Learning from Data: Generalisation

Amos Storkey, School of Informatics

November 7, 2005

http://www.anc.ed.ac.uk/∼amos/lfd/

Various Terms

- Regularisation.
- Overfitting.
- Prior parameter distributions.
- Validation set.

All About Generalisation

- We have talked about maximum likelihood learning.
- In fact maximum likelihood learning is problematic.
- Problems show up when the number of parameters is large.
- The fundamental problem is called *overfitting*.

But isn't Maximum Likelihood *the Right Thing*?

- Well actually no, because ...
- ...picking one maximum likelihood parameter doesn't take into account the fact that there might be
  - Other nearby settings which could be almost as good, but have qualitatively quite different effects.
  - Completely different parameter setting which are also good.
  - A different large group of parameter settings which are all different but have qualitatively similar effects.
- In other words...
- We haven't taken into account the distribution of parameters.

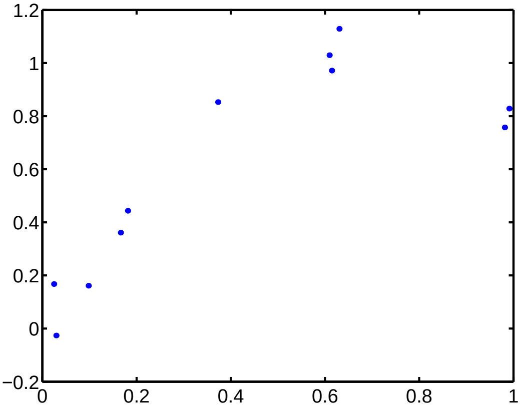
Distribution of Parameters??

- But the parameters are just numbers.
- Maybe. But are you certain about what they should be?
- Use distributions to represent uncertainty.

Back to the Beginning

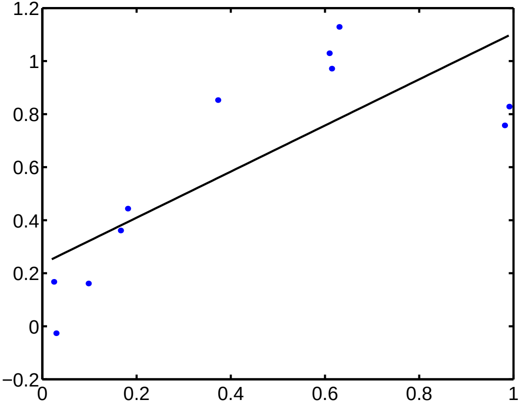
- Let us look at some simple data and look at various polynomial fits.
- We know how to do polynomial fits now: we use a generalised linear model, and the pseudo-inverse solution.
- We will try various orders of polynomial.

Some Data



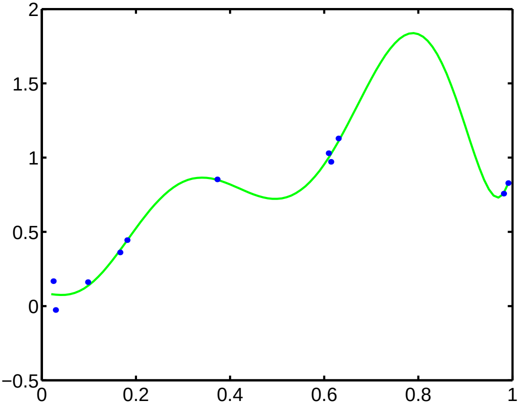
11 data points

Linear Regression



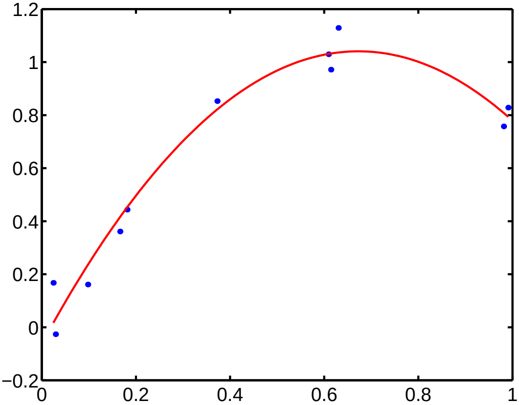
A linear fit to the data

Polynomial Fit



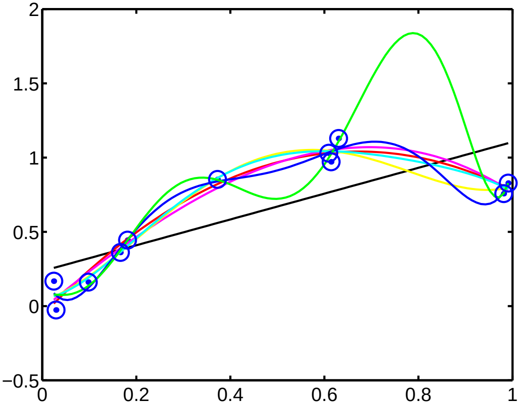
A seventh order polynomial fit to the data

