Model selection

Akito Sakurai

Model selection

- When there are plural of stochastic models that explain a set of data, we want to select one of them, which should be "the best."

- What do you mean by best?
- How to implement it?

- The "best" means to output the prediction for an unseen input that has the smallest error among the models.
- A method: first, estimate "the error for the unseen sample"
  - A method may use samples in
    - Validation dataset, or/and
    - Apply cross validation; and
  - To estimate theoretically

Second, find the minimum

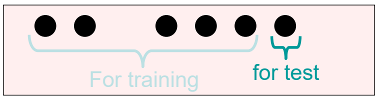
Model selection

- A method
  - Select the model that has the smallest estimate of prediction (generalization) errors; by using unseen samples in:
    - Validation dataset or
    - execute cross validation

- Another method
  - To estimate the generalization error based on some "information criteria"

k-fold cross validation

Divide the training dataset into k groups, train the model with the $(k-1)$ groups and measure the prediction errors on the remaining group (test set) ; and repeat the process k times by changing the test set.



It is not almighty, but works in many cases.

CV measures goodness of algorithms/model architectures

CV is used to determine the best architecture and/or parameters.

Typical information criteria

- AIC

- MDL

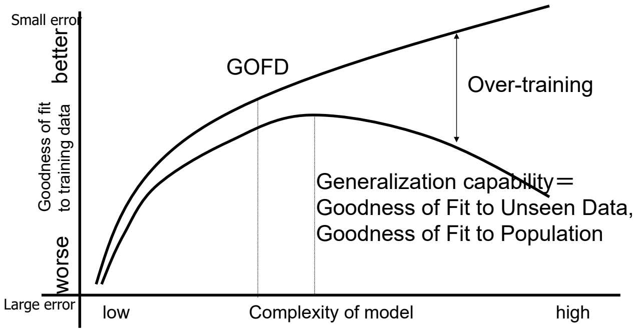
Generalization capability

- Generalization capability is the one to measure how well the learned model works (not for training dataset but) for unseen dataset.
- Training dataset is, in general, deteriorated by errors of labels or output values, which we call noises.
- Therefore, the goodness of fit to data, or GOFD, reflects not only finding regularity but also goodness of fitting to the noise.

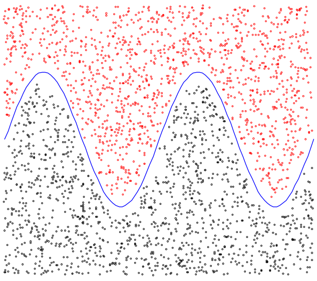
Generalization capability

- GOFD
  = fitness to regularity (generalization capability)
  + fitness to noise, which is divergence from the regularity (over-fitting/over-training)

Behavior in general



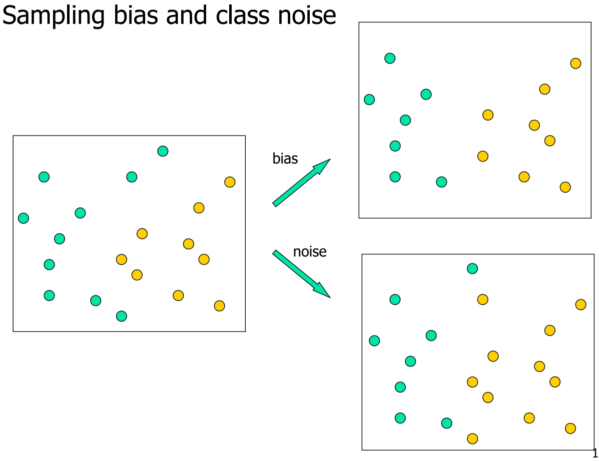
Generalization capability＝
Goodness of Fit to Unseen Data,
Goodness of Fit to Population



“complexity” is not really complexity. It is rather degree of freedom to change a function. Even the shape of $f(x)=0$ is complex, if is has no parameter, it can fit to a very limited set of boundaries for classification.

