Intro

Paradigms

- Supervised Learning

  $Given\ D = \{X_i, Y_i\},\ learn\ f(\cdot) : Y_i = f(X_i),\ s.t. D^{new} = \{X_j\} => \{Y_j\}$
- Unsupervised Learning

  $Given\ D = \{X_i\},\ learn\ f(\cdot) : Y_i = f(X_i),\ s.t. D^{new} = \{X_j\} => \{Y_j\}$

Example

Polynomial curve fitting

Fit the data using a polynomial function of the form:

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

$\mathbf{w}$ is the parameters we need to adapt according to dataset $\{(x_n, y_n)\}_N$

Minimize "loss function" to find the w:

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

Overfitting

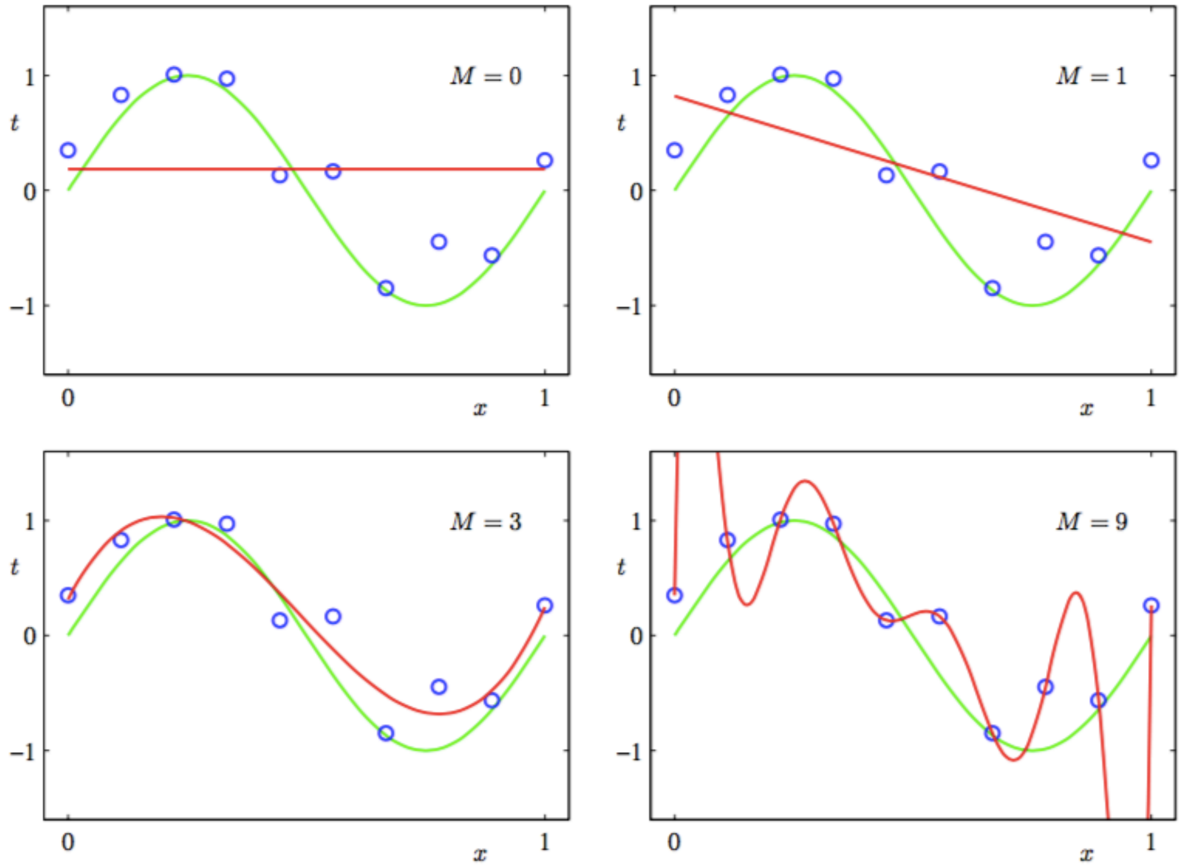


Figure 1.4 Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.2.

