Going further

Recap – what we've learnt so far

- Partitioning variance:
  - Chi-sq, Student's T
  - ANOVA, regression
  - GLMs, GLIMs and others
- Techniques to do so robustly:
  - Model selection
  - Model comparison
  - Model optimisation / fitting

Pervasive issues

- Independence
- Autocorrelation
- Numerator/denominators in tests
- Strategies and gradients

Independence

- We have assumed that samples in our population of interest are **independent and identically distributed (iid)**
- This can be violated very, very easily:
  - Errors and biases arising from the experiment or dataset itself
  - Systemic/phenomenological reasons for autocorrelation

Autocorrelation (I)

- Two or more samples share information or dependence, such that (e.g.) the N^{th} sample provides information about the N+1
- Parameters may also be autocorrelated, e.g car distance and fuel consumed

Autocorrelation (II) – spatial autocorrelation

- If not explicitly modelling spatial phenomena, we should be aware that at arbitrarily close physical scales, adjacent samples will be correlated
- We can include spatial terms (or a surrogate – blocking)
- We can separate by large enough distances in space
- We can bootstrap / jacknife, or pair samples with distant ones

Autocorrelation (III) – time-series

- As with spatial sampling, but even more strongly, time-series phenomena will generate adjacent samples that are highly correlated
- Explicitly include e.g.
  - smoothing, sliding-window analyses
  - Kernel density estimation
  - Fourier transforms
- Alter sampling: downsample / jitter / random sample

Autocorrelation (IV) – pylogenetic non-independence

- Unlike spatiotemporal autocorrelation, phylogenetic non-independence is less obviously a problem, but it is:
- Evolution means no comparison among multiple species is orthogonal – at least two will be more closely related (and reciprocally) than the others, simply because they diverged more recently
- Harder to correct for – use phylogenetic comparative method

Significance testing

- All statistics boils down to making statements about our belief that some event has, or will, occur, based on underlying structure/model
- Invariably this involves a comparison between models, with evidence in favour (numerator in F test) and against (denominator)

Numerators

- Numerator e.g. evidence for usually easy to calculate or estimate
- But we may be uncertain whether we have the optimal value (false negative)
- Especially true for high dimensions

Denominators

- a.k.a evidence against
- May be hard to calculate
- Null may be ill-defined, or several plausible nulls
- Prior distributions may be hard to sample

Test / comparison statistics

- Weight-of-evidence calculations usually realised through our test statistic, under certain assumptions
- E.g. Student's T test: "difference in means is test statistic, *t*; *t* sampled from T distribution if data normally distributed; iid"
- We've met others; F, Chi-sq etc
- Other approaches: odds-ratio, AIC, BIC, Bayes factors, etc
- No correct one, just different assumptions

Model design

- Models aren’t given on stone tablets. Every model we’ve met was devised by *someone*
- Nothing preventing us from joining the party
- Models can be numerically explicit with help from mathematicians; or visual.
- If we can code it we can model it.
- Simple models usually preferred if biologically realistic.

Search spaces and optima

- Parameter optimisation / model fitting will rarely take place on monotonically increasing, smoothe search spaces
- We should always bear in mind that ‘our’ best model may not be ‘the’ best
- Similarly, many other ‘fairly good’ models/ parametisations may lie close in model-space but have radically different parameter values / model terms

Strategies

- Especially for complex problems, our inference will only be as robust as our strategies to find an optimum
- Most searches split into a quick-and-dirty vs and a robust and slow phase
- Rapid gradient descent from starting points and some combination of cunning (random jumps, exchanges, etc)

Other techniques

Non-parametric stats

- Rank-sums
  - Mann-Whitney U, Spearman's rank correlation..
- Distribution comparisons
  - Kolmogorov-Smirnov

MCMC

- “Wouldn’t it be good if we could sort of hop around likely values in the parameter space?” – there’s an app for that…
- MCMC lets us sample from complex distributions with the nice property that parameters’ estimates’ confidence can be ascertained
- Unfortunately, while writing and running MCMCs is simple, obtaining good and efficient mixing may not be