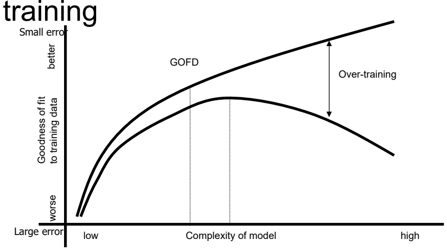
In the figure above, if you have a sin function, supposing that the left most middle point is origin and the white band oscillates between -1 and +1, then the sin implements the classification. If, though, the points move 0.2 along the *x*-axis, the sin function cannot separate the points. But if the function space is of one parameter family as sin($x + p$) where p is a parameter then a function in the space would separate the points.

Sampling bias and class noise

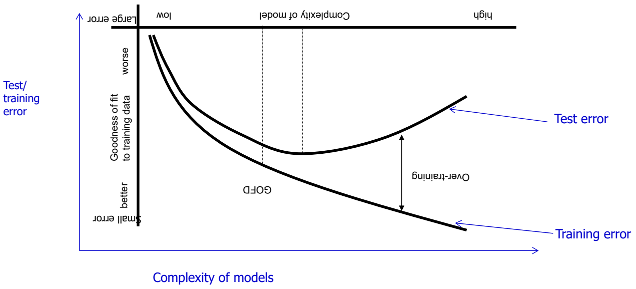


Generalization capability

- The higher the complexity of the model is, the higher the probability of over-training



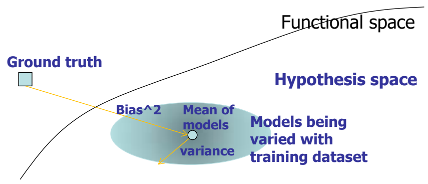
Test/training error



Bias-variance decomposition

- By using a learned model, let us decompose the squared error that is associated with unseen samples into bias^2 + variance which correspond, intuitively, to
  - difference between the "mean of models" and the ground truth,
  - Difference between each model and "the mean of models"

respectively



Bias-variance decomposition

Suppose $Y = f(X) + \varepsilon$ where $\varepsilon \sim N(0, \sigma_\varepsilon^2)$

x_0 is unknown

E is the expectation on ε

Expected loss

$$L(x_0) = E\left[\left(Y - \hat{f}(x_0)\right)^2 \middle| X = x_0\right]$$

$$= E\left[\left(\left(Y - f(X)\right) + \left(f(X) - \hat{f}(x_0)\right)\right)^2 \middle| X = x_0\right]$$

$$= E\left[\left(Y - f(X)\right)^2 \middle| X = x_0\right] + E\left[\left(f(X) - \hat{f}(x_0)\right)^2 \middle| X = x_0\right]$$

$$+ 2E\left[\left(Y - f(X)\right)\left(f(X) - \hat{f}(x_0)\right) \middle| X = x_0\right]$$

$$= \sigma_\varepsilon^2 + E\left[\left(f(x_0) - \hat{f}(x_0)\right)^2\right]$$

$$+ 2\left(f(x_0) - \hat{f}(x_0)\right)E\left[\left(Y - f(X)\right)\middle| X = x_0\right]$$

$$= \sigma_\varepsilon^2 + \left(f(x_0) - \hat{f}(x_0)\right)^2$$

Note: $f(x_0) = E[Y|X = x_0]$

Bias-variance decomposition

Expectation on a training dataset D

$\hat{f}$ depends on D

$$E_D[L(x_0; D)] = \sigma_\varepsilon^2 + E_D\left[\left(f(x_0) - \hat{f}(x_0; D)\right)^2\right]$$

The second term in the right-hand side

$$E_D\left[\left(\left(f(x_0) - E_D[\hat{f}(x_0; D)]\right) + \left(E_D[\hat{f}(x_0; D)] - \hat{f}(x_0; D)\right)\right)^2\right]$$

$$= E_D\left[\left(f(x_0) - E_D[\hat{f}(x_0; D)]\right)^2\right] + E_D\left[\left(E_D[\hat{f}(x_0; D)] - \hat{f}(x_0; D)\right)^2\right]$$

$$+ 2E_D\left[\left(f(x_0) - E_D[\hat{f}(x_0; D)]\right)\left(E_D[\hat{f}(x_0; D)] - \hat{f}(x_0; D)\right)\right]$$