For M = 9, the training set error goes to zero, while test set error become very large due to overfitting. The reason is that we have 10 coefficients($w_0\ to\ w_9$) thus containing **10 degrees of freedom, and so they can be tuned exactly to the 10 data points in the training set**.

Avoid overfitting(1)

More data

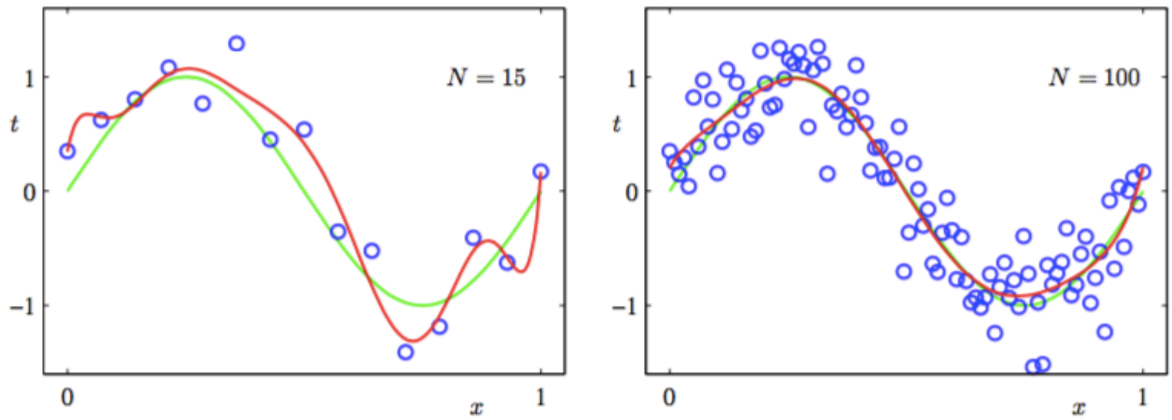


Figure 1.6 Plots of the solutions obtained by minimizing the sum-of-squares error function using the $M = 9$ polynomial for $N = 15$ data points (left plot) and $N = 100$ data points (right plot). We see that increasing the size of the data set reduces the over-fitting problem.

Avoid overfitting(2)

Loss function with **panalty item(or regularization)** on ||w||

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

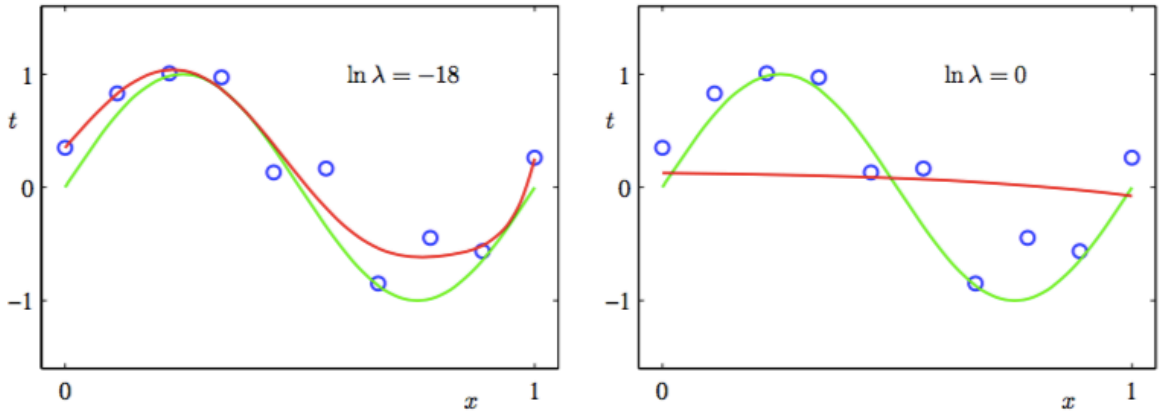


Figure 1.7 Plots of $M = 9$ polynomials fitted to the data set shown in Figure 1.2 using the regularized error function (1.4) for two values of the regularization parameter λ corresponding to $\ln \lambda = -18$ and $\ln \lambda = 0$. The case of no regularizer, i.e., $\lambda = 0$, corresponding to $\ln \lambda = -\infty$, is shown at the bottom right of Figure 1.4.

