Info 411
Machine Learning and Data Mining
Lecture 9: Regression and Model Selection

Brendon J. Woodford

12-Sep, 2017

Department of
Information Science

Lecture Outline

- Introduction
- Regression
- Bias-Variance Dilemma
- Model Selection

Introduction

- We have studied quite a few classifiers:
  - k-NN, Bayes, SVM, Decision Tree, LVQ, RBF, LDA, ...
- Questions remain:
  - How to assess their ***performance*** and make a choice?
  - There are many possible parameter settings too...
  - From another perspective, data samples are usually limited (and expensive to collect). How do we make good use of the limited resource for both training and model validation?
- The same issue applies to clustering and ***regression***

Regression

- Example applications:
  - Prediction of time series (e.g. stock indicies, telco traffic, power consumption, ...)
  - Line / curve fitting of a function

Regression (continued)

- Assume data is $r = f(x) + \epsilon$ where:
  - ϵ is zero mean Gaussian with constant variance σ
- Function $f(x)$ however is ***unknown***
- We have observations $\{r_t\}$ over $X = \{x_t\}$, $t = 1, \cdots, N$
- We can construct an estimator $g(x|\theta)$, θ being a parameter set
- If we assume the error $\epsilon \sim \mathcal{N}(0, \sigma^2)$
- The p.d.f $p(r|x) \sim \mathcal{N}(g(x|\theta), \sigma^2)$
- Then we can use maximum likelihood to learn θ:

$$\begin{aligned}\mathcal{L}(\theta|X) &= \log \prod_{t=1}^{N} p(x_t, r_t) \\ &= \log \prod_{t=1}^{N} p(r_t|x_t) + \log \prod_{t=1}^{N} p(x_t)\end{aligned}$$

From LogL to Error

- Using $p(r|x) \sim \mathcal{N}(g(x|\theta), \sigma^2)$, we have

$$\begin{aligned}\mathcal{L}(\theta|X) &= \log \prod_{t=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{[r_t - g(x_t|\theta)]^2}{2\sigma^2}\right) \\ &= -N \log \sqrt{2\pi}\sigma - \frac{1}{2\sigma^2} \sum_{t=1}^{N} [r_t - g(x_t|\theta)]^2\end{aligned}$$

- Hence, maximising $\mathcal{L}(\theta|X)$ is equivalent to minimising the error:

$$E(\theta|X) = \frac{1}{2} \sum_{t=1}^{N} [r_t - g(x_t|\theta)]^2$$

Regression Models

- There are a number of ways to construct the estimator $g(x|\theta)$ and estimate θ
- From simple to more complex models: linear regression, polynomial regression, kernel methods, ...
- Question: How do we know what level of complexity is the right one for our given data?

Linear Regression

- Assume $g(x^t|w_1, w_0) = w_1 x^t + w_0$.
- Minimising $E(\theta|X)$ leads to

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

and

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

- The above two equations can be expressed in a matrix form: $\mathbf{y} = \mathbf{A}\mathbf{w}$, where

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

- And $\mathbf{w}$ can be worked out as $\mathbf{w} = \mathbf{A}^{-1}\mathbf{y}$.

Polynomial Regression

- Or, we can employ polynomials for the estimator:

$$g(x_t|w_k,\cdots,w_2,w_1,w_0) = w_k(x^t)^k + \cdots + w_2(x^t)^2 + w_1x^t + w_0$$

- Let

$$\mathbf{w} = \begin{pmatrix} w_0, \\ w_1, \\ w_2, \\ \vdots \\ w_k \end{pmatrix}, \mathbf{D} = \begin{pmatrix} 1 & x^1 & (x^1)^2 & \cdots & (x_1)^k \\ 1 & x^2 & (x^2)^2 & \cdots & (x^2)^k \\ \vdots & & & \vdots & \\ 1 & x^N & (x^N)^2 & \cdots & (x^N)^k \end{pmatrix}, \mathbf{r} = \begin{pmatrix} r^1, \\ r^2, \\ \vdots \\ r^N \end{pmatrix},$$

then we have $\mathbf{r} = \mathbf{D}\mathbf{w}$, and $\mathbf{w}$ can be solved by getting the pseudo-inverse of $\mathbf{D}$, i.e.,

$$\mathbf{w} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\mathbf{r}$$

Bias and Variance

- Given a sample of X and r, we construct the estimate $g(.)$. The estimated squared error can be split into two parts:

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

- Ignore the noise, and examine the average error across all samples:

$$\begin{aligned} E_X[(E[r|x]-g(x))^2|x] = & \quad (E[r|x]-E_X[g(x)])^2 && \textit{(bias)} \\ & + E_X[(g(x)-E_X[g(x)])^2] && \textit{(variance)} \end{aligned}$$

