Bayes rule \( \rightarrow \) posterior
\[
P(C_i | x) = \frac{p(x | C_i) p(C_i)}{\frac{\sum_{k=1}^{K} p(x | C_k) p(C_k)}{K}}
\]

discriminant function
\[
\tilde{g}_i(x) = p(x | C_i) p(C_i)
\]
or
\[
g_i(x) = \log p(x | C_i) + \log p(C_i).
\]

If Gaussian:
\[
p(x | C_i) = \frac{1}{\sqrt{2\pi \sigma_i^2}} e^{-\frac{(x - \mu_i)^2}{2\sigma_i^2}}
\]

Then
\[
g_i(x) = -\frac{1}{2} \log 2\pi - \log \sigma_i - \frac{(x - \mu_i)^2}{2\sigma_i^2} + \log p(C_i)
\]

Example: Car Company \( x \) income.
\( C_i \) customers who buy type 1.

Sample \( x = \langle x^1, x^2, \ldots \rangle \), \( i \in C_0 \cup C_1 \)

\( g_i \in \{1, 0\} \) if \( x^t \in C_i \)
\( = 0 \) if \( x^t \in C_k, k \neq i \)

For each class - estimate of mean & variance
\[
m_i = \frac{\sum x^t r_i^t}{\sum r_i^t}, \quad s_i^2 = \frac{\sum (x^t - m_i)^2 r_i^t}{\sum r_i^t}
\]

Estimate prior
\[
\hat{p}(C_i) = \frac{\sum r_i^t}{\sum r_i^t}, \quad \sum r_i^t
\]
Plugging estimates into discriminant function:

\[ g_i(x) = -\frac{1}{2} \log 2\pi - \log \sigma_i - \frac{(x-m_i)^2}{2\sigma_i^2} + \log \hat{\pi}(y) \]

**Common Simplification:**

(i) \( \hat{\pi}(y) \) constant, \( \sigma_i \) constant (map of \( i \))

dropping terms that are constant:

\[ g_i(x) = -\frac{(x-m_i)^2}{2\sigma_i^2} \]

Choose \( C_i \) of \( |x-m_i| = \min_k |x-m_k| \) nearest model.

![Decision boundary diagram](image)

- **Decision boundary is**
  - at \( x = \frac{1}{2} (m_1 + m_2) \)

**Note:**

1. If we have prior knowledge of \( \{m_i, \sigma_i\} \)
   then we should estimate them by MAP & not ML.

2. Beware of Gaussian assumption. Gaussian is non-robust. We can make big errors if we assume data is Gaussian but it isn’t.

Statistics literature - has tests for Gaussian nature.

At least - look at the data to see if it has Bell shape.
\[ E[R(x)] = w_1 x + w_0 \]

Write output (dependent variable) as a function of the input (independent variable).

\[ y = f(x) + \epsilon \]

Output = unknown function of input

Write to approximate \( f(x) \) by an estimator \( g(x|\theta) \).

\[ \theta \text{ - unknown parameter} \]

Standard assumption:

\[ \epsilon \sim N(0, \sigma^2) \]

\[ p(r(x)) = N(g(x|\theta), \sigma^2) \]

Use ML to learn the parameters \( \theta \)

\[ p(x, r) = p(r(x)|p(x)) \quad x = \{x^t, r^1, \ldots, r^N\} \]

Log likelihood:

\[ L(\theta|X) \]

\[ = \prod_{t=1}^{N} p(x^t, \epsilon^t) \]

\[ = \sum_{t=1}^{N} \log p(r^t|x^t) + \sum_{t=1}^{N} \log p(x^t) \]

\[ L(\theta|X) \]

\[ = -N \log(\sqrt{2\pi\sigma}) - \frac{1}{2} \sum_{t=1}^{N} \left[ \frac{r^t - g(x^t|\theta)}{\sigma^2} \right]^2. \]

Max w.r.t. \( \theta \) is equivalent to min.

\[ \sum_{t=1}^{N} \left[ r^t - g(x^t|\theta) \right]^2 \]

Least square estimator

(Gaussian distribution - quadratic minimization)
Regression (Cont.)

