MODEL SELECTION AND ERROR ESTIMATION

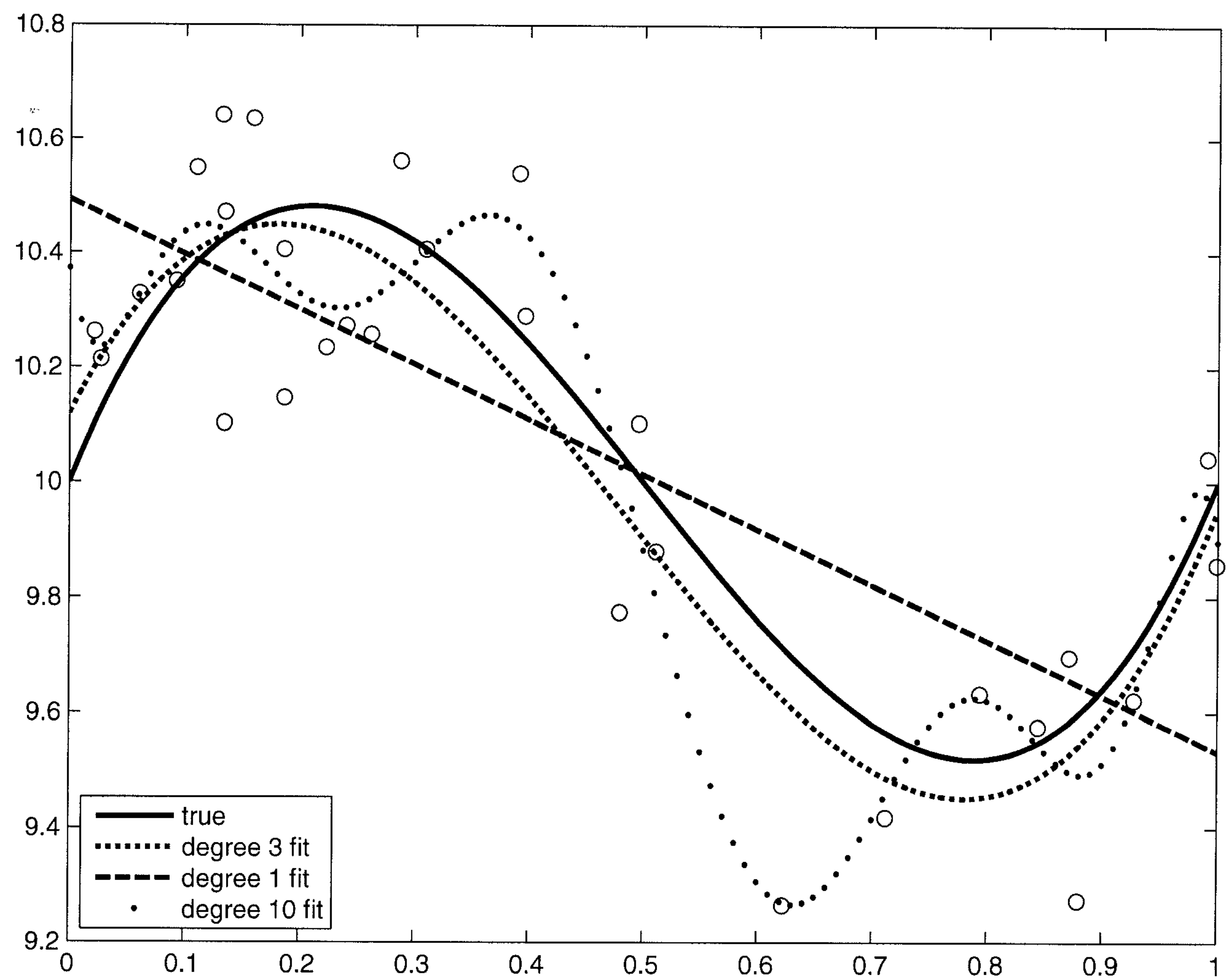
Model Selection

In statistical machine learning, a <u>model</u> is a mathematical representation of a function such a classifier, density, regression function, etc.