- Challenge: both need to be kept small –
  - Bias: how good $g(x)$ is on average regardless of samples
  - Variance: how much fluctuation $g(x)$ fluctuates with each sample, around the expected value $E[g(x)]$

Estimating Bias and Variance

Given M samples $X_i = \{x_i^t, r_i^t\}$, $i = 1,\dots,M$, each used to fit $g_i(x)$,

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

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

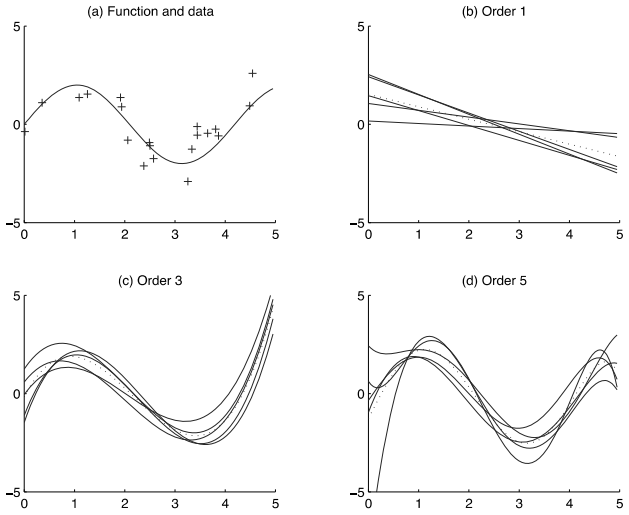
with

$$\bar{g}(x) = \frac{1}{M}\sum_i g_i(x)$$

Bias/Variance Dilemma

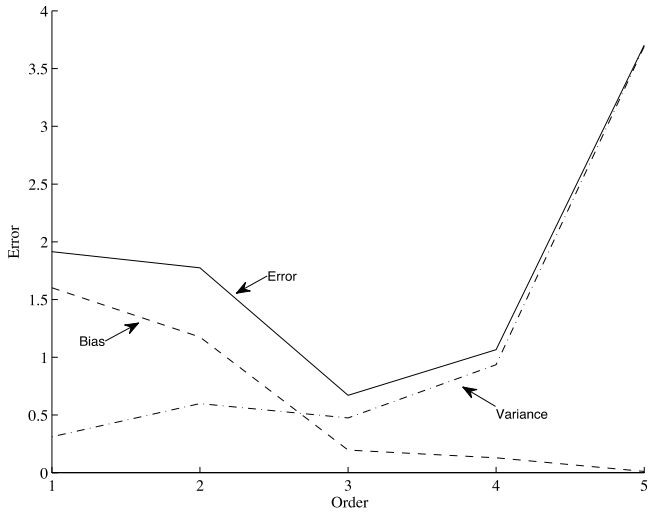
- [Geman et al.;1992] "A neural network that fits closely the provided training examples has a low bias but a high variance. If we reduce the network variance this will lead to a decrease in the level of fitting the data."
- As we increase complexity,
  - bias decreases (a better fit to data) and
  - variance increases (fit varies more with data)
- If variance is kept low, we may not be able to make a good fit to the data (bias increases).
- *underfitting* versus *overfitting*
- The optimal model has the best trade-off between bias and variance

An Example: Polynomial Regression[1]



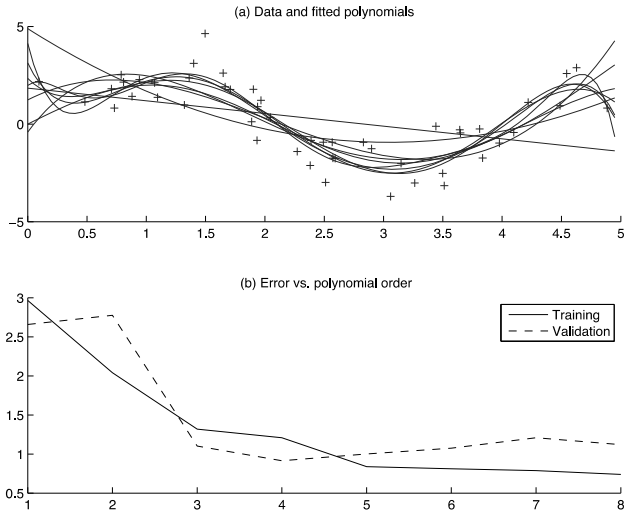
[1]Sourced and reproduced from [Alpaydin;2010, p. 78].

Bias-Variance versus Order[2]



[2]Sourced and reproduced from [Alpaydin;2010, p. 79].