Linear regression: \( g(x^t | w_0, w_1) = w_1 x^t + w_0. \)

Differentiate energy w.r.t. \( w_0, w_1 \) gives two equations:

\[
\sum_t r^t = NW_0 + W_1 \sum_t x^t
\]

\[
\sum_t r^t x^t = W_0 \sum_t x_t + W_1 \sum_t (x_t)^2 .
\]

Expressed in linear algebra form as: \( A \mathbf{w} = \mathbf{y} \)

\[
A = \begin{bmatrix} N & \sum_t x^t \\ \sum_t x^t & \sum_t (x^t)^2 \end{bmatrix}, \quad w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad y = \begin{bmatrix} \sum_t r^t \\ \sum_t r^t x^t \end{bmatrix}
\]

Solved to give: \( \mathbf{w} = A^{-1} \mathbf{y}. \)

Polynomial Regression:

\( g(x^t | w_0, w_1, \ldots, w_k, w_{k+1}) = w_0 (x^t)^k + \ldots + w_k x^t + w_{k+1}. \)

Let \( k+1 \) parameters \( w_0, \ldots, w_{k+1}. \)

Diff. energy - gives \( k+1 \) linear eqns in \( k+1 \) vars.

Can write: \( A = \mathbf{D}^T \mathbf{D} , \quad y = \mathbf{D}^T \mathbf{r} \)

\[
\mathbf{D} = \begin{bmatrix} x^t_1 \\ \vdots \\ x^t_t \end{bmatrix}
\]

Solve to get:\( \mathbf{w} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{r} \)

\[
r = \begin{bmatrix} r^1 \\ \vdots \\ r^t \end{bmatrix}
\]

Must adjust the complexity of the model to the amount of data available.

Complexity of poly regression is \( k \) parameters \( k \).

Need to pick \( k \) to give best generalization error.
1. **Tuning Model Complexity: Bias/Variance Dilemma.**

Sample $X = \{x^i, r^i\}$ drawn from unknown $p(x, r)$.

Construct an estimate $g(\cdot)$.

Expected Square Error can be expressed as:

$$E_X (r - g(x))^2 | x] = \frac{E_X [r - E_X r(x)]^2 | x]}{\text{noise}} + \frac{(E_X r(x) - g(x))^2}{\text{square error}}$$

Can't be removed no matter what estimator we use. Doesn't depend on $g$ or $x$.

$$E_X \{ (E_X r(x) - g(x))^2 | x] = (E_X r(x) - E_X g(x))^2$$

$$+ E_X \{ (g(x) - E_X g(x))^2$$

Variance.

2. **Inductive Example**

Average on datasets to quantify how good $g$ is.

$X$ is the sample $\{ x^i, r^i \}$

$E_X$ is averaging draw the sample from distribution $p(x, r)$.

Generate a set of datasets $X_i = \{ x^i, r^i \}$

$i = 1$ to 10.

Use each dataset to make an estimate $g_i(\cdot)$.

Then estimate

$$E_T g(x) = \frac{1}{N} \sum_{i=1}^{m} g_i(x)$$

$$\text{Bias}^2(g) = \frac{1}{N} \sum_{i=1}^{N} [g_i(x) - \bar{g}(x)]^2$$

$$\text{Variance}(g) = \frac{1}{N} \sum_{i=1}^{N} [g_i(x) - \bar{g}(x)]^2.$$
A more complex models gives better fit to the data (i.e. to underlying model)
→ reduces bias.

But small changes in dataset lead to big change in fitted model
→ increases variance.

Low orders - risk of underfitting
High orders - risk of overfitting.

To get a small error we should have the proper inductive bias and have large enough dataset so that variability is constrained by data.

Note: take very high-variance models, use average (lale)
Model Selection Procedure

1. Cross-validation: divide dataset into two parts as training & validation set.
   - Train models of different complexity and test their error on the validation set.
   - As model complexity increases, error on training set decreases. But error on validation set decreases and increases.

   \[ e' = e_{\text{error on data}} + \lambda \cdot \text{model complexity} \]  

2. Regularization

   - Augment error function
   - Also penalizes model complexity

3. Structural Risk Minimization (Vapnik)

4. Minimum Description Length (Rissanen)

   - Penalize complexity by cost of encoding model

5. Bayesian Model Selection

   - If some prior knowledge
   \[ p(\text{model} | \text{data}) = \frac{p(\text{data} | \text{model})p(\text{model})}{p(\text{data})} \]  
   (gives higher prob to simpler models)