Polynomial Fit



A second order polynomial fit to the data

Polynomial Fit

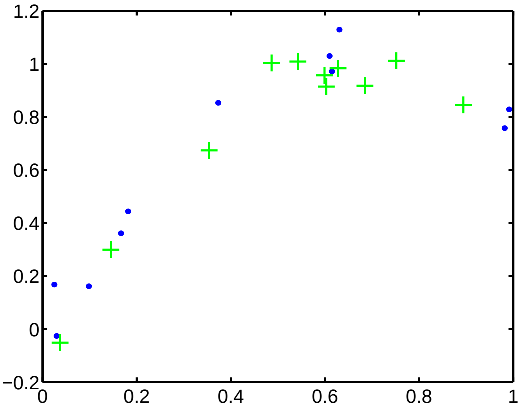


An overlay of first to seventh order polynomial fits

So Which is the Best?

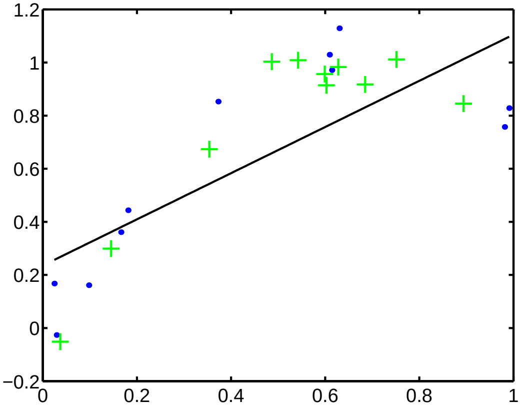
- More parameters = more powerful.
- More parameters will fit the data better: minimising error
- But how well will it predict new data?

Test Data



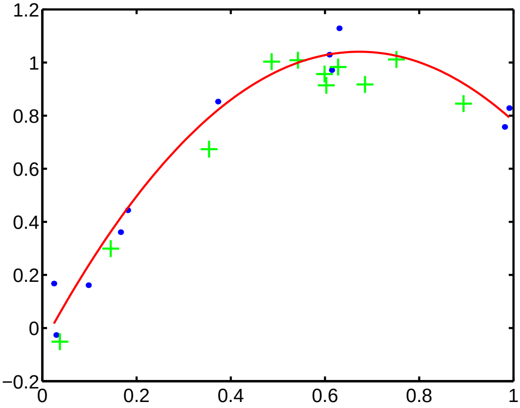
$+$ indicates new data

How Well does the Method Predict?



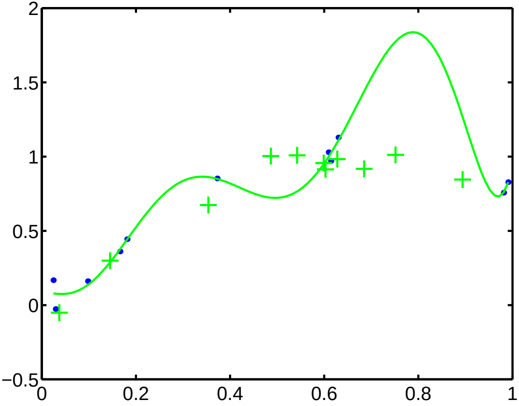
First order prediction. Not great.

How Well does the Method Predict?



Second order prediction. Pretty good.

How Well does the Method Predict?

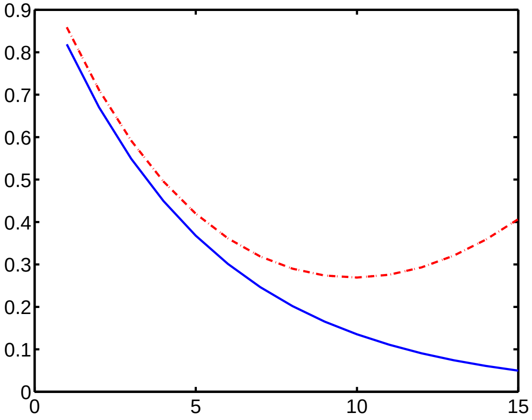


Seventh order prediction. Oh dear.

Test Error versus Training Error

- Increasing the power of the model will improve the training error.
- However that does not mean it will necessarily perform well on a test set.
- For more powerful models, we find the model fits itself to the noise in the data, and tries to model that noise deterministically.
- This is called *overfitting*.

Typical Test versus Training Error.



Training error (blue) and test error (red) with increasing model power

What Should We Do?

- We could set aside some data for validation purposes, and then see what order of polynomial to use based on performance on this *validation set*
- Training set: for learning the parameters of the model.
- Validation set: for model selection between different possible models.
- Test set: check how well the final chosen model performs.
- Note this approach, and much of this discussion applies more generally than just for polynomials.