MCMC

- Consists of a chain of states (x values) with the Markov property e.g. state x_{t+1} depends entirely and only on state x_t and a transition matrix
  - Propose new guesses, x' from sampling distribution
  - Calculate probability Pr(x'|D) under target distribution, D(x)
  - Generate a random number A
  - Accept x' if Pr(x'|D) > A

Machine learning

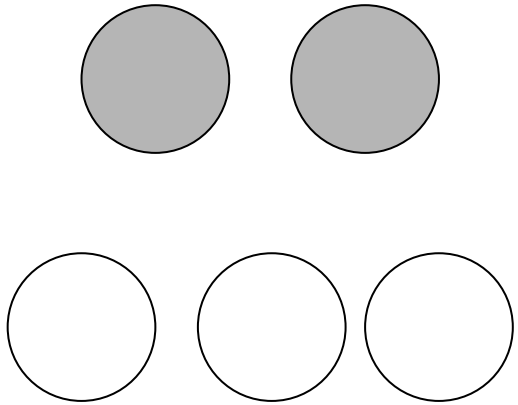
- Both *supervised* and *unsupervised* machine learning attempt to discover higher-order and/or higher-dimensional patterns, based on ‘training’ (labelled) data
- Can then be evaluated on test dataset
- Good volume and *quality* of training data essential
- Often data scaled [-1,1]
- RIRO

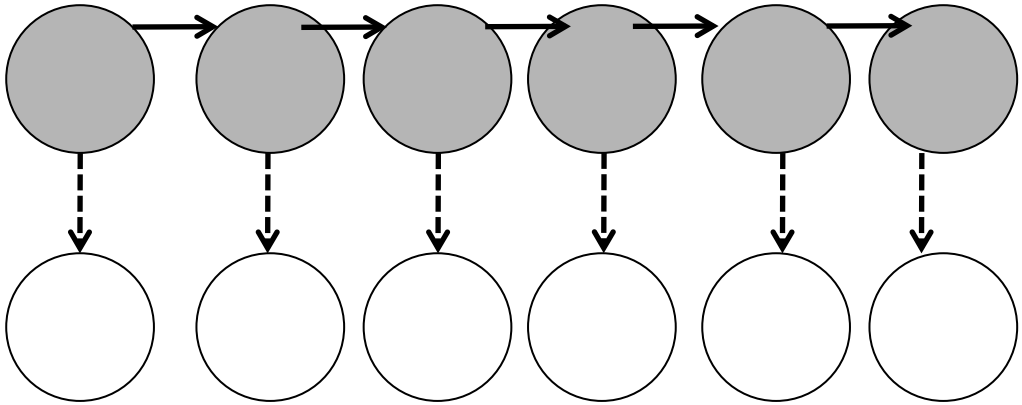
HMMs

- Consider levels of a phenomenon including *hidden* states we cannot observe directly
- Model as a state-change system
- Very powerful but can take time and data to train
- Once trained can *sometimes* be used for slightly unrelated problems
- E.g. gene annotation

HMMs

- Define a series of hidden states or *latent variables* $X=\{x_1, x_{2,...}x_i\}$, and observed states $Y=\{y_1, y_{2,...}x_n\}$
- Transition probabilities define movement between hidden states
- Emission probabilities define chance of observing a state for each possible X
- If we have a trained HMM we can evaluate probability of observations
- If we have observations and some form of labelling we can train HMM (assign weights to transition and emission probabilities)





