Computational Intelligence: Methods and Applications

Lecture 14
Bias-variance tradeoff – model selection.

Włodzisław Duch
SCE, NTU, Singapore
Google: Duch

Linear model

In the simplest case (2 classes, diagonal covariance matrices) linear model has been derived for discrimination function:

$$g_i(\mathbf{X}) = \sum_{j=1} W_{ij}^T X_j + W_{i0}$$

This model has *d*+1 parameters **W** for each pair of classes;

these parameters are estimated using Bayesian approach from means, covariance matrix and priors:

$$\mathbf{W}_i = \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1}$$

$$W_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_i + \ln P(\omega_i)$$

Estimations in high-dimensional small sample case are inaccurate.

Direct estimation

Simple approach: given a linear model:

$$g(\mathbf{X};\mathbf{W}) = \mathbf{W}^{\mathrm{T}}\mathbf{X}; \quad \mathbf{W}^{\mathrm{T}} = [W_0, W_1, \ldots W_d]$$

$$\mathbf{X}^{\mathrm{T}} = [1, X_1, \ldots X_d]$$

calculate how many errors *E*(W) the decision procedure Ĉ makes

g(X;W)>0 then class 1, otherwise class 2;

$$E(\mathbf{W}) = \sum_{\mathbf{X}} \left(\hat{C}(\mathbf{X};\mathbf{W}) - C(\mathbf{X})\right)^2$$

Here C(X) is the desired answer, frequently ±1, or a binary class indicator (0,0,..1,0,0), and decision procedure has the same form.

Note: the Mean Square Error (MSE) function may be replaced by any other formula with non-negative contributions.

2 Gaussians LDA

A generative model: mix number of Gaussians, pick up randomly one Gaussian and select randomly one point from Gaussian distribution.

Use one Gaussian per class, diagonal covariance matrix

Minimize E(W) directly using generated data – this will generate least square solution.

Linear Discrimination Analysis methods will be considered in some details later.

In this case results may be optimal in the Bayesian sense, errors are made due to the real overlap of two classes.

Demo: GM for g1000.dat

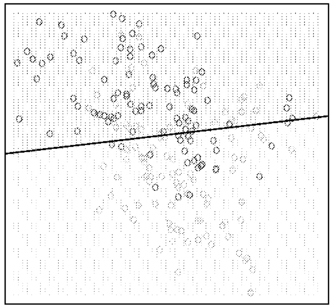


Fig. 2.1, Hasti, Tibshirani, Friedman

2 Gaussians 1-NN

Consider another extreme model case: distance from the mean

$$D(\mathbf{X},\boldsymbol{\mu}_i) = (\mathbf{X}-\boldsymbol{\mu}_i)^T(\mathbf{X}-\boldsymbol{\mu}_i)$$

Avoid calculating the mean, adopt simpler decision rule:

find nearest reference vector $X^{(i)}$

min $D(X,X^{(i)})$

assign $C(X) = C(X^{(i)})$

Since all training vectors are used the number of errors for the training data is 0, unless the test vector X is excluded.

Is it really the best solution?

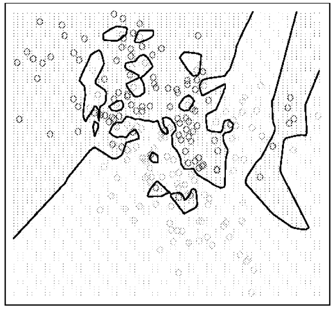


Fig. 2.3, Hasti, Tibshirani, Friedman

Bias and variance of the model

Sometimes a model is too simple (linear model in realistic cases), for approximation of complex functions and decision borders models should be sufficiently flexible to avoid **data underfitting**.

Models that learn perfectly the available data (like the nearest neighbor model) are just lookup tables! They overfit the data.
They will not work well on new data – will generalize poorly.

Models should not be too flexible, use some assumptions about data.

3 sources of error:

- Bias of the model: too simple or too complex?
- Variance of the model: how will accuracy differ from average if the learning process is repeated many times? It depends on the training procedure.
- Noise in the data: usually difficult to estimate.

Noise: observations do not reflect true probability distributions.

The goal of learning

Given:

- a training data set $D=(X^{(i)}, Y^{(i)})$, $i=1 .. n$,
- a model M that learns to make predictions $Y_M=M(X;\mathbf{W})$ using a set of parameters and decision procedures $\mathbf{W}$.
- a cost, error, risk or loss function $L(M(X;\mathbf{W}),Y)=L(Y_M,Y)$, ex:
  - a squared function $L(Y_M,Y) = (Y_M-Y)^2$
  - linear abs function $L(Y_M,Y) = |Y_M-Y|$
  - a zero-one loss $L(Y_M,Y) = 1-\delta(Y_M,Y)$, or a risk $\lambda(Y_M,Y)$

The goal of learning from data is to create an optimal model, or minimize the average $L(Y_M,Y)$ over all examples in the dataset.