Probability Theory

Rules of Probability

- **sum rule:** $p(Y) = \sum_Y p(X, Y)$
- **product rule:** $p(Y, X) = p(Y|X)P(X)$
- **Bayes' theorem:** $p(Y|X) = \frac{p(X|Y)p(Y)}{P(X)},\ P(X) = \sum_Y p(X|Y)p(Y)$

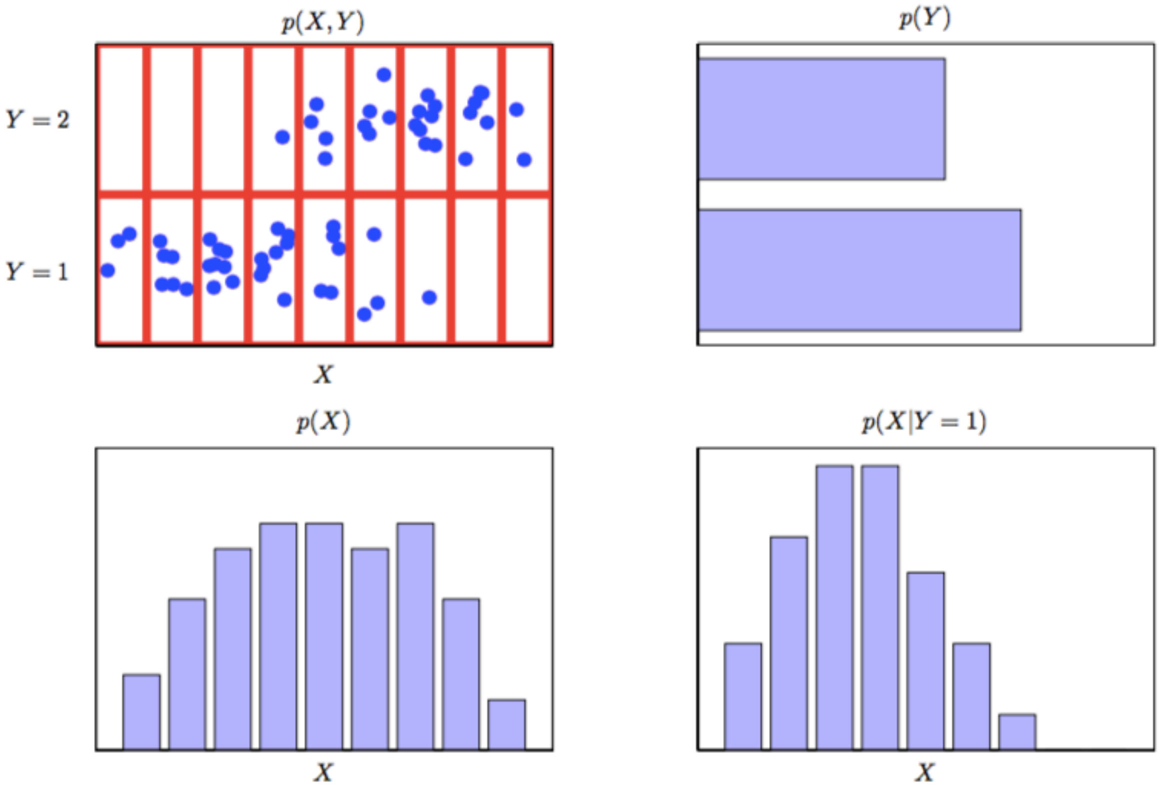


Figure 1.11 An illustration of a distribution over two variables, X, which takes 9 possible values, and Y, which takes two possible values. The top left figure shows a sample of 60 points drawn from a joint probability distribution over these variables. The remaining figures show histogram estimates of the marginal distributions $p(X)$ and $p(Y)$, as well as the conditional distribution $p(X|Y=1)$ corresponding to the bottom row in the top left figure.