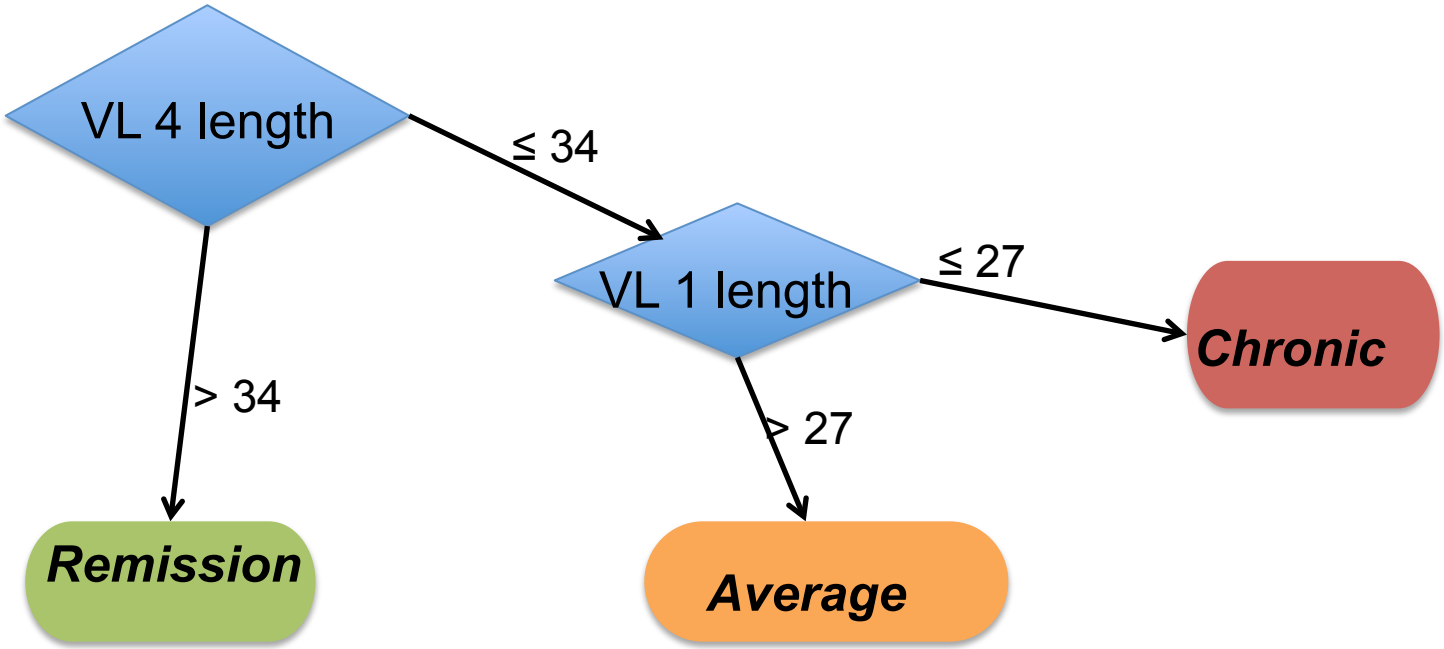
Decision trees

- Simple classifications of the data, based on splits/partitions of data
- Simple/fast to generate and very easy to interpret
- Very heterogeneous data eg.
  - Genetic features
  - Real-valued parametric numerical observations
  - Categorical data
- But prone to overfitting, local optima, and sensitivity to training data

Decision trees

- For each split, achieve most efficient labelling of the data
- Splits / decisions structured in order of variation explained
- We may iterate e.g. ‘forestry’ and employ various combinations of cunning to make efficient, globally-optimal trees

Decision trees



SVMs

- Support vector machines: find an optimal partitioning of the data in higher-dimensional space
- Same principle as decision trees, but interaction
- Can be very efficient but harder to understand output

SVMs

- *Label* the data discretely
- Each datapoint *d* is a *vector* of values in {*w,x,y,z…}*
- Find a partition in the data which completely ('hard-margin') or maximally (soft-margin) separates $d_{label\ 1}$ from $d_{label\ 2}$ (a *hyperplane*)
- Gap between groups is the margin
- Points (vectors) at the margin are the *support vectors*
- Where linear hyperplanes do not work, we can use a *kernel* to reshape the space so they do. A handy cheat!