The Whole Process.

- Decide on a set of models to test (eg a set of polynomial model orders).
- Learn the parameters for all these models using maximum likelihood learning.
- Check the performance of each model with the maximum likelihood parameters on the validation set.
- Use the (log) probability of the validation data given each model as the performance measure.
- Pick the model which performs best on the validation set.
- Test it on the test set to see how well you should expect it to perform.

This is Nonsense!

- A seventh order polynomial contains a second order polynomial as a special case.
- There should be some way to automatically learn a seventh order polynomial that is at least as good as a second order one.
- We use exactly the same data in each case. So why is this not happening?
- Or in other words... what is wrong with maximum likelihood!

Three Problems: Problem 1

- Problem 1: we haven't provided our priors.
- If we believe an exceedingly squiggly line is worse than a flat one, we certainly haven't told anyone.
- Maximum likelihood treats all parameters as equally valid. But they are not.
- For example we are likely apriori to be happier with a polynomial $y = 0.8x - 0.4$ than with $y = 20200.33 + 3932x^2 - 44x^3 + 2923x^4 + 21045x^5 + 140x^8 + 30x^{15}$ as a solution.
- So we can encode this by putting a prior distribution over parameters W: $P(W)$. Commonly this might be a Gaussian prior.

Three Problems: Problem 1

- Then we can calculate the *maximum a posteriori* parameter solution.
- Instead of $\max \log P(data|W)$ we calculate $\max \log P(W|data) = \max(k + \log P(data|W) + \log P(W))$.
- This approach is also called regularisation. It involves adding a penalty term $\log P(W)$ to the log likelihood which penalises large parameter values.
- Note that for Gaussian $P(W)$, $\log P(W)$ is quadratic.
- Here we have taken an important step. Parameters W have become random variables and are treated in just the same way as unseen data: we calculate posterior distributions.

Three Problems: Problem 2

- Maximum likelihood model selection chooses the model order k according to $P(data|W^*, k)$ where $W^* = \arg\max P(data|W, k)$.
- Hence maximum likelihood model selection will choose a higher order model over a lower order one.
- This is problematic as really we want to know $P(data|k)$.
- This is called Bayesian model selection, and it involves choosing k to maximise $P(data|k)$ instead of $P(data|W^*, k)$.
- The details of this approach is beyond the scope of this course.

Three Problems: Problem 3

- We should look at the results of using all high posterior probability parameters, not just the highest.
- In fact we should average over the predictions for each of the parameters weighted by the posterior probability.
- That is we want $P(test\ data|k)$ not $P(test\ data|W^*, k)$.
- This is called the full Bayesian inference for the target values. We integrate out over all the possible parameter values.
- The details of this approach is beyond the scope of this course.
- See e.g. Bishop chapter 10 for more details of these last two approaches.

Regularisation for Generalised Linear Models

- We set a prior $P(\mathbf{w})$ for the parameters $\mathbf{w}$ of the generalised linear model $y = \mathbf{w}^T\phi$.
- Let $\mathbf{w}$ have a zero centred d dimensional Gaussian distribution

$$P(\mathbf{w}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp(-\frac{\mathbf{w}^2}{2\sigma^2})$$

- Then then negative log posterior $-\log P(data|\mathbf{w}) - \log P(\mathbf{w}) + \log P(data)$ can be written

$$A(\sum_{\mu=1}^{N}(y^\mu - \mathbf{w}^T\phi^\mu)^2 + \lambda\mathbf{w}^2) + B$$

using the notation from the previous lecture.

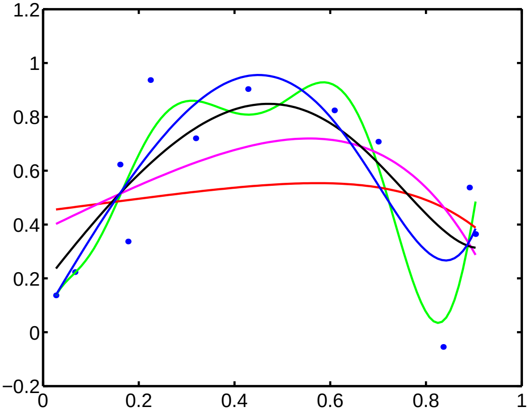
Regularisation for Generalised Linear Models.

- We can then differentiate this w.r.t **w**, and find the optimal **w** given λ. Then we get

$$\mathbf{w} = (\mathbf{\Phi}\mathbf{\Phi}^T + \lambda I)^{-1}\mathbf{\Phi}\mathbf{y}$$

- In other words we have a simple modification to the pseudo inverse solution.

Regularisation



Regularisation for various values of λ

Summary

- Overfitting
- Maximum likelihood problems
- Model selection
- Bayesian methods