Probability densities

$p(x \in (a,b)) = \int_a^b p(x)dx,\ p(x) \geq 0:\ density\ function$

- Note: Under a **nonlinear change of variable**, a probability density transforms differently from a simple function, due to the **Jacobian factor**.

$$\begin{aligned} &given\ x = g(y) \\ \because\ &p_x(x)dx \simeq p_y(y)dy \\ \therefore\ &p_y(y) = p_x(x)|\frac{dx}{dy}| \\ &= p_x(g(y)) \end{aligned}$$

  One consequence of this property is that the concept of the maximum of a probability density is **dependent on the choice of variable**.

Expectations and covariances

$E[f] = \sum_x p(x)f(x),\ E[f] = \int p(x)f(x)dx$

$var[f] = E[(f(x) - E[f])^2] = E[f^2] - E[f(x)]^2$

$cov[x,y] = E_{x,y}[(x - E[x])(y - E[y])] = E_{x,y}[xy] - E[x]E[y]$

$cov[\mathbf{x},\mathbf{y}] = \mathbf{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x} - \mathbf{E}[\mathbf{x}]\mathbf{y}^\mathbf{T} - \mathbf{E}[\mathbf{y}^\mathbf{T}]] = \mathbf{E}_{\mathbf{x},\mathbf{y}}[\mathbf{x}\mathbf{y}^\mathbf{T}] - \mathbf{E}[\mathbf{x}]\mathbf{E}[\mathbf{y}^\mathbf{T}]$

Bayes' View

Bayes' theorem was used to **convert a prior probability into a posterior probability by incorporating the evidence provided by the observed data**.

Prior probability can be regarded as **knowledge gained before or "common sense"**.

From frequests' view, the **w** learned from dataset is fixed(by maximize likelihood function), while From Bayes' view, it's an uncertain variable represented by a probability distribution $p(\mathbf{w})$.

Common path of Bayes' learning:

Loop

1. prior: $p(\mathbf{w})$
2. Observed dataset: $D = t_1, \ldots, t_N$
3. Posterior: $p(\mathbf{w}|\mathbf{D}) = \frac{\mathbf{p(D|w)p(w)}}{\mathbf{p(D)}}$ and regard it as new prior(updated by observations).

$posterior \propto likelihood \times prior$

$p(D) = \int \mathbf{p(D|w)p(w)dw}$

Gaussian distribution

$\mathcal{N}(x|u, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\{-\frac{1}{2\sigma^2}(x-u)^2\}$

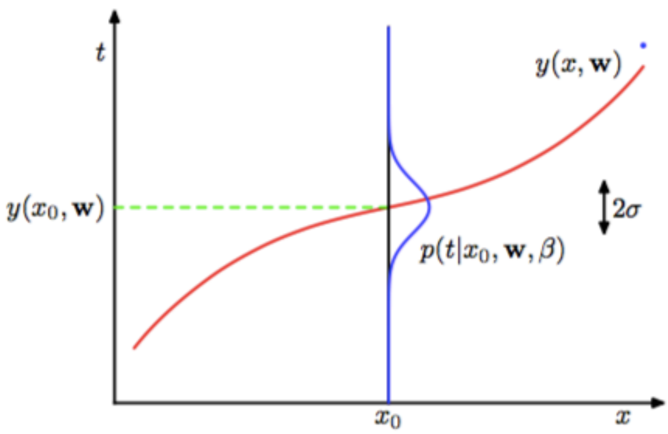
$\mathcal{N}(x|u, \sum) = \frac{1}{\sqrt{(2\pi)^D|\sum|}} exp\{-\frac{1}{2}(x-u)^T \sum^{-1}(x-u)\}$