Look for Best Trade-Off[3]

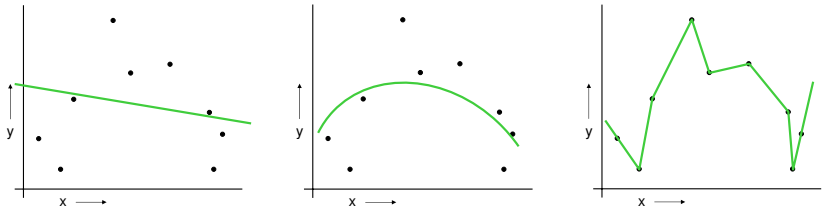


[3]Sourced and reproduced from [Alpaydin;2010, p. 81].

Model Selection: Approaches

- Cross-validation: Measure generalisation accuracy by testing on data unused during training
- Regularisation: Penalise complex models $E' = Err + \lambda C$, where Err stands for error on data, and C is model complexity
- Bayesian Information Criterion BIC $= -2\ln\mathcal{L} + k\ln N$
  - $\mathcal{L}$: likelihood of the model, k: model's number of parameters, N: data size.
- AkaikeÕs information criterion AIC $= 2k - 2\ln(\mathcal{L})$
  - Arguably better than BIC
- Besides BIC & AIC, there are also Minimum Description Length (MDL) and Structural Risk Minimisation (SRM)

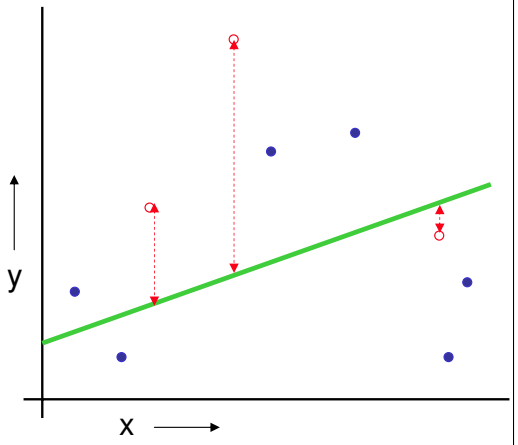
Validation Approaches



- How do we validate our models?
  - E.g., fitting a "curve"
- Typical approaches:
  - Test set
  - Leave-one-out
  - k-fold cross validation
- Each has its own pros and cons

Test set

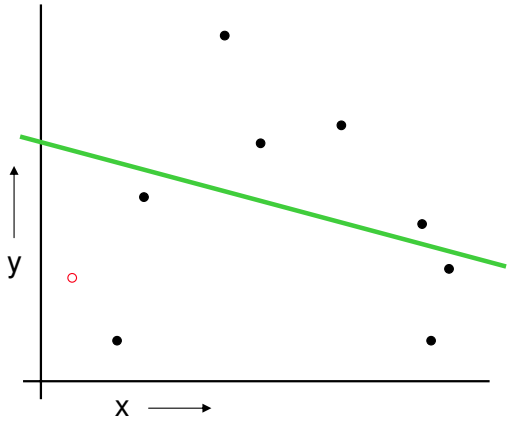
- Randomly choose a portion (e.g. 30%) of the data to be in a test set
- The remainder is a training set
- Perform the regression on the training set
- Estimate your future performance with the test set



Mean squared error MSE=2.4

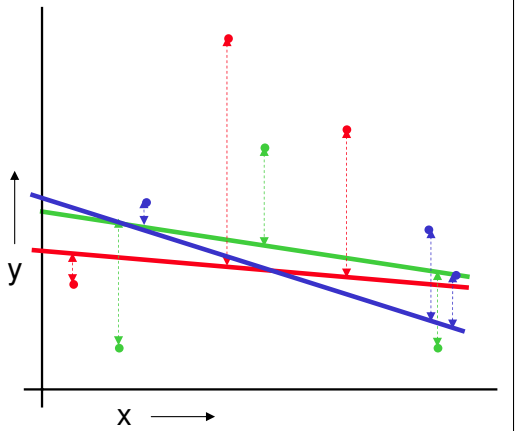
Leave-One-Out Cross Validation (LOOCV)

- For $k = 1$ to N do
  1. Let (x_k, y_k) be the k-th record
  2. Temporarily remove (x_k, y_k) from the dataset
  3. Train on the remaining $N-1$ data points
  4. Test on (x_k, y_k) and note the error
- Report the mean error when all points are done