Targets Y may not be quite deterministic, since the process $f(X)$ generating data may include some noise ε, so $Y=f(X)+\varepsilon$, and $E(\varepsilon)=0$

Then different targets are produced for the same feature vectors X.

Optimal prediction

- Optimal prediction for sample X associated with targets Y is:

$$Y_M^* = \arg\min_{Y_M} E_Y\left[L(Y_M,Y)\right] = \arg\min_{Y_M}\int L(Y_M,Y)P(Y)dY$$

The expectation value of the loss function calculates an average loss for a given prediction Y_M, and optimal prediction minimizes this loss.

Optimal prediction should approximate true process $f(X)$, not Y.

The most common bias-variance decomposition is derived for quadratic loss functions although quite general derivations exist.

$$MSE = E_D\left[L(Y_M,Y)\right] = \frac{1}{n}\sum_{i=1}^{n}\left(Y_M^{(i)} - Y^{(i)}\right)^2$$

What is the expected value of the *MSE* error?

Expected MSE

Imagine that we have infinite number of samples from the $f(\mathrm{X})$ function, and $Y^{(i)}=f(\mathrm{X}^{(i)})+\varepsilon$, calculate the expectation value for the MSE error:

$$E[MSE]=\frac{1}{n}\sum_{i=1}^{n}E\left[\left(Y_{M}^{(i)}-Y^{(i)}\right)^{2}\right]$$

Using the true value $f^{(i)}=f(\mathrm{X}^{(i)})$ to calculate this expectation

$$E\left[\left(Y_{M}^{(i)}-Y^{(i)}\right)^{2}\right]=E\left[\left(Y_{M}^{(i)}-f^{(i)}+f^{(i)}-Y^{(i)}\right)^{2}\right]$$

$$=E\left[\left(Y_{M}^{(i)}-f^{(i)}\right)^{2}\right]+E\left[\left(f^{(i)}-Y^{(i)}\right)^{2}\right]+2E\left[\left(Y_{M}^{(i)}-f^{(i)}\right)\left(f^{(i)}-Y^{(i)}\right)\right]$$

$$=E\left[\left(f^{(i)}-Y_{M}^{(i)}\right)^{2}\right]+E\left[\varepsilon^{2}\right]$$

$$+2\left(E\left[Y_{M}^{(i)}f^{(i)}\right]-E\left[Y_{M}^{(i)}Y^{(i)}\right]-E\left[f^{(i)2}\right]+E\left[f^{(i)}Y^{(i)}\right]\right)$$

More MSE expectations

$f^{(i)}$ is deterministic and thus its expectation is equal to function

$$E\left[f^{(i)}Y^{(i)}\right]=f^{(i)}E\left[Y^{(i)}\right]=f^{(i)2}$$

assuming that noise has zero variance.

Noise is independent of the model predictions, therefore

$$E\left[Y_{M}^{(i)}Y^{(i)}\right]=E\left[Y_{M}^{(i)}\left(f^{(i)}+\varepsilon\right)\right]=E\left[Y_{M}^{(i)}f^{(i)}\right]+E\left[Y_{M}^{(i)}\right]E[\varepsilon]$$

$$=E\left[Y_{M}^{(i)}f^{(i)}\right]$$

Final result: MSE for true function/predicted values + variance of the noise:

$$E\left[\left(Y_{M}^{(i)}-Y^{(i)}\right)^{2}\right]=E\left[\left(f^{(i)}-Y_{M}^{(i)}\right)^{2}\right]+E\left[\varepsilon^{2}\right]$$

MSE decomposition

Using again the expansion trick the first term is:

$$E\left[\left(f^{(i)}-Y_{M}^{(i)}\right)^{2}\right]=E\left[\left(f^{(i)}-E\left[Y_{M}^{(i)}\right]+E\left[Y_{M}^{(i)}\right]-Y_{M}^{(i)}\right)^{2}\right]=$$

$$E\left[\left(f^{(i)}-E\left[Y_{M}^{(i)}\right]\right)^{2}\right]+E\left[\left(E\left[Y_{M}^{(i)}\right]-Y_{M}^{(i)}\right)^{2}\right]+$$

Bias + variance

$$2E\left[\left(f^{(i)}-E\left[Y_{M}^{(i)}\right]\right)\left(E\left[Y_{M}^{(i)}\right]-Y_{M}^{(i)}\right)\right]$$

+ 0

The first term $E\left[\left(f^{(i)}-E\left[Y_{M}^{(i)}\right]\right)^{2}\right]$ is a bias of the model, expected loss due to the ability of the model to approximate true function.

the second term $E\left[\left(E\left[Y_{M}^{(i)}\right]-Y_{M}^{(i)}\right)^{2}\right]$ is the variance of the model;

The third term is zero, as one may prove without trouble.