Revisit Curve fitting

Given dataset: $\mathbf{x} = (x_1, \ldots, x_N)^T$, $\mathbf{t} = (t_1, \ldots, t_N)T$

We assume: $p(t|x, \mathbf{w}, \beta) = \mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$

Figure 1.16 Schematic illustration of a Gaussian conditional distribution for t given x given by (1.60), in which the mean is given by the polynomial function $y(x, \mathbf{w})$, and the precision is given by the parameter β, which is related to the variance by $\beta^{-1} = \sigma^2$.



Likelihood function:

$$
\begin{aligned}
p(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) &= \prod_{n=1}^{N} \mathcal{N}(t_n|\ y(x_n,\mathbf{w}),\beta^{-1}) \\
&= \prod_{n=1}^{N} (\frac{\beta}{2\pi})^{\frac{1}{2}} e^{-\frac{\beta}{2}(y(x_n,\mathbf{w})-t_n)^2}
\end{aligned}
$$

Log likelihood:

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

Maximize log likelihood with respect to $\mathbf{w}$ is equivalent to maximize **sum-of-squares error function** defined before. While maximize it with respect to β gives:

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

Predict:

$$
p(t|x,\mathbf{w_{ML}},\beta_{\mathbf{ML}}) = \mathcal{N}(t|y(x,\mathbf{w_{ML}}),\beta_{ML}^{-1})
$$

Introduce a prior distribution over $\mathbf{w}$:

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

posterior for $\mathbf{w}$:

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

Maximum posterior(MAP):

$\mathbf{w_{MAP}} = \min_{\mathbf{w}} \mathbf{ln\ p(w|x,t,\alpha,\beta)} = \min_{\mathbf{w}}\{\frac{\beta}{\mathbf{2}}\sum_{\mathbf{n=1}}^{\mathbf{N}}[\mathbf{y(x_n,w)-t_n}]^{\mathbf{2}} + \frac{\alpha}{\mathbf{2}}\mathbf{w^Tw}\}$

Thus, maximizing the posterior distribution is equivalent to minimizing the regularized sum-of-squares error function.

Bayesian Curve fitting

MAP(maximizing the posterior) and ML(maximizing the likelihood) are both "**point estimate**" methods.

A more Bayesian way is introduced:

$$
\begin{aligned}
p(t|x,\mathbf{x},\mathbf{t}) &= \int p(t|x,\mathbf{w})p(\mathbf{w}|\mathbf{t},\mathbf{x})d\mathbf{w} \\
&= \mathcal{N}(t|m(x),s^2(x))
\end{aligned}
$$

where the mean and var will be given and discussed in detail in Chapter 3.

Problems when dimen goes high

- too many coefficients.
- Our geometrical intuitions, formed through a life spent in a space of three dimensions, can fail badly when we consider spaces of higher dimensionality.

  Example:

  Consider a sphere of radius r = 1 in a space of D dimensions, and ask what is the fraction of the volume of the sphere that lies between radius r = 1 − ϵ and r = 1. We can evaluate this fraction by noting that the volume of a sphere of radius r in D dimensions must scale as r^D, and so we write

  $$V_D(r) = K_D r^D$$
  $$\frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = 1 - (1-\epsilon)^D$$

  for large D(high dimen space), the fraction tends to 1 even for small ϵ. Which means most of the volume of a sphere in high-dimen space is **concentrated in a thin shell near the surface.**

Decision theory

Two stages of typical machine learning:

