Assessment and Model selection

Ken Wang

Today's topic about "MODEL SELECTION" is really interesting and important topic. Model selection is a key conceptual for performing dimension reduction and exploiting hidden structures in the data. The general idea is to compare different statistical models corresponding to different possible hidden structures and then select among them the one that is more suited for estimation. Model selection is a very powerful theory, but it suffers in many cases from a very high computational complexity that can be prohibitive.

Q1. Up today, as your knowledge, Why we need to perform feature selection?
Q2. What's the different between model selection and feature selection?

1. Fitting and Bias-Variance trade-off

As we have seen in earlier chapters, the phenomenon of over-fitting is really an unfortunate property of maximum likelihood. We can use the perspective of frequencist to think the complexity of the model. Which we call it bias-variance trade-off

when we discussed decision theory for regression problems, we considered various loss functions each of which leads to a corresponding optimal prediction once we are given the conditional distribution $p(y|\mathbf{x})$. A popular choice is the squared loss function, for which the optimal prediction is given by the conditional expectation, which we denote by $h(\mathbf{x})$ and which is given by

$$h\left(x\right)=E[y|x]=\int yp\left(y|x\right)dy$$

The point is to determine the conditional distribution , we have been shown the square loss expectation can be written as

$$E[L]=\int\left\{f(x)-h(x)\right\}^{2}p(x)dx+\iint\left\{h(x)-y\right\}^{2}p\left(x,y\right)dxdy$$

The second term is the variance of the data, it represent the minimum value of expectation loss function. Because the first term is non-negative, we need to find a f to minimize it to zero, considering we have infinite dataset, If we had an unlimited supply of data (and unlimited computational resources), we could in principle find the regression function h(x) to any desired degree of accuracy, and this would represent the optimal choice for y(x). However, in practice we have a data set D containing only a finite number N of data points, and consequently we do not know the regression function h(x) exactly.

频率学家的方法涉及到根据数据 D 对 w 进行点估计，然后试着通过下面的思想实验来表示估计的不确定性。假设我们有许多数据集，每个数据集的大小为 N，并且每个数据集都独立地从分布 p(y; x)中抽取。对于任意给定的数据集 D，我们可以运行我们的学习算法，得到一个预测函数 y(x; D)。不同的数据集会给出不同的函数，从而给出不同的平方损失的值。这样，特定的学习算法的表现就可以通

过取各个数据集上的表现的平均值来进行评估。

$$\{f(x;D)-h(x)\}^2=\{y(x;D)-E_D[y(x;D)]+E_D[y(x;D)]-h(x)\}^2$$

So, the expectation can be written as

$$E_D[\{f(x;D)-h(x)\}^2]=\{E_D[f(x;D)]-h(x)\}^2+E_D[\{f(x;D)-E_D[f(x;D)]\}^2]$$

The first term is called the bias, represents the extent to which the average prediction over all data sets differs from the desired regression function. the second term is called variance, measures the extent to which the solutions for individual data sets vary around their average, and hence this measures the extent to which the function y(x; D) is sensitive to the particular choice of data set.

So far, we obtain the following decomposition of the expected squared loss

Expected loss = (bias)^2 + variance + noise

Our target is to minimize the expected loss, which we have decomposed into the sum of a (squared) bias, a variance, and a constant noise term. As we shall see, there is a trade-off between bias and variance, with very flexible models having low bias and high variance, and relatively rigid models having high bias and low variance. The model with the optimal predictive capability is the one that leads to the best balance between bias and variance.

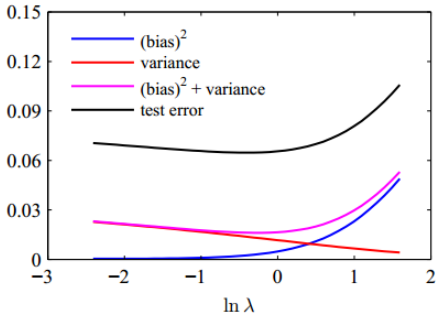
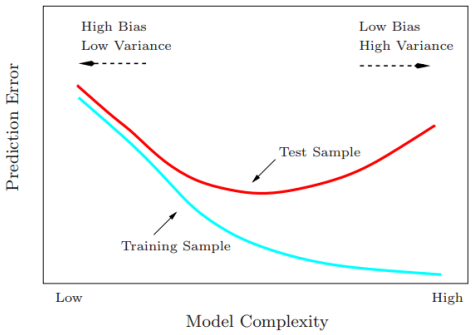


Figure1. the illustration of bias-variance trade-off

Example. Overfitting and underfitting

More generally, as the model complexity of our procedure is increased, the variance tends to increase and the squared bias tends to decrease. The opposite behavior occurs as the model complexity is decreased. Typically we would like to choose our model complexity to trade bias off with variance in such a way as to minimize the test error. An obvious estimate of test error is the training error. Unfortunately training error is not a good estimate of test error, as it does not properly account for model complexity.



2. Cross validation

We have already seen that, in the maximum likelihood approach, the performance on the training set is not a good indicator of predictive performance on unseen data due to the problem of over-fitting. If data is plentiful, then one approach is simply to use some of the available data to train a range of models, or a given model with a range of values for its complexity parameters, and then to compare them on independent data, sometimes called a **validation set,** and select the one having the best predictive performance. If the model design is iterated many times using a limited size data set, then some over-fitting to the validation data can occur and so it may be necessary to keep aside a third **test set** on which the performance of the selected model is finally evaluated.

n many applications, however, the supply of data for training and testing will be limited, and in order to build good models, we wish to use as much of the available data as possible for training. However, if the validation set is small, it will give a relatively noisy estimate of predictive performance. One solution to this dilemma is to use **cross-validation.** This allows a proportion (*S* -1)*/S* of the available data to be used for training while making use of all of the data to assess performance. When data is particularly scarce, it may be appropriate to consider the case *S* = *N*, where *N* is the total number of data points, which gives the *leave-one-out* technique.

