Exam Questions for the Machine Learning Part of 894.305 "Computational Biology"

Peter Sykacek
Machine Learning in Bioinformatics Research Group
BOKU University
peter.sykacek@boku.ac.at

February 28, 2011

The following exam question concern the lecture given by Peter Sykacek in the Computational Biology course. All references given in the sketched answers point to the respective slide numbers in the document ml_cmpbio_hdout.pdf. For further references please consult my slides or for more information the books [1, 2, 3].

1 Discuss the origin of noise and its implications on data analysis

1.1 Slide numbers

page 3, 4, 10, 11.

1.2 Sketched answer

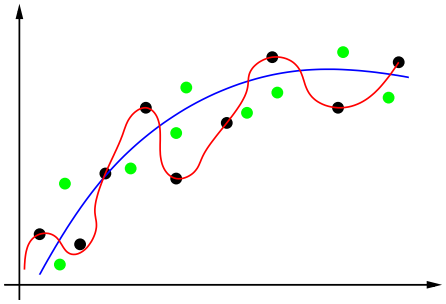
1.2.1 Origin of noise

Noise results from measurement errors (repeating an experiment leads for technical reasons always to different observations), missclassifications (experts get for example the phenotype wrong), and simplifications during modelling. The latter "modelling noise" refers to ignoring certain influence factors in the model which do however alter observations. Although this should be avoided if possible, sometimes these factors are very difficult to control and thus ignored.

Example of the latter: gene expression changes in response to the metabolic state of the model organism (not all organisms are really equally fed at the same time). Such factors must never be confounded with the question at hand since that would lead to misleading results. If they can not be controlled perfectly they will inevitably add random variation to the observations and thus constitute part of the "noise".

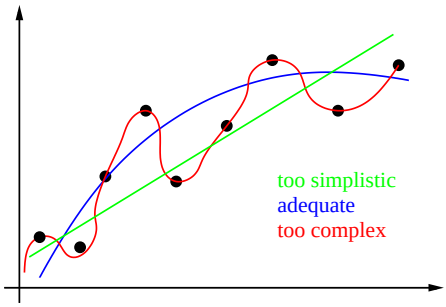
1.2.2 Implications on data analysis

Data analysis attempts finding an appropriate abstraction which explains some given observations (training data) reasonably well. An optimally chosen model should explain new observations and avoid fitting the noise component in the measurements.



In the above sketch, the more complex (red) model fits the training data (black dots) better than the blue one. This improvement is though a result from poor averaging and trying to explain the noise.

A good fit is obtained by local averaging of observations and appropriately adjusting the model complexity (blue model).



In the above sketch the too complex and the too simplistic model are inadequate.

2 List the two data analysis strategies and link application scenarios to data analysis methods

2.1 Slide numbers

Page 16, 24

2.2 Sketched answer

2.2.1 Analysis strategies

There are two distinct strategies for data analysis: Situations which require modelling of a given target variable are in machine learning called “super-

vised problems". Solving such problems requires regression type data analysis methods. Situations which require exploring data for unknown structure are in machine learning called "unsupervised problems". These problems are approached with exploratory methods which either search for unknown groups in the data or for explanations of reduced dimension.

2.2.2 Data analysis applications and methods

Task	$->$	Method
predict continuous y from input data	$->$	Regression
predict discrete y from input data	$->$	Classification
find unknown groups in input data	$->$	Clustering (e.g. k-means, mixture models)
find low dimensional representation for input data	$->$	Dimensionality reduction (PCA, ICA)

3 Provide a brief discussion of two important supervised learning tasks

3.1 Slide numbers

Page 17, 19, 20

3.2 Sketched answer

The two different regression tasks listed below do only differ in the nature of the response variable (i.e. the type of variable we want to predict): "Classical" regression is concerned with continuous valued responses. Classification is concerned with discrete valued response variables.

