(1)

Alpaydin (chp 4) cont'd (4.5)

Note Title 3/27/2008

Parametric Classification.

Bayes rule → posterior

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{\sum_{k=1}^{K} p(x|C_k)P(C_k)}$$

discriminant function

$$g_i(x) = p(x|C_i)P(C_i)$$

or $$g_i(x) = \log p(x|C_i) + \log P(C_i).$$

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

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 $X = \{x^t, r^t\}_{t=1}^{N}$ $r \in \{0,1\}^K$

$$r_i^t = \begin{cases} 1, & \text{if } x^t \in C_i \\ 0, & \text{if } x^t \in C_k, k \neq i \end{cases}$$

For each class - estimates of mean & variance

$$m_i = \sum_t x^t r_i^t / \sum_t r_i^t, \quad s_i^2 = \sum_t (x^t - m_i)^2 r_i^t / \sum_t r_i^t$$

estimate prior $$\hat{P}(C_i) = \sum_t r_i^t / N$$

Parametric Classification (cont).

Plugging estimates into discriminant function

$$g_i(x) = -\frac{1}{2}\log 2\pi - \log s_i - \frac{(x-m_i)^2}{2s_i^2} + \log \hat{P}(C_i)$$

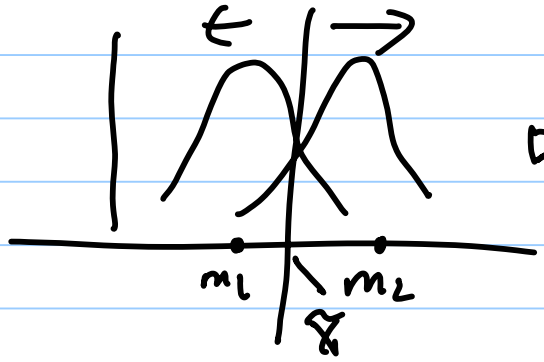
Common Simplification:

(i) $\hat{P}(C_i)$ constant, s_i constant (index of i)

dropping terms that are constant.

$$g_i(x) = -(x-m_i)^2.$$

Choose C_i if $|x-m_i| = \min_k |x-m_k|$ nearest mean.



Two classes:
Decision boundary is at $\bar{x} = \frac{1}{2}(m_1+m_2)$

Note: (i) If we have prior knowledge of $\{m_i, s_i\}$ then we should estimate them by MAP & not ML.

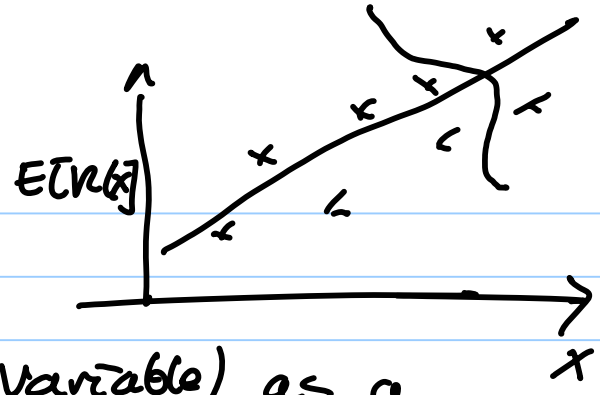
(ii) Beware of Gaussian assumption.

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-ness.

At least - look at the data to see if it has Bell shape.

(3) # Regression



$E[R|x] = wx + w_0$

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

$$r = f(x) + \epsilon$$

r: output; $f(x)$: unknown function of input; ϵ: random noise

Want to approximate $f(x)$ by an estimator $g(x|\theta)$, θ - unknown parameters

Standard assumption:

$$\epsilon \sim N(0, \sigma^2)$$

$$p(r|x) \sim N(g(x|\theta), \sigma^2)$$

Use ML to learn the parameters θ

$$p(x, r) = p(r|x)\,p(x) \qquad \mathcal{X} = \{x^t, r^t\}_{t=1}^N$$

log likelihood

$$\mathcal{L}(\theta|\mathcal{X}) = \log \prod_{t=1}^N p(x^t, r^t)$$

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

(indep of θ)

$$\mathcal{L}(\theta|\mathcal{X}) = -N \log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2} \sum_{t=1}^N [r^t - g(x^t|\theta)]^2$$

Max w.r.t. θ is equivalent to min. $\sum_{t=1}^N [r^t - g(x^t|\theta)]^2$

least square estimate

(Gaussian distribution → quadratic minimization)

(4) Regression (cont.)

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

Differentiate energy wrt w_1, w_0 gives two equations

$$\sum_t r^t = N w_0 + w_1 \sum_t x^t$$

$$\sum_t r^t x^t = w_0 \sum_t x_t + w_1 \sum_t (x^t)^2.$$

Expressed in linear algebra form as $\underline{A}\,\underline{w} = \underline{y}$

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

solved to give $\underline{w} = \underline{A}^{-1}\underline{y}$.

Polynomial Regression.

$$g(x^t | w_k, \ldots, w_2, w_1, w_0) = w_k (x^t)^k + \ldots + w_1 x^t + w_0$$

$k+1$ parameters $w_k \ldots w_0$

Diff. energy - gives $k+1$ linear eqn's in $k+1$ variables.

$$\underline{A}\,\underline{w} = \underline{y}$$

Can write $\underline{A} = \underline{D}^T \underline{D}$, $\underline{y} = \underline{D}^T \underline{r}$ $\quad \underline{D} = \begin{bmatrix} 1 & x_1 & x_1^k \\ 1 & x_2 & x_2^k \\ 1 & \cdot & \cdot \end{bmatrix}$

Solve to get $\underline{w} = (D^T D)^{-1} D^T r$ $\quad r = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \end{bmatrix}$

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

Complexity of poly regression is no. parameters k.