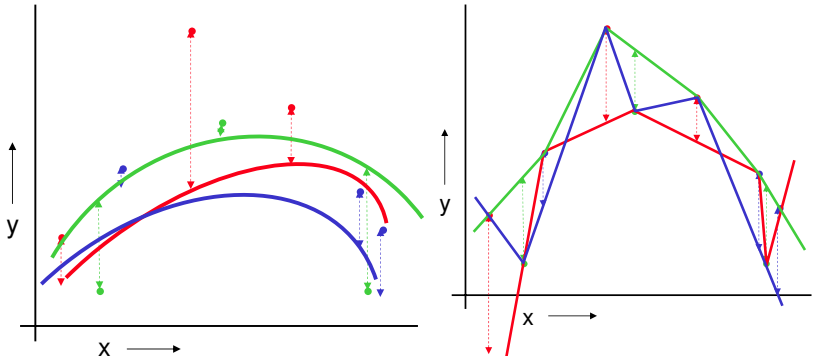
k-fold Cross Validation

1. Randomly break the dataset into k partitions (e.g. k=3, partitions coloured)
2. For the red partition: Train on all the points not in the red partition. Find the test-set sum of errors on the red points
3. Repeat the above process on the *blue* and the *green* partitions
4. Then report the mean error



Linear Regression $\text{MSE}_{3FOLD} = 2.05$

More 3-fold CV Results



Quadratic Regression $\text{MSE}_{3FOLD} = 1.11$

Join-the-dots $\text{MSE}_{3FOLD} = 2.93$

Pros and Cons

Methods	Cons	Pros
Test set	Variance, unreliable estimate of future perf.	Cheap
LOOCV	Expensive; has some weird behaviour.	Doesn't waste data.
10-fold CV	Waste 10% of the data; 10 times more expensive than test set.	Only wastes 10%. Good statistical characteristics.
3-fold CV	Wastier than 10-fold.	Slightly better than test-set.

Clustering Revisited

- How to find the right k in k-means?
- G-means [Hamerly and Elkan;2003]
- X-means [Pelleg and Moore;2000]
  - Progressively growing k
  - Assessing BIC of clusters; stop at certain complexity level.
  - $BIC(M_j) = \log \prod_i p(x_i|M_j) - \frac{k_j}{2} \log N$
    k_j: number of parameters in M_j

X-means: Example[4]

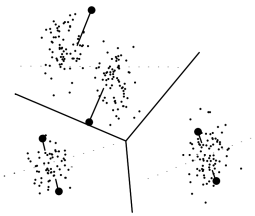


Figure 3: The first step of parallel local 2-means. The line coming out of each centroid shows where it moves to.

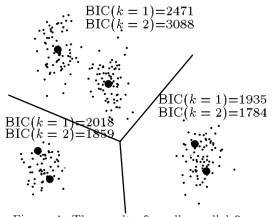


Figure 4: The result after all parallel 2-means have terminated.

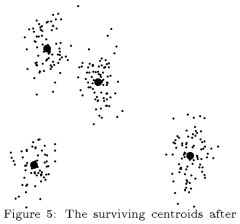
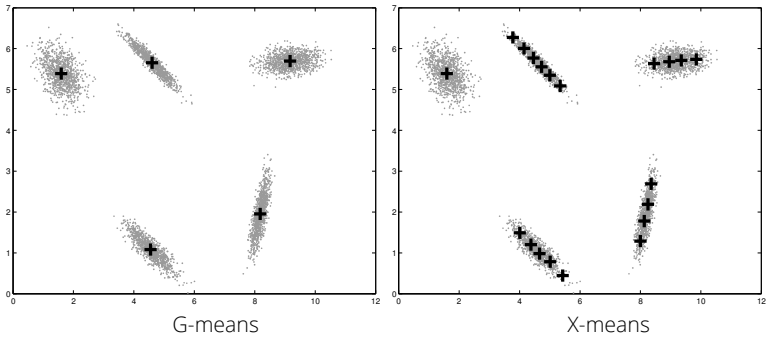


Figure 5: The surviving centroids after all the local model scoring tests.

[4] Sourced and reproduced from [Pelleg and Moore;2000, p. 730].

G-means vs X-means[5]

According to Hamerly & Elkan, BIC causes X-means to overfit the data.



[5] Sourced and reproduced from [Hamerly and Elkan;2003, p. 6].

Recap

- Bias-Variance dilemma
- Model selection: principles and methods
- Cross validation
- References:
  - A. Moore, Cross validation, `http://www.autonlab.org/tutorials/overfit.html`
  - [Alpaydin;2010] Chapter 4 & 19
- Next: Combining Multiple Learners

References

E. Alpaydin
Introduction to Machine Learning, 2nd Edition
The MIT Press: Cambridge, Massachusetts, 2010

S. Geman, E. Bienenstock & R. Doursat
"Neural Networks and the Bias/Variance Dilemma"
Neural Computation, **4**: 1–58, 1992

G. Hamerly & C. Elkan
"Learning the k in k-means"
Neural Information Processing Systems (NIPS), 2003

D. Pelleg & A. W. Moore
"X-means: Extending K-means with Efficient Estimation of the Number of Clusters"
International Conference on Machine Learning (ICML), 727–734, 2000