The third term is:

$$2\left(f(x_0) - E_D[\hat{f}(x_0; D)]\right)E_D\left[\left(E_D[\hat{f}(x_0; D)] - \hat{f}(x_0; D)\right)\right] = 0$$

Then $E_D[L(x_0; D)]$ is

$$\sigma_\varepsilon^2 + \left(f(x_0) - E_D[\hat{f}(x_0; D)]\right)^2 + E_D\left[\left(E_D[\hat{f}(x_0; D)] - \hat{f}(x_0; D)\right)^2\right]$$

Bias-variance decomposition

Consequently $E_D[L(x_0; D)]$ is

$$\underbrace{\sigma_\varepsilon^2}_{\text{noise}} + \underbrace{\left(f(x_0) - E_D[\hat{f}(x_0; D)]\right)^2}_{\text{Square of bias}} + \underbrace{E_D\left[\left(E_D[\hat{f}(x_0; D)] - \hat{f}(x_0; D)\right)^2\right]}_{\text{variance}}$$

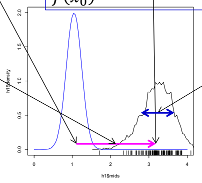
True value of $f(x_0)$

Expectation based on D of prediction of $f(x_0)$

Prediction of $f(x_0)$ which depends on D

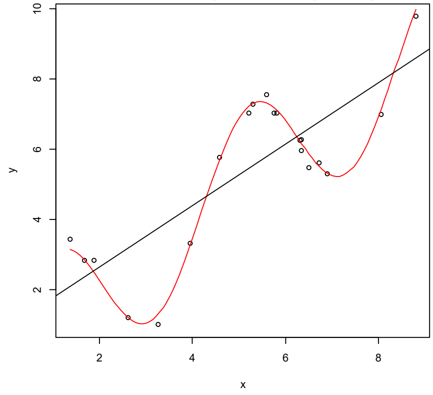
bias

variance

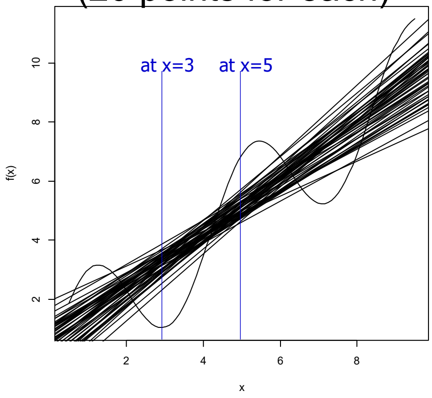


A simple example（20 points）

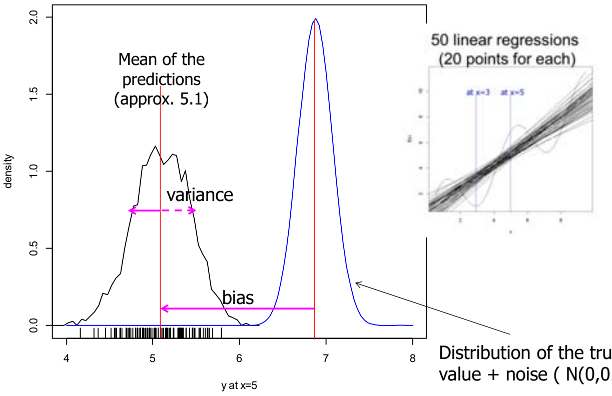
```
y = x + 2*sin(1.5*x)+N(0,0.2)
```



50 linear regressions (20 points for each)