Need to pick k to give best generalization error

(5) Tuning Model Complexity: Bias/Variance Dilemma.

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

Construct an estimate $g(.)$.

Expected Square Error can be expressed as:

$$E[(r-g(x))^2|x] = \underbrace{E[(r-E[r|x])^2|x]}_{\text{noise}} + \underbrace{(E[r|x]-g(x))^2}_{\text{squared error}}$$

noise: can't be removed no matter what estimator we use → doesn't depend on g or $\mathcal{X}$

squared error: how much $g(x)$ varies for regression $E[r|x]$

$$E_{\mathcal{X}}[(E[r|x]-g(x))^2|x] = \underbrace{(E[r|x]-E_{\mathcal{X}}[g(x)])^2}_{\text{bias}} + \underbrace{E_{\mathcal{X}}[(g(x)-E_{\mathcal{X}}[g(x)])^2]}_{\text{variance}}$$

Average over datasets to quantify how good g is.

$\mathcal{X}$ is the samples $\{x^t, r^t\}$

$E_{\mathcal{X}}$ is averaging over the sample from distribution $p(x,r)$

Didactic Example

Generate a set of datasets $\mathcal{X}_i = \{x_i^t, r_i^t\}$, $i = 1$ to M.

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

Then estimate $\bar{g}(x) = \frac{1}{M}\sum_{i=1}^{M} g_i(x)$

$$\text{Bias}^2(g) = \frac{1}{N}\sum_t [\bar{g}(x^t) - f(x^t)]^2$$

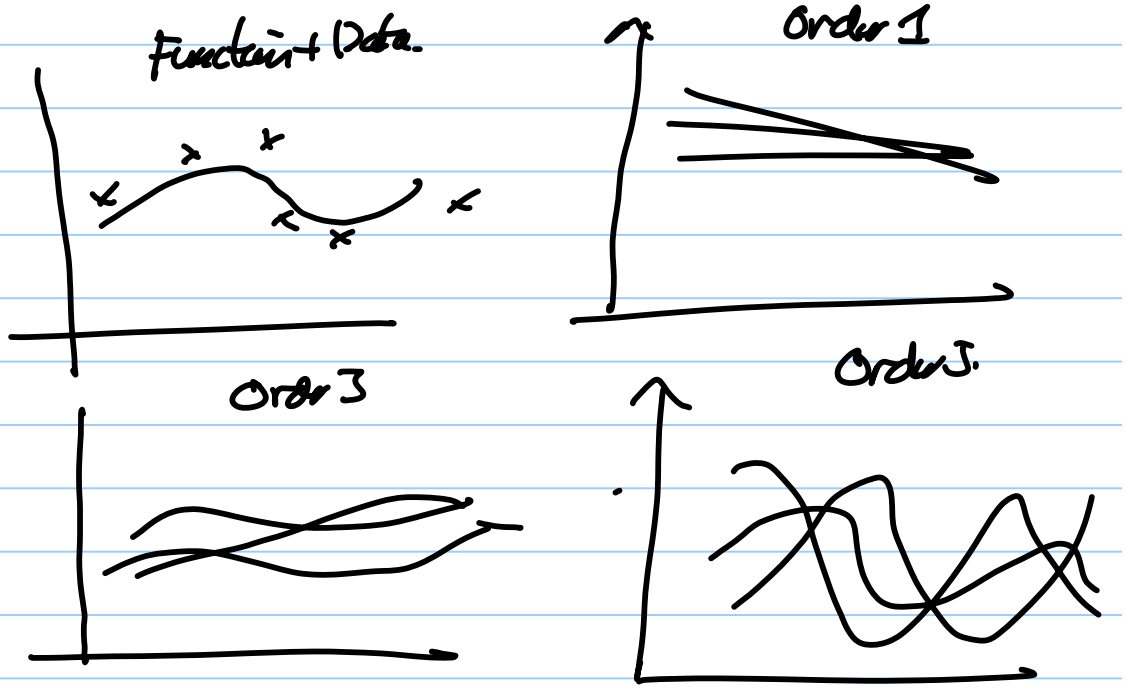
$$\text{Variance}(g) = \frac{1}{NM}\sum_t \sum_i [g_i(x^t) - \bar{g}(x^t)]^2$$

(6) Bias/Variance Dilemma cont'd

Examples:

$f(x) = 2\sin(1.5x)$

$\epsilon \sim N(0,1)$



A more complex models gives better fit to the data (i.e. to underlying model)
→ reduces bias.

But small changes in dataset lead to bigger variations in fitted model
→ increases variance.

Low orders — risk of underfitting
High orders — risk of overfitting. (fit the noise, not the function)

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 many high-variance models, use average (later)

(7) # Model Selection Procedures

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 then increases.

regularization

augmented error function

$E' = \text{error on data} + \lambda \cdot \text{model complexity}$.

(λ optimized using cross-validation)

structural risk minimization (Vapnik) (VC dim)

- also penalizes model complexity.

Minimum Description Length (Rissanen)

penalize complexity by cost of encoding model.

Bayesian Model Selection. if some prior knowledge

$$P(\text{model} \mid \text{data}) = \frac{P(\text{data} \mid \text{model})\, P(\text{model})}{P(\text{data})}$$

(gives higher prob to simpler models)