3.2.1 Regression with continuous response variable

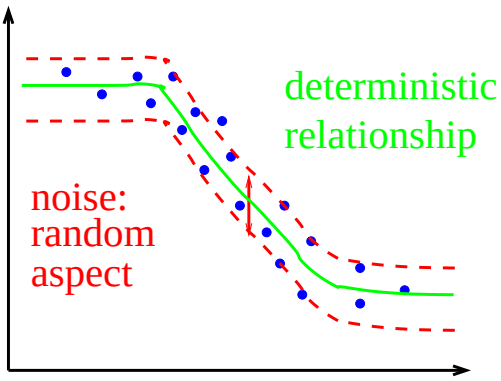
The regression task uses a noisy data set $\mathcal{Z} = \{(y_1, \boldsymbol{x}_1), ..., (y_N, \boldsymbol{x}_N)\}$ from a life science experiment which allows separating the variables into one continuous valued target variable y which we would like to predict as good as possible and a set of independent variables $\boldsymbol{x}$ which are assumed providing information about the value of the target variable. Regression fits based on $\mathcal{Z}$ an "optimal" function relating $\boldsymbol{x}$ and y:

$$y = f(\boldsymbol{x}; \boldsymbol{\theta}) + \epsilon(\lambda)$$

Noise requires a deterministic and a random component.

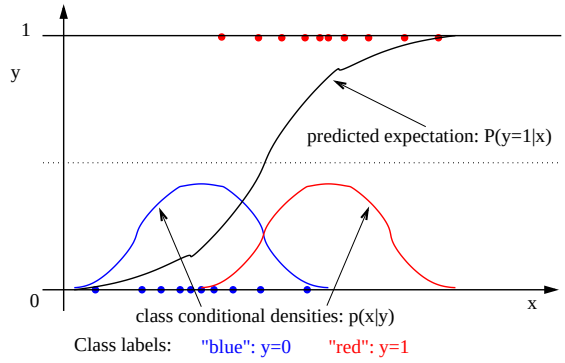
The noise which affects measurements implies that y is a random variable and that we do best when predicting expected y values from x (local

averages). The complete description of the regression model includes noise characteristics. The red error bars represent the standard deviation around the expected y value. This is a complete description in the case of Gaussian noise.



3.2.2 Classification

Although it appears to be different, classification is just a special case of regression with the target values being discrete. The general case allows for multiple class labels. Binary classification is a special case, where $y = \{0, 1\}$. In this case the predicted expectations are between zero and one and correspond to the posterior probability for class label 1, $P(y = 1|x)$.



4 Discuss two important instances of unsupervised learning

4.1 Slide numbers

Page:21, 22, 23

4.2 Sketched answer

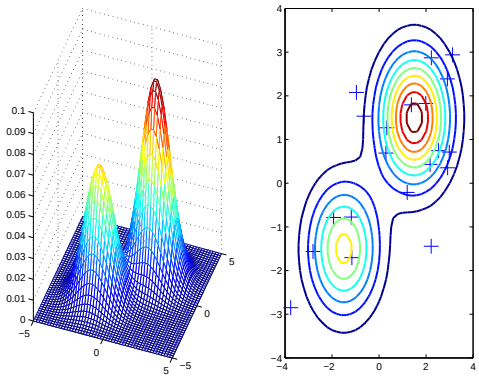
Both method classes for unsupervised learning attempt providing a simplified view of a data set. These approaches are useful for exploring data when

fitting a model for a dedicated variable is not intended. *These approaches should not be used if we want to answer a question about relations between different variables!! In this case we have to use regression type models.* The two approaches discussed below differ in the nature of the summary they provide about a data set: Clustering and mixture density models provide a discrete summary which tells us which component of the model did most likely "generate" a particular observation. Continuous latent variable models provide a continuous summary which provides most of the information in a data set and is used for visualising data in a lower than original dimension.

The problem statement in unsupervised learning or exploratory data analysis is finding unknown structure in a data set $\mathcal{X} = \{x_1, x_2, ..., x_N\}$, x_n distributed according to unknown pdf $p(x)$. The learning task is summarising x by an unobserved variable t.

4.2.1 Mixture density models and clustering

Mixture density modelling represents the data generating density as $p(x) = \sum_k P(t = k)p(x|t = k)$, with $t \in \{1, .., K\}$ being a discrete variable. The term $p(x|t = k)$ represents the component density and $P(t = k)$ the prior probability that kernel k generates samples. An important example is the so called mixture of Gaussians model which uses $p(x|t = k) = \mathcal{N}(x; \mu_k, \lambda_k)$, i.e. Gaussians as component densities.

