0 + w_1 x + \ldots + w_{m-1} x^{m-1} \]

Limitation of polynomial model: basis function are global (changes in one part of space affect everywhere else) and difficulty in numerical optimization).

### 3.2 “Gaussian” basis functions

\[ \phi_0(x) = 1, \quad \phi_j(x) = \exp\left(-\frac{(x - \mu_j)^2}{2\sigma^2}\right), \quad y(x; w) = w^T \varphi(x) \]

Where the basis functions are local in extent and we are trying to fit the function: \( y(x; w) \).

### 3.3 (logistic) sigmoid

\[ \phi_0 = 1; \quad \phi_j(x) = \sigma \left( \frac{x - \mu_j}{\sigma} \right), \text{ where } \sigma(a) = \frac{1}{1 + e^{-a}} \]

### 4 Least Squares for Regression

Assume a linear basis function model where \( \varepsilon = \mathcal{N}(0, \beta^{-1}) \) and \( \beta^{-1} = \text{precision} \).

\[
\begin{align*}
t &= y(x; w) + \varepsilon \\
  &= w^T \varphi(x) + \varepsilon \\
  \Rightarrow t &\sim \mathcal{N}(w^T \varphi(x), \beta^{-1})
\end{align*}
\]

Minimum sum-of-squares error function:

\[
E_D(w) \triangleq \frac{1}{2} \sum_{i=1}^{n} (t_i - w^T \varphi(x))^2
\]

MLE:

\[
\begin{align*}
p(t; w, \beta) &= \prod_{i=1}^{n} p(t_i; w, \beta) \\
  &= \prod_{i=1}^{n} \mathcal{N}(t_i; w^T \varphi(x_i), \beta^{-1})
\end{align*}
\]

\[
\begin{align*}
\log p(t; w, \beta) &= \sum_{i=1}^{n} \log \mathcal{N}(t_i; w^T \varphi(x_i), \beta^{-1}) \\
  &= \frac{n}{2} \log(2\pi) - \beta \left( \frac{1}{2} \sum_{i=1}^{n} (t_i - w^T \varphi_i)^2 \right)
\end{align*}
\]

max with respect to \( w = \min E_D(w) \)
5 Regularized Least Square (for Regression)

Total error function: $E(w) = E_D(w) + \lambda E_W(w)$ where $E_W(w) = w^T w$

assumption: $N(0, \alpha^{-1}I)$
prior: $p(w) = N(\mu_0, S_0)$
likelihood: $p(t \mid w) = N(\Phi w, p^{-1}I)$

design matrix: $\Phi = \begin{bmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & \cdots & \phi_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & \cdots & \phi_{M-1}(x_N) \end{bmatrix}$

posterior: $p(w \mid t) = N(m_n, S_n)$
where $m_n = \beta S_n \Phi^T t$ and $S_n = \alpha I + \beta \Phi^T \Phi$ (see Bishop 2.116 for more detailed explanation)

max:

$$\log \text{posterior} = \text{constant} + \log \text{prior} + \log \text{likelihood}$$

$$= -\frac{\alpha}{2} w^T w - \frac{\beta}{2} \sum_{i=1}^{n} (t_i - w^T \varphi_i)^2 + \text{constant}$$

min:

$$E(w) = \frac{1}{2} \sum_{i=1}^{n} (t_i - w^T \varphi_i)^2 + \frac{\alpha}{\beta} w^T w$$

comparing this to the total error function above we see that $\alpha = \text{prior precision}$ and $\lambda = \frac{\alpha}{\beta}$