Final decomposition

Third term:

$$E\left[\left(f^{(i)}-E\left[Y_{M}^{(i)}\right]\right)\left(E\left[Y_{M}^{(i)}\right]-Y_{M}^{(i)}\right)\right]=$$

$$E\left[f^{(i)}E\left[Y_{M}^{(i)}\right]\right]-E\left[f^{(i)}Y_{M}^{(i)}\right]-E\left[E\left[Y_{M}^{(i)}\right]^{2}\right]+E\left[E\left[Y_{M}^{(i)}\right]Y_{M}^{(i)}\right]=$$

$$f^{(i)}E\left[Y_{M}^{(i)}\right]-f^{(i)}E\left[Y_{M}^{(i)}\right]-E\left[Y_{M}^{(i)}\right]^{2}+E\left[Y_{M}^{(i)}\right]^{2}=0$$

Final decomposition is:

$$E[MSE]=Var(\varepsilon)+\mathrm{Bias}^{2}\left(f,Y_{M}\right)+Var\left(Y_{M}\right)$$

Variance of noise is fixed, but bias and variance of the model predictions depends on the selection of model, decision procedures and model parameters.

Bias – variance tradeoff

Both bias and the variance of the model should be minimized

$$\text{Bias}^2\left(f,Y_M\right)=E\left[\left(f^{(i)}-E\left[Y_M^{(i)}\right]\right)^2\right];$$

$$Var\left(Y_M\right)=E\left[\left(Y_M^{(i)}-E\left[Y_M^{(i)}\right]\right)^2\right]$$

Large bias: too simple model, but then variance is small.

Example: constant predictions: zero variance, high bias.

Small bias: very flexible model, following the training points almost exactly (because $E(Y)=f$), but then variance may be large, following noise in the data, and thus generalization may be poor.

Example: functions fitting, showing bad effects of low bias on variance.

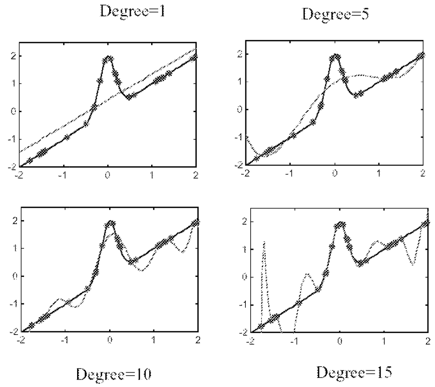
Function fitting

Red dots: experimental data,

blue line – generating function, noiseless case

Large bias: too simple model, low-degree of polynomials, small variance.

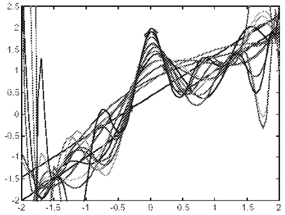
Small bias: high polynomial degrees, but then large variance.



Test and training errors

Try many different models (polynomials):

some curves may pass directly through the training points.



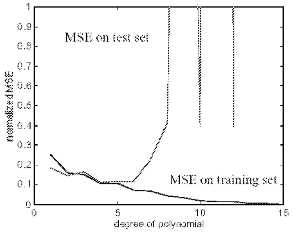
With the increasing model complexity:

bias decreases

MSE on the training set decreases

variance increases

MSE on the test set increases.



Test and training errors

Relation between model complexity and prediction error.

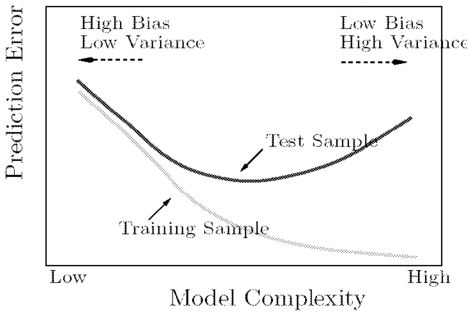


Fig. 2.11, from Hasti, Tibshirani, Friedman

See more on model selection in: chap. 9.3 in Duda et al, or Webb, chap. 11.1, or Hasti et al, chap. 2.9