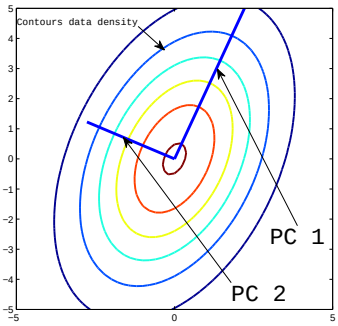


The figure above visualises this graphically. The left hand side shows a mesh plot of a resulting density over a two dimensional data set. The left hand plot shows the contour lines and the training data $\mathcal{X}$ which was used for model fitting. As a summary we obtain from a mixture of Gaussians model the component number $t = k$ which was most likely responsible for generating a data point $\boldsymbol{x}$.

4.2.2 Continuous latent variable models and dimensionality reduction

Continuous latent variable models use a similar representation of the data generating density. The summary provided about the data is captured by a continuous valued latent (unobserved) variable t and the summation in the mixture model is replaced by an integral $p(x) = \int_t p(t)p(x|t)dt$, $x \in \Re^k$. The latent variable is multivariate and continuous $t \in \Re^d$ and typically lower dimensional than the data itself, $k > d$.

Although non probabilistic (i.e. the model does not really use the density formulation above), principle component analysis (PCA) is an example of latent variable modelling. PCA represents a data point x as $x = m + W_1 t_1 + ... + W_d t_d$ with $x \in \Re^k$ and $t = [t_1, ..., t_d] \in \Re^d$ and $W_d : [k \times 1]$ denoting the d-th eigenvector of the sample covariance matrix. The underlying assumption is thus that the data set $\mathcal{X}$ was generated by a d dimensional Gaussian density.



The above figure illustrates the mapping we obtain when assuming that the contour lines represent curves of equal density under the data generating Gaussian. The summary provided by PCA are the projections on the principal component directions. We typically use fewer principal components than data dimensions and reduce thus the dimensionality of the data.

5 Formulate an objective function which can be used for model fitting and discuss its most important limitation

5.1 Slide numbers

Page: 26, 27

5.2 Sketched answer

5.2.1 An objective function for model fitting

Assuming that we may influence the behaviour of the model by changing the model coefficients, the objective of model fitting is minimising the discrepancy between the predictions we obtain with the chosen model and the given training data.

The goal of model fitting is thus tuning $\boldsymbol{\theta}$ such that $f(\boldsymbol{x}_n; \boldsymbol{\theta})$ represents all $(y_n \boldsymbol{x}_n)$ pairs well.

In order to achieve this goal, we need an expression we may optimise (maximise, minimise) for fitting all n "training" samples well.

One possible choice is using the so called sum of squared errors (SSE). The idea of the SSE is subtracting the deterministic part from all y_n. We get thus $\epsilon_n = y_n - f(\boldsymbol{x}_n; \boldsymbol{\theta})$. To penalise deviations for all data points in both directions equally, we sum over the squared difference.

$$\text{SSE} = \sum_n \epsilon_n^2 = \sum_n (y_n - f(\boldsymbol{x}_n; \boldsymbol{\theta}))^2$$

There are other objective functions as well. One of them is the so called (log)-likelihood.

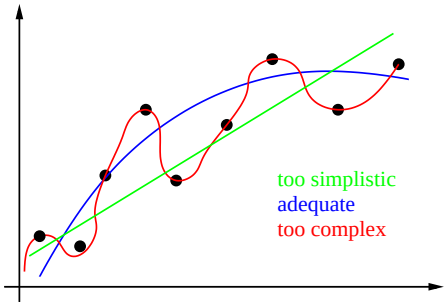
5.2.2 The most important limitation of the SSE

The most important limitation of the SSE is apparent, by analysing the behaviour of the SSE when changing the complexity of the model. Assuming that we have a linear model, the true relation in the data is captured by

$$y_n = \boldsymbol{x}_n^T \boldsymbol{\theta} + \epsilon_n.$$

This corresponds to a situation, where each sample can only be explained up to a small but finite ϵ_n, which must be there due to the noise component we have in all data sets. The SSE of the optimal model is thus larger than zero.

In general the optimal model is unknown and finding the appropriate complexity part of model fitting. To allow in the above example for models of varying complexity we can for example introduce nonlinearities (e.g. by allowing for arbitrary powers of x) and increase the number of parameters (here implicit in the dimension of the vector $\boldsymbol{\theta}$.



