Learning from Data: Real Data

Amos Storkey, School of Informatics
University of Edinburgh

Semester 1, 2004

Summary

- Handling data
- What to use when?
- Discriminative or generative
- Assessing prior assumptions
- Typical data issues
- Performance measures

Handling Data

- Always allow a significant amount of time for pre-processing and formatting data.
- Always randomly reorder the data before selection of training/test/validation sets.
- Ensure you spend enough time understanding the characteristics of the data and the problem before doing anything else.
- Use visualisation tools to help you.
- Ensure you set aside enough data for testing. At each stage of analysis you might need to use new test data. Save some test data for the case where you are 100 percent certain you will not make any further changes.
- Keep a fully invertible audit trail of how you got from the original data to the working data.

Encoding Data

- Follow rules for encoding *logical*, *ordinal*, *categorical* and *real* data.
- For many methods data should be rescaled. In general data should be scaled to zero mean and unit standard deviation. However other methods may be more appropriate based on knowledge of the form of distribution.
- For neural networks, weights should be initialised to avoid saturation of the nodes (see notes).
- For high dimensional data consider feature selection or feature extraction or other dimensionality reduction methods.

What to Use and When

- Which tool should be used in what circumstances?
- In a hurry? Use a pre-packaged tool. Ensure the package is robust and reliable. Start simple, progress to more complicated models if you have to.
- Will the tool provide answers in the right form? E.g. want logical rules as the output - decision trees will suit you better than neural networks.
- Does the tool map well to your assumptions: e.g. continuous or discontinuous functions? What form might reasonably discriminative features take? Local (rbf) or partitioning (neural networks)?

What to Use and When cont.

- Does the underlying distribution of the data matter? If so consider generative rather than discriminative procedures?
- Any history of methods for a similar application?
- Do you have time to encode more complicated modelling techniques/build feature sets etc? If so use the simpler methods as benchmarks and to provide suggestions.
- Will the tool cope with the quantity or size of the data you have? e.g. RBF networks are faster to train than MLPs for regression.

What to Use and When cont.

- Is the data significantly biased towards a particular class? This needs to be considered in setting error measures. Might need to pre-select training data.

- How will your outputs be used? Will they be fed in to another system? What information do you really need about the outputs?

- Try to use a tool which will produce error bars (the standard deviation of the predicted values), otherwise you have no idea of your confidence. Compare computed error bars with actual errors on a test set.

- Use good optimisers. Always! Does it matter if you only obtain a local minima?

Discriminative or Generative

- Consider how important the distribution of the data is to the problem.
- Can you assume the distribution of future test data will be the same as the training data?
- Do you need to use the distribution information of output values for future processing?
- Are you asking generative questions about the data?
- Are you able to characterise the distributional information well using your model?

Assessing Prior Information

- **Structural information**: what elements can reasonably be considered independent or conditionally independent?
- Are there any hidden concepts that we can bring in to play to *explain* some aspect of the data?
- Are there reasonable smoothness assumptions or assumptions regarding feature structure that can be made.
- **Quantitative information**: are there particular forms of model we have a reasonable bias towards?
- Are there sparsity assumptions that can be made (many things set to zero)?
- **Empirical information**: are there other plentiful sources of information that can be used to build empirical prior distributions for the problem.

Typical Data Issues

- Missing data
  - Why is data missing? Is it missing at random, or is there some systematic reason for missing data.
  - Missing attributes/labels? Again, is this random? Do we need to choose a method which deals with occasional missing attributes properly (e.g. Naive Bayes).
  - Are we required to fill in for missing data, or can we use a process which infers missing labels/attributes on the fly (e.g. EM).

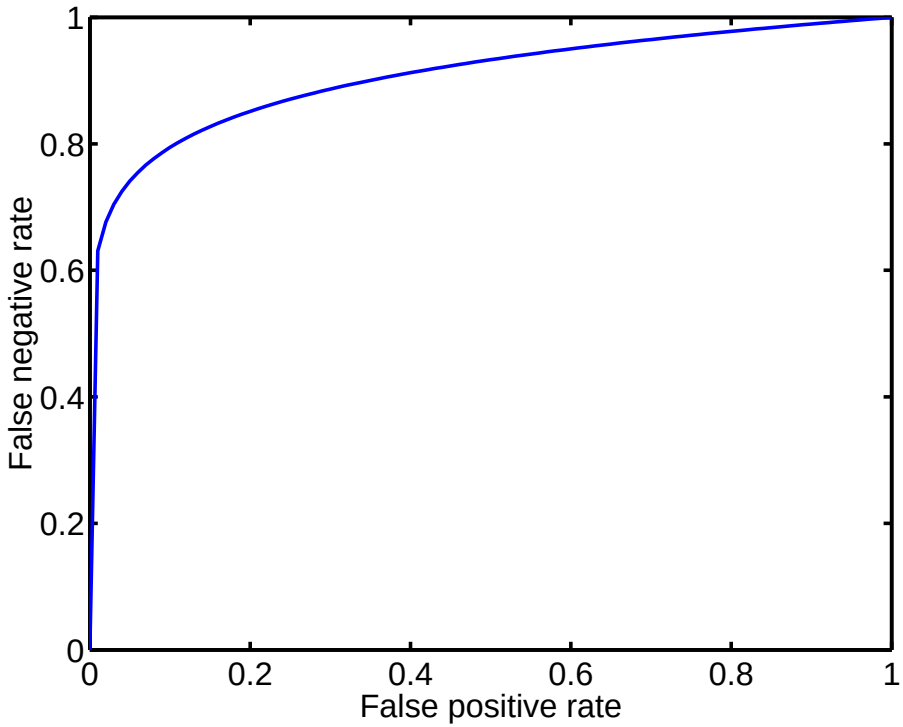
Typical Data Issues

- Dominating class(es) - significant bias towards one or more class labels.
  - Ensure your error measure matches your utility. Is it much more important to classify the minority class correctly? If so adapt your error measure.
  - Need enough data for minority classes. Can balance data and adjust error measure accordingly.
  - Local minima: easy to settle on a model which effectively ignores minority classes. Harder to find better model which incorporates minority class variation - adjust error measure.

Output and performance measures

- Adjust probability thresholds
- Error reject curves
- Loss matrices
- ROC (Receiver Operating Characteristic) curves: False Negative versus false positive.

ROC curves


1
0.8
0.6
0.4
0.2
0
False negative rate
0
0.2
0.4
0.6
0.8
1
False positive rate

At the end of the day

- Need both data and priors to get answers to questions. Which means:
- Your data must be meaningful - represent it well
- Your questions must be meaningful: make sure you are asking the question you want to ask rather than the one some model says you might want to ask.
- Your priors must be meaningful - choose good models and good distributions, where good means a good fit to your domain.
- This means you need to know the assumptions behind a particular model.
- Be sceptical of claims of universality.
- Don’t substitute black box machine intelligence methods for using your brain.