Distribution of predictions at x=5



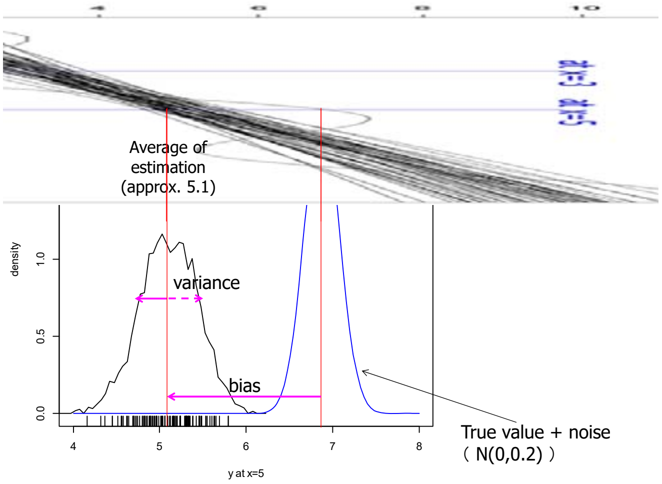
50 linear regressions (20 points for each)

Mean of the predictions (approx. 5.1)

variance

bias

Distribution of the true value + noise (N(0,0.2))

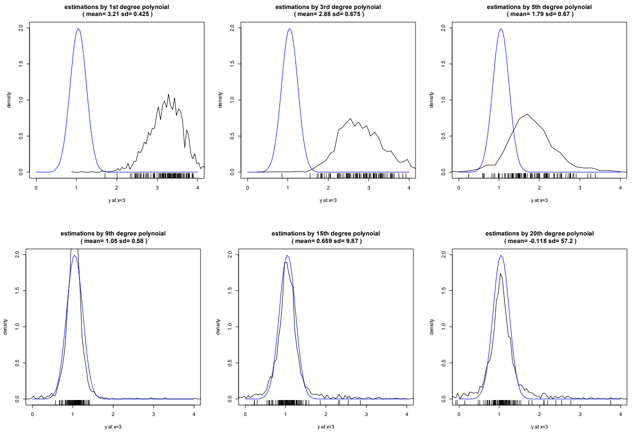


Average of estimation (approx. 5.1)

variance

bias

True value + noise (N(0,0.2))



estimations by 1st degree polynoial (mean= 3.21 sd= 0.425)

estimations by 3rd degree polynoial (mean= 2.88 sd= 0.675)

estimations by 5th degree polynoial (mean= 1.79 sd= 0.67)

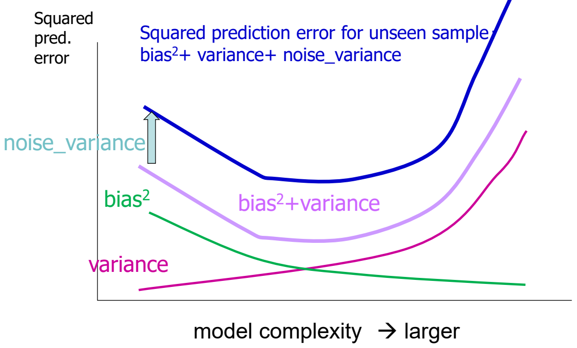
estimations by 9th degree polynoial (mean= 1.05 sd= 0.58)

estimations by 15th degree polynoial (mean= 0.659 sd= 9.87)

estimations by 20th degree polynoial (mean= -0.118 sd= 57.2)

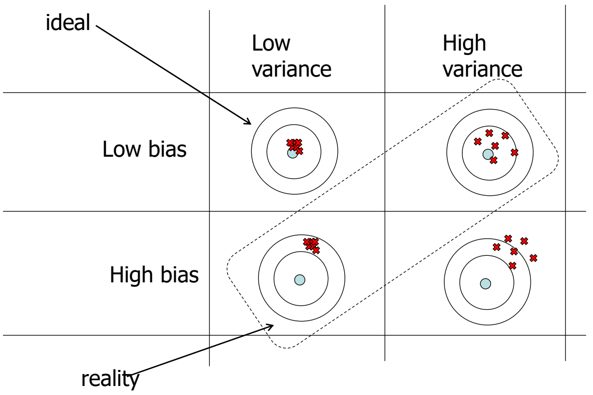
Note: at x=3; even in higher order, outliers exist

Prediction error vs. complexity



Squared prediction error for unseen sample: bias^2+ variance+ noise_variance

bias^2+variance



Generalization capability

- The higher the degree of freedom of models is, the better GOFD is (i.e., better fitting).