- Inference: Determine $p(\mathbf{x}, \mathbf{t}) \ or \ p(\mathbf{t}|\mathbf{x})$ from training dataset.
- Decision: choose the best $\mathbf{t}$ given $\mathbf{x} \ and \ p(\mathbf{x}, \mathbf{t}) \ or \ p(\mathbf{t}|\mathbf{x})$

  $$\begin{aligned} p(mistake) &= p(x \in R_1, C_2) + p(x \in R_2, C_1) \\ &= \int_{R_1} p(x, C_2)dx + \int_{R_2} p(x, C_1)dx \end{aligned}$$

  so, we should choose the bigger $p(x, C_k)$ to minimize the mistake. Also, because $p(x, C_k) = p(x|C_k)p(x)$, we should choose the bigger $p(x|C_k)$.

reject option

Introduce a threshold θ and make no decisions when the largest probability $p(C_k|\mathbf{x})$ is less than θ

generative and discriminative

classification problem

(a) First solve the inference problem of $p(x|C_k) \ and \ p(C_k)$, then use Bayes' theorem to get $p(C_k|x)$

Equivalently, we can model $p(x, C_k)$ and then normalize to obtain the posterior distribution.

(b) Solve the inference problem of $p(C_k|x)$ directly.

- (a) is known as **generative model**, (b) is called **discriminative model**

(a) need more computation but provides more information about data distribution. (b) is more efficient.

Example:

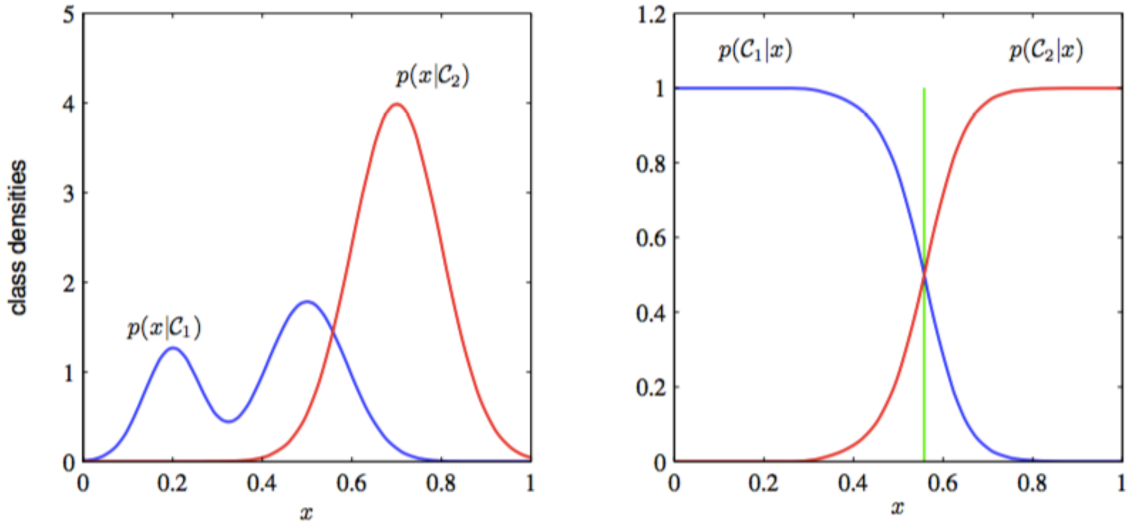


Figure 1.27 Example of the class-conditional densities for two classes having a single input variable x (left plot) together with the corresponding posterior probabilities (right plot). Note that the left-hand mode of the class-conditional density $p(\mathbf{x}|\mathcal{C}_1)$, shown in blue on the left plot, has no effect on the posterior probabilities. The vertical green line in the right plot shows the decision boundary in x that gives the minimum misclassification rate.

$p(C_k|x)$ can be used to determine class directly, while $p(x|C_k)$ contains raw info about data distribution.

regression problem

(a) First solve the inference problem of determining the joint density $p(x,t)$. Then normalize to find the conditional density $p(t|x)$, and finally marginalize to find the conditional mean.

(b) First solve the inference problem of determining the conditional density $p(t|x)$, and then subsequently marginalize to find the conditional mean.

(c) Find a regression function $y(x)$ directly from the training data.