K-means and clustering

- Find structure in the dataset e.g. interactions between sets of parameter values
- Choice of clustering method and initialisation can have a big impact
- Good for dimensionality reduction e.g. 1000s of cancer cell genes →9 cancer genotypes

K-means

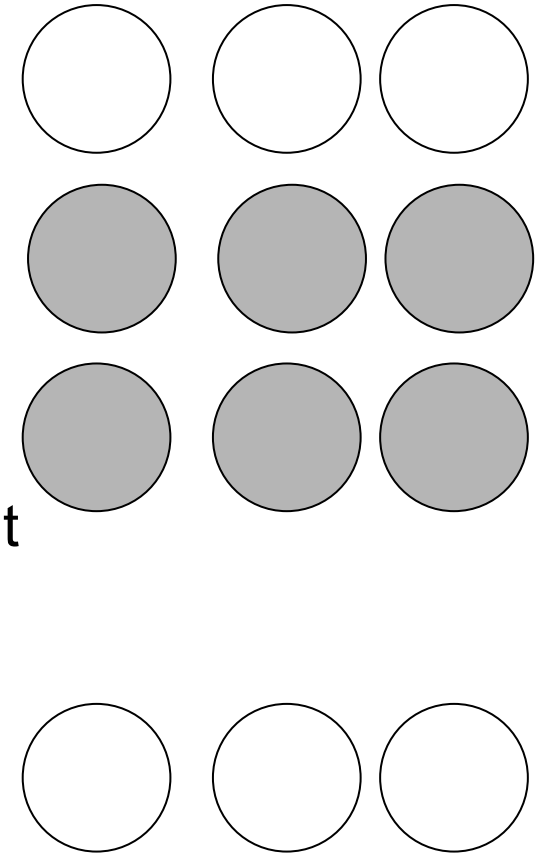
- (actually just one clustering method, there are many…)
- We want to calculate K separate means, or *medoids* that define maximal clusterings of the data
  - *Propose* a centroid for the Kth cluster
  - Calculate goodness-of-fit
  - Repeat for each K
  - Define labellings (clustering) for every point as closest centroid
  - Optimise globally
- **WARNING: not just hard but *NP*-hard!**

Neural networks

- Not actually a big computer brain…
- Rather than attempting to model structure of a system directly, and/or detect partitions, neural networks are simply connected nodes which transform input into output
- Iterative training compares output to error and refines node connections in highly nonlinear ways
- Powerful technique for making predictions and classifications (handwriting/facial pattern recognition etc) but **very** hard to understand internal model logic
- RIRO++

Neural networks

- Define a NN as *layers* of *nodes*:
  - *Input* layer
  - *Output* layer
  - One or more *hidden layers*
  - An *activation function*
- *Train* the NN as follows:
  - Begin with a maximally-connected NN
  - For each node in first layer, calculate output as (input * activation function)
  - For nodes in next layer, sum (inputs *weights), and repeat
  - Calculate error from output layer
  - Backpropagate error to train e.g. refine weights



Genetic algorithms

- Instead of *a priori* model selection rules, allow for a ‘population’ of models, with a pool of possible parameters / methods to choose from
- Allow them to compete for ‘fitness’ (goodness-of-fit) over generations
- Sounds cool
- Can be a lot of work for not noticeably improved models
- Hard to reason why they should be inherently any better

More dimensionality reductions

- Principal components analysis (PCA)
- Multidimensional scaling (MDS), many others
- Different approaches will depend on nature, dimensionality, and heterogeneity in the dataset.

Visualisations

- Good visualisations really will get you noticed
- Ggplot2, Hadley Wickham etc. See Qmplus
- Good visualisations should also highlight structure in the data. They may even help us uncover it.
- Animation too? Why not.

Final thoughts

- Biology needs bioinformatics and stats. Science does. The world needs you.
- Our skills are highly transferable
- Our problems are cool
- There are many different types of bioinformatician and ways of doing bioinformatics
- Good luck!