Good to some degree; not too good

- The fit to the training dataset is required to be good. But it does not mean that the fit to the test dataset is good enough. In other words, it does not mean that the model discovers a true hidden regularity.
- That is, the goodness (to some degree) of fit is not sufficient, although it is necessary
- Goodness of fit to any test dataset is generalization capability

Model selection

- Generalization capability is the key to best utilization of machine learning
- The essence is:
  - GOFD = fit to regularity (gen. cap.)
    + fit to biases/noises (over-training)
  - Gen. cap. = GOFD – over-training
  - Gen. cap. ≈ GOFD – complexity
  - Therefore, – gen. cap. ≈ – GOFD + complexity

Complexity

- If we could define complexity properly, could we estimate the generalization capability reasonably well?
  - By selecting a model based on the estimated generalization error, we could expect that the model truly minimizing generalization error is obtained (?).

AIC and MDL are typical solutions (there are many others)

AIC

- Akaike Information Criterion (AIC)
  - Akaike himself coined a term “An Information Criterion.” But after some time, the name mentioned has become in common use
- AIC itself represents badness of generalization capability, i.e., the larger AIC is, the worse the model is.

Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. Proc. 2nd International Symposium on Information Theory (B. N. Petrov and F. Csaki eds .) Akademiai Kiado, Budapest, (1973) 267-281.

The first one is: Hirotugu Akaike. Determination of the number of factors by an extended maximum likelihood principle. Research Memorandum 44, Inst. Statist. Math. (March 1971).

AIC

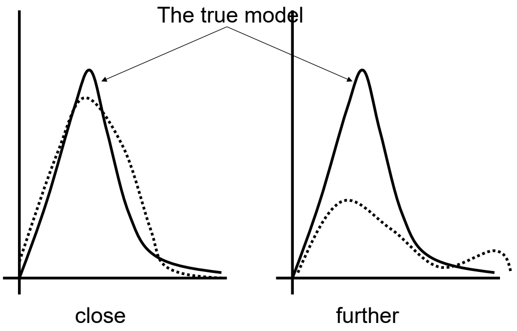
- AIC = -2 log L($\hat{\theta}$ |D) + 2k
  - D: training dataset
  - $\hat{\theta}$: the maximum likelihood estimator (MLE)
  - L : likelihood（L($\hat{\theta}$ |D) = Prob(D| $\hat{\theta}$)）
  - k : the number of parameters that specifies the model
  - log : the natural logarithm

AIC

- measures the model’s complexity by the number of its parameters.
- Does not consider the complexity of functional form（*f*: parameters → probability）

What is complexity of functional form?
In the first place, what is the functional form?
It should be difficult but worth to contemplate.

Distance between distributions.



KL-divergence

- KL-divergence (Kullback-Leibler divergence) is a pseudo distance between two distributions, which is not mathematical distance.
- For two distributions P_i and Q_i where $Q_i \neq 0$

$$\boxed{D(P,Q) = \sum_{i=1}^{k} P_i \log_2 \frac{P_i}{Q_i}}$$

Note: cross entropy

$$H(P,Q) = \sum_{i=1}^{k} P_i \log_2 \frac{1}{Q_i}$$

$$H(P,Q) = H(P) + D(P,Q)$$

- Properties of $D(P,Q)$:

1 $D(P,Q) \geq 0$
2 $D(P,Q) = 0 \text{ iff } P = Q$

- Not symmetric. Triangle inequality does not hold.