Many models involve "free" parameters that are not automatically determined by the learning algorithm. Frequently, the value chosen for such parameters has a significant impact on the performance of the algorithm's output.

Examples

Method	Parameter
• k-NN classification	k = # of neighbors
• kernel density estimation	σ = kernel bandwidth
• decision tree pruning	λ = penalty weight
• polynomial regression	p = polynomial degree
• Gaussian mixture model	K = # of components



For most parameters, the challenge is to strike the right balance between ______________ and ______________. (A)

Error Estimation

A general approach to model selection is the following: Let $\{f_\theta\}$ be a collection of models.

1. Identify a performance measure, or <u>error</u>,

$$E(f_\theta)$$

for assessing the quality of a model.

2. Form an <u>estimate</u> of the error

$$\hat{E}(f_\theta)$$

for each θ.

3. Select

$$\hat{\theta} =$$

Error Functions

Typically error functions depend on the unknown, underlying probability distribution, which is why they must be estimated.

Example Classification

A model is a *classifier*

$$f : \mathbb{R}^d \to \{1, 2, \ldots, M\}$$

The "error" associated to a classifier is

(B)

$$E(f) :=$$

which is the probability of misclassification.

Another error is the "minmax error,"

$$E(f) =$$

Example | Regression

A model is a function

$$f : \mathbb{R}^d \longrightarrow \mathbb{R}$$

A common error is

$$E(f) =$$

©

called the ______ __________ ________.

An alternative is

$$E(f) =$$

called the ______ __________ ________

Example | Density Estimation

A model is a function

$$f : \mathbb{R}^d \to \mathbb{R}$$

such that $f \geq 0$, $\int f = 1$.

Suppose f^* is the true density.

A common error is

(D) $$E(f)$$

called the __________ __________ ________

or L^2 distance.

Another is the Kullback-Liebler divergence

$$E(f) =$$

This error is not a proper distance, but it does satisfy

-
-

Errors as Expectations

Conceptually, our methods for error estimation do two things:

1. Express the error in terms of an expected value
2. Estimate the expected value

Examples

Misclassification rate:

$$E(f) = \Pr\{f(X) \neq Y\}$$

E $=$

Minmax error:

$$E(f) = \max_y \Pr\{f(X) \neq y \mid Y = y\}$$

$$=$$

KL Divergence:

$$E(f) = \int f^*(x) \log\left[\frac{f(x)}{f^*(x)}\right] dx$$

$$=$$

Law of Large Numbers

Suppose that $Z_1, \ldots, Z_n$ are independent and identically distributed realizations of the random variable Z. Then

$$\frac{1}{n} \sum_{i=1}^{n} Z_i \longrightarrow \mathbb{E}\{Z\}$$

as $n \to \infty$.

Therefore we can estimate the expectation of a random variable if we have access to a random sample of the variable.

Fortunately, this is the case in machine learning problems.

Training error

For concreteness, consider a classification problem. Suppose we have training data $(x_1, y_1), \ldots, (x_n, y_n)$. Let $\{f_\theta\}$ be a collection of models (classifiers) and we wish to select the one with smallest error.

Then

$$
\begin{aligned}
E(f_\theta) &= Pr(f_\theta(X) \neq Y) \\
&= E\{1_{\{f_\theta(X) = Y\}}\} \\
&=
\end{aligned}
$$

F.

where

$$\approx$$

$$=$$

This quantity is called the ________________.

or ____________________ error.

By the LLN, it is an estimate of the true error. Thus, we can select θ by minimizing the training error with respect to θ. So that's pretty much all there is to say, right?

Recall that f_θ was constructed from $(x_1, y_1), \ldots,$ (x_n, y_n). Therefore the variables

$$\mathbb{1}_{\{f_\theta(x_i) \neq y_i\}}$$

(G) are not ________________.

Minimizing the training error results in ________________, and should not be employed when the parameter θ determines the ____________ of the model.