Recommend Reading: Teschendorff A E[1]

3. AIC

Let us compute the risk $r_m = R(\hat{f}_m)$of the estimator, Starting from $Y = f^* + \varepsilon$, f* represent h(x) as mentioned. We consider a collection of $\{S_m , m \in M\}$ of linear subspaces of R^n, called model.

Associate to each subspace S_m the constrained maximum likelihood estimators $\hat{f}_m = Proj_{S_m} Y$.

So, we obtain the decomposition,

$$f^* - \hat{f}_m = (I - Proj_{S_m}) f^* - Proj_{S_m} \varepsilon$$

Further we have,

$$r_m = E_M \left[\left\| f^* - \hat{f}_m \right\|^2 \right] = \|(I - Proj_{Sm}) f^*\|^2 + d_m \sigma^2$$

Where dm = dim(Sm), the risk r_m involves two terms. The first term is a bias term that reflects the quality of S_m for approximating f*, the second term is a variance term that increase linearly with the dimension of Sm. The oracle model S_{m0} is then the model in the collection with achieves the best trade-off between the bias and the variance.

A natural idea is to use an unbiased estimator of the risk r_m. It follows from the decomposition

$$Y - \hat{f}_m = (I - Proj_{Sm})(f^* + \varepsilon)$$

So, we further have

$$E\left[\left\|Y-\hat{f}_m\right\|^2\right]=\|(I-Proj_{S_m})f^*\|^2+(n-d_m)\,\sigma^2=r_m+(n-2d_m)\sigma^2$$

As a consequence,

$$\hat{r}_m=\left\|Y-\hat{f}_m\right\|^2+(2d_m-n)\sigma^2$$

Is an unbiased estimator of the risk. Note that the last term does not change the choice of m, This choice gives the Akaike Information Criterion(AIC).

$$\hat{m}_{AIC}\in \underset{m\in M}{argmin}\left\{\left\|Y-\hat{f}_m\right\|^2+2d_m\sigma^2\right\}$$

This criterion is very natural and popular. Nevertheless it can produce very poor results in some cases because it does not take into account the variability of the estimated risks around their mean. the case when the number of models S_m with dimension d grows exponentially with d. due to the fact that in this setting, for large dimensions d, we have a huge number of models S_m with dimension d. Therefore, we have a huge number of estimators r^*_m, and due to the randomness, some of them deviate seriously from their expected value r_m. In particular, some r^*_m are very small, much smaller than $r^*_{m_o}$ associated to the oracle m_o. This leads the AIC criterion to select a model S_{m^*} much bigger than S_{m_o} with very high probability.

About the estimators of r_m , there also has differen version, like BIC and so on.

4. Computational Issues

Unfortunately, calculate each model risk cannot be implemented in practice due to its prohibitive computational complexity.

number of variables (or groups)	p	10	20	40	80	270
cardinality of $\mathcal{M}$	2^p	1 024	$1.1\,10^6$	$1.1\,10^{12}$	$1.2\,10^{24}$	number of particles in the universe

But, it still can implement on principle component regression.

Recommend Reading: Mevik B H[2,3]

The principe of forward–backward minimization is to approximately minimize Criterion by alternatively trying to add and remove variables from the model. More precisely, it starts from the null model and then builds a sequence of models $(m_t)_{t\in\mathbb{N}}$, where two consecutive models differ by only one variable. At each step t, the algorithm alternatively tries to add or remove a variable to m_t in order to decrease Criterion When the criterion cannot be improved by the addition or deletion of one variable, the algorithm stops and returns the current value. Below crit(m) refers to Criterion.

Forward–backward algorithm

Initialization: start from $m_0 = \emptyset$ and $t = 0$.

Iterate: until convergence

- **forward step:**
  - search $j_{t+1} \in \operatorname{argmin}_{j \notin m_t} \operatorname{crit}(m_t \cup \{j\})$
  - if $\operatorname{crit}(m_t \cup \{j_{t+1}\}) \leq \operatorname{crit}(m_t)$ then $m_{t+1} = m_t \cup \{j_{t+1}\}$ else $m_{t+1} = m_t$
  - increase t of one unit
- **backward step:**
  - search $j_{t+1} \in \operatorname{argmin}_{j \in m_t} \operatorname{crit}(m_t \setminus \{j\})$
  - if $\operatorname{crit}(m_t \setminus \{j_{t+1}\}) < \operatorname{crit}(m_t)$ then $m_{t+1} = m_t \setminus \{j_{t+1}\}$ else $m_{t+1} = m_t$
  - increase t of one unit

Output: $\widehat{f}_{m_t}$

In practice, the forward–backward algorithm usually converges quickly. So the final estimator can be computed efficiently. Yet, we emphasize that there is no guarantee at all, that the final estimator has some good statistical properties. See reference Zhang's[4] work to derive a variant of the forward-backward algorithm.

Recommend Reading

[1] Teschendorff A E. Avoiding common pitfalls in machine learning omic data science[J]. Nature materials, 2018: 1

[2] Mevik B H, Cederkvist H R. Mean squared error of prediction (MSEP) estimates for principal component regression (PCR) and partial least squares regression (PLSR)[J]. Journal of Chemometrics, 2004, 18(9): 422-429

[3] Wehrens R, Mevik B H. The pls package: principal component and partial least squares regression in R[J]. 2007.

[4] T. Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. IEEE Trans. Inform. Theory, 57(7):4689–4708, 2011