MDL

- These are coded by a programming language as follows:
  - 0001000100010001000100010001
    - 7.times{ print "0001" }
  - 0111010011010000101010101011
    - puts("0111010011010000101010101011")

MDL

- Regularity is vital to compress data.
- In general, the more regular the data is, the shorter the program is, although the real length of the compressed data depends on coding method.
  - Selection of coding method is a minor problem in theory, because the term relating to the coding method is upper bounded by a constant.

MDL

- Suppose a program is a model.
- In general, the program that grasps regularity in data most is a shortest program, i.e., a shortest code.
  - 0001000100010001000100010001
    - 7.times{ print "0001" }
  - 0111010011010000101010101011
    - puts("0111010011010000101010101011")

more regular
more random

Regularity or randomness of a sequence is defined.
Standard definition of randomness is for data source not for a sequence generated.

MDL

- If a model captures regularity in data, the model can predict the next data to come more correctly. That is, it shows better generalization capability.
- In other words, a model with the shortest length is the model whose prediction capability is the highest.

```
0001000100010001000100010001
7.times{ print "0001" }
puts("0001000100010001000100010001")
```

Occam’s razor

- What is known most is:
  - Entities should not be multiplied beyond necessity.
- According to Bertrand Russell
  - It is vain to do with more what can be done with fewer.
- Most common interpretation:
  - Among the theories that are consistent with the observed phenomena, one should select the simplest theory.

MDL and Occam's razor

- Occam's razor: "Choose the simplest"

length of hypothesis

length of residuals (errors)

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D \mid h)$$

ex. bit length to describe h.

$\propto$ Bit length of corresponding codes

h being given, bit length to describe D

$\propto$ The number of misclassified data

This is not practical. Feasible formulations are:
1. Stochastic MDL by Rissanen
2. MDL based on program complexity by Kolmogorov/Chaitin and a group of Lin & Vitanyi

Code theoretic interpretation

- MDL: Select a hypothesis that minimize:

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(D \mid h) P(h) \\ &= \arg\min_{h \in H} -\log_2 P(D \mid h) - \log_2 P(h) \\ &= \arg\min_{h \in H} L_{C_2}(D \mid h) + L_{C_1}(h) \end{aligned}$$

length of code for conditional probability

length of code for hypothesis

Note: probability and code length

- Suppose a set X is finite or countable
  - A code $C(x)$ of X is:
    - A 1-to-1 mapping from X to $U_{n>0}\{0,1\}^n$
    - $L_C(x)$: code length in bits when a code system C is used.
  - P: a probability distribution defined on X.
    - $P(x)$: the probability of x
    - An observed sequence (iid) $x_1, x_2, \ldots, x_n$: x^n

$$P(x^n) = \prod_{i=1}^{n} P(x_i)$$

Stochastic MDL

- Under stochastic framework, i.e., when data distributes, MDL principle is:

$$MDL = \underline{-\ln f(x \mid \hat{\theta})} + \underline{\frac{k}{2} \ln \frac{N}{2\pi}} + \underline{\ln \int \sqrt{\det I(\theta)}\, d\theta}$$

Badness of fit (error)

Penalty for the number of parameters

Penalty related to form of probability distribution

J.Rissanen, Modeling by shortest data description. *Automatica,* vol. 14 (1978), pp. 465-471.
J.Rissanen, Fisher information and stochastic complexity. *IEEE Trans. Information Theory,* vol. 42 (1996), pp. 40-47.

MDL Reading http://www.mdl-research.org/reading.html

Comparison of AIC and MDL

- Let us compare AIC and MDL:

$$AIC = -2\sum_{i=1}^{N} \log p(X_i; \hat{\theta}) + 2k$$

$$2MDL = -2\sum_{i=1}^{N} \log p(X_i; \hat{\theta}) + k \log N$$

- The second terms say: When N is large, MDL is larger than AIC. This is why MDL prefers a model with smaller number of parameters.

This is clearly visible when Bayesian network is learnt.
Try it with Weka.