Example | k-nearest neighbors : minimizing training error will result in $k =$

(recall that the decision boundary gets smoother as k increases, because we vote over a larger set of neighbors)

Example | Consider a kernel density estimate

$$f_\sigma(x) = \frac{1}{n} \sum_{i=1}^{n} k_\sigma(x - x_i), \qquad \sigma > 0$$

The training error estimate of the KL divergence is

$$E(f_\sigma) = D(f_\sigma \,\|\, f^*)$$

$$= \int f^*(x) \log \left[\frac{f^*(x)}{f_\sigma(x)} \right]$$

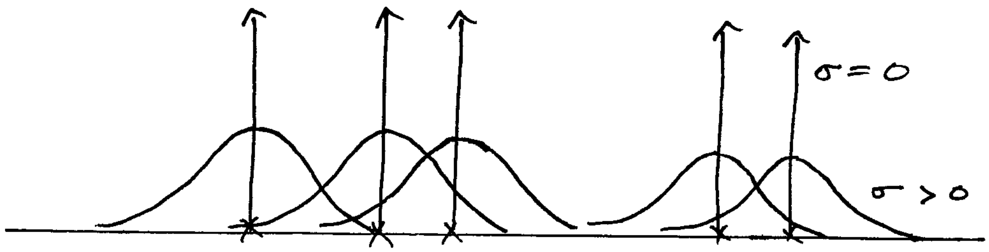
$$= - \int f^*(x) \log f_\sigma(x) \; + \; \text{constant}$$

$$= - \mathbb{E}\{ \log f_\sigma(\mathbf{x}) \}$$

(H) $=$

$\approx$

so the selected σ is ______.



The larger σ, the smoother (less complex) the resulting density estimate.

Holdout Error Estimate

The "holdout" approach to model selection partitions the available data into two sets:

$$(x_1, y_1), \ldots, (x_n, y_n)$$

$$\swarrow \qquad\qquad \searrow$$

$(x_1, y_1), \ldots, (x_m, y_m)$ $\downarrow$ used to construct models

$(x_{m+1}, y_{m+1}), \ldots, (x_n, y_n)$ $\downarrow$ used to estimate error

Example | Consider the polynomial regression problem. Our models are $\{f_d\}$, $d \geq 1$, where

f_d = least squares regression estimate of degree d.

If we use $(x_1, y_1), \ldots, (x_m, y_m)$ to fit the models, then the holdout error estimate is

(I) $$\hat{E}_{HO}(f_d) =$$

If we have lots of data (large n), the holdout estimate can be a good approach. When n is small, however, we'd prefer to use as much of our data as possible for <u>fitting</u> <u>models</u>. This motivates our next strategy.

Cross Validation

Let K be an integer, $1 \le K \le n$.

Assume we have n training points.

Let $I_1, I_2, \dots, I_K$ be a partition of $\{1, 2, \dots, n\}$ such that

J $$|I_k| \approx$$

for each k, $1 \le k \le K$.

<u>Example</u> $n = 10$, $K = 3$

K $I_1 =$

$I_2 =$

$I_3 =$

Let $\{f_\theta\}$ be a model class indexed by θ.

Define

$$f_\theta^{(k)} = \text{model based on } \{(x_i, y_i)\}_{i \notin I_k}$$

and

$$\hat{E}^{(k)}(f_\theta^{(k)}) = \frac{1}{|I_k|} \sum_{i \in I_k} 1_{\{f_\theta^{(k)}(x_i) \neq y_i\}}$$

Then the <u>K-fold cross validation</u> estimate of $E(f_\theta)$ is

$$\hat{E}_{cv}(f_\theta) := \frac{1}{K} \sum_{k=1}^{K} \hat{E}^{(k)}(f_\theta^{(k)})$$

or, alternatively,

L $\quad \hat{E}_{cv}(f_\theta) :=$

(approximate if $|I_k| \neq \frac{n}{K}$ exactly)

Remarks

- Common choices of K are 5, 10, and n. When $K = n$ the estimate is called

(M) ______ ______ ______ cross-validation.

- Since the CV estimate depends on the partition $I_1, \ldots, I_K$, it is common to form several estimates based on several random partitions and average them.
- When using CV for classification, you should ensure that the sets I_k contain training data from each class in the same proportion as the full training sample.

The Bootstrap

Fix $B \geq 1$, an integer. For $b = 1, \ldots, B$, let I_b be a subset of size n obtained by sampling from $\{1, 2, \ldots, n\}$ with replacement.