Doing so, we will eventually get curves like the red one in the above illustration which pass through all points in the data set exactly. This corresponds to having all ϵ_n equal to zero and thus an SSE of zero, which is the smallest value the SSE can attain. Adapting the model complexity during fitting can thus not be obtained by applying simple objective criteria like SSE since this will inevitably lead to too complex models. This property of the SSE and other objective functions such as likelihoods is also the reason, why over complex model are much more of a problem than too simplistic models.

6 Discuss the philosophical principle of getting model complexity right and list three important implementations of this principle in machine learning

6.1 Slide numbers

Page:29, 30, 31

6.2 Sketched answer

6.2.1 A philosophical principle for choosing appropriate complexity

Human intuition will in most questions about appropriate complexity chose a model which is "just complex enough" to explain the observations. This principle appears as in the work of Karl Popper as requirement that scientific theories must be "falsifiable". The accounts of this idea date however back to William of Occam (1288 - 1348) and are now known as Occam's Razor. The corresponding statement of William of Occam can be interpreted as a statement that we should always opt for an explanation with as few as possible causes, factors, or variables which explains observations reasonably well.

6.2.2 Occam's razor in machine learning

To apply Occam's razor in machine learning applications we can

- add a complexity penalty to objective functions like SSE or likelihoods. There are many such choices like AIC (Akaikes information criterion), BIC (Bayesian information criterion) or MDL (minimum description length).
- use learning frameworks like Bayesian inference which have Occam's razor built in.
- use empirical approaches for comparing model classes with validation tests. The latter are computer simulations mimicking evaluations on independent data, that is data which was not used for adjusting the model coefficients.

7 Discuss objectives, measures and algorithms for validation testing

7.1 Slide numbers

Page:33, 34, 35

7.2 Sketched answer

7.2.1 Objective of validation testing

Validation tests are used for estimating the performance of fitted regression models when applied to future test cases. Due to inherent dangers of adjusting over complex models too close to the training data, such estimation must be based on data that was not used for training. Otherwise the results will be biased (i.e. over optimistic). Validation testing is done for the purpose of comparing the suitability of different methods for a particular analysis task. One application is for choosing an appropriate model complexity.

7.2.2 Measures for validation testing

An example for a measure of validation capability for continuous valued regression is the so called mean square generalisation error (MSE_{test}, average SSE!).

$$\text{MSE}_{test} = \frac{1}{N} \sum_n \left(y_n^{test} - f(\boldsymbol{x}_n^{test}, \boldsymbol{\theta}^{train}) \right)^2$$

Classification is often tested via the generalisation accuracy acc_{test}.

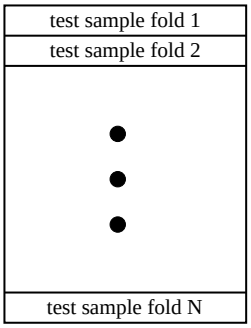
$$\forall n \; \hat{y}_n^{test} = \text{argmax}_k(P(y = k | \boldsymbol{x}_n^{test}, \boldsymbol{\theta}^{train}))$$

$$\text{acc}_{test} = \frac{1}{N} \sum_n \delta(y_n^{test}, \hat{y}_n^{test})$$

For estimating generalisation accuracy, we classify test samples such that the most probable class wins and estimate the fraction of correctly classified test cases.

7.2.3 Estimation procedures for validation tests

Estimating the validation capabilities of fitted models requires overcoming a dilemma. There is a trade-off between well calibrated model coefficients which requires large training data sets and reliable estimates of the test set performance which requires large test sets. To remain unbiased and avoid over optimistic validation results, one must never use test cases which were used for model fitting. All together is achieved by a computer simulation which is called N-fold cross testing. As is sketched below, the approach splits the entire data set into N equally sized chunks and uses each chunk in turn as independent test data, whereas the remaining $N-1$ chunks are used for model fitting.




```
allres=[]; allpred=[];
for n=1:n_folds
    % split into training and test data
    [train, test]=foldsplit(orig_data, n_folds, n);
    % model inference
    [model]=trainfunc(train, fiddleparams);
    % store this folds true targets and predictions
    [res]=truetarg(test);
    [pred]=predtarg(test, model);
    allres=[allres; res];
    allpred=[allpred; pred];
end
```

Variants of this approach are "leave one out" which has as many folds as samples. An alternative by resampling with replacement is called Bootstrapping.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, 2006.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, Second Edition.* Wiley, New York, 2000.

[3] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, Cambridge UK, 2003.