$\underline{\text{Example}}$ $n = 6$

$$I_1 = \{3, 4, 5, 4, 1, 2\}$$

$$I_2 = \{1, 2, 6, 6, 2, 5\}$$

Again consider a model class $\{f_\theta\}$ indexed by θ.

Define

$$f_\theta^{(b)} = \text{model based on } \{(x_i, y_i)\}_{i \in I_b}$$

$\boxed{\text{bootstrap sample}}$

and

N $\hat{E}^{(b)} =$

Then the <u>bootstrap</u> error estimate is

$$\hat{E}_B(f_\theta) := \frac{1}{B} \sum_{b=1}^{B} \hat{E}^{(b)}(f_\theta^{(b)})$$

Remarks

- Typically B must be large, say $B \approx 200$, for the estimate to be accurate. It can therefore be computationally demanding.

- $\hat{E}_B$ tends to be pessimistic, so it is common to combine the bootstrap and training error estimates. A common choice is

$$\hat{E}_{B,0.632} := 0.632\,\hat{E}_B + 0.368\,\hat{E}_{train}$$

called the "0.632 bootstrap estimate"

- The "balanced" bootstrap chooses $I_1, \dots, I_B$ such that each $i = 1, \dots, n$ appears exactly B times.

- Reference: Efron + Tibshirani, An Introduction to the Bootstrap.

For all methods (holdout, CV, bootstrap), once the tuning parameter(s) have been set, the model is retrained using the full sample.

$\boxed{\text{Key}}$ A. underfitting / overfitting   B. $\hat{\theta} = \arg\min_{\theta} \hat{E}(f_\theta)$

B. $E(f) = \Pr\{f(X) \neq Y\}$, $E(f) = \max_{m=1,\dots,M} \Pr\{f(X) = m \mid Y = m\}$

C. $E(f) = E\{(f(X) - Y)^2\}$, $E(f) = E\{|f(X) - Y|\}$

mean squared error   mean absolute deviation

D. $E(f) = \int (f(x) - f^*(x))^2 dx$, integrated squared error

$E(f) = -\int f^*(x) \log\left[\frac{f(x)}{f^*(x)}\right] dx$, $\begin{cases} E(f) \geq 0 \\ E(f) = 0 \Rightarrow f = f^* \end{cases}$

E. $E\{1_{\{f(X) \neq Y\}}\}$, $\max_m E\{1_{\{f(X) \neq m\}} \mid Y = m\}$

$-E_{f^*}\left[\log\left(\frac{f(X)}{f^*(X)}\right)\right]$

F. $= E\{Z_\theta\}$, $Z_\theta = 1_{\{f_\theta(X) \neq Y\}}$ (Bernoulli)

$\approx \frac{1}{n}\sum_{i=1}^{n} Z_{\theta,i} = \frac{1}{n}\sum_{i=1}^{n} 1_{\{f_\theta(x_i) \neq y_i\}}$,   training / resubstitution

G. independent, overfitting, complexity, $k = 1$

H. $= -E\{\log(\frac{1}{n}\sum_{j=1}^{n} k_\sigma(X - x_j))\} \approx -\frac{1}{n}\sum_{i=1}^{n} \log\left(\frac{1}{n}\sum_{j=1}^{n} k_\sigma(x_i - x_j)\right)$, 0

I. $\hat{E}_{HO}(f_d) = \frac{1}{n-m}\sum_{i=m+1}^{n} (y_i - f_d(x_i))^2$ J. $|I_k| \approx \frac{n}{K}$

K. $I_1 = \{1,3,4,8\}$, $I_2 = \{2,7,9\}$, $I_3 = \{5,6,10\}$

L. $\frac{1}{n}\sum_{k=1}^{K}\sum_{i \in I_k} 1_{\{f_\theta^{(k)}(x_i) \neq y_i\}}$ M. leave-one-out

N. $\hat{E}^{(b)} = \frac{1}{n - |I_b|}\sum_{i \notin I_b} 1_{\{f_A^{(b)}(x_i) \neq y_i\}}$