

Bayesian networks for bioinformatics

An introduction to inference and learning

Dr Chris J. Needham

School of Computing, University of Leeds, UK.

Dr James R. Bradford

Institute of Molecular and Cellular Biology, University of Leeds, UK.

Andrew J. Bulpitt and David R. Westhead

Outline

Bayesian networks provide a neat compact representation for expressing joint probability distributions and for inference. They are becoming increasingly important in biology for inferring cellular networks and pathways, biological data integration and genetics. This tutorial introduces the Bayesian approach to inference and learning parameters and structures for Bayesian networks.

Many applications in computational biology have taken advantage of Bayesian networks or more generally, probabilistic graphical models. These include: protein modelling, systems biology; gene expression analysis, inferring cellular networks and pathway modelling; biological data integration; protein protein interaction and functional annotation; DNA sequence analysis; genetics and phylogeny linkage analysis.

With this growing use of Bayesian networks and Bayesian methodologies, there has been a lack of suitable introductory information about Bayesian networks which is accessible to an audience without significant mathematical and statistical backgrounds.

This tutorial builds on our recent primer (Needham *et al.*, 2006), and is aimed at the multi-disciplinary ISMB audience, both students and researchers, since it will be based around biological examples and begin at an introductory level with numerous examples to demonstrate how to use Bayesian networks. In the second half, the focus will be on the higher level concepts, rather than becoming involved in the complicated mathematics behind the learning methods.

This will provide the audience with an understanding how and why Bayesian networks work, and at a time when they are becoming the machine learning method of choice.

Contents

Introduction	4
Modelling a simple cell signalling pathway	4
Models with continuous variables	7
Learning for Bayesian networks	7
Bayesian learning	9
Learning from incomplete data	10
Structure learning	10
Dynamic Bayesian networks	11
Causality	11
Software	12
Summary: Bayesian networks for computational biology	12
References	13

Introduction

Bayesian networks are a useful tool for statistical modelling. They are increasingly popular in the biological sciences for the tasks of inferring cellular networks (Friedman, 2004), modelling protein signalling pathways (Sachs *et al.*, 2005), data integration, classification, and genetic data analysis (Beaumont and Rannala, 2004). Bayesian networks provide a neat compact representation for expressing joint probability distributions and for inference. The representation and use of probability theory makes Bayesian networks suitable for learning from incomplete datasets, expressing causal relationships, combining domain knowledge and data, and avoid over-fitting a model to training data.

This primer aims to provide an accessible introduction to Bayesian networks for the computational biologist, focusing on the concepts behind methods for learning the parameters and structure of models. It begins with a simple toy example, and then considers the points made above. More in-depth tutorials are provided by Heckerman (1998) and Husmeier *et al.* (2005).

A Bayesian network can be viewed as a collection of probabilistic classification/regression models, organised by conditional-independence relationships. – Heckerman (1998)

Modelling a simple cell signalling pathway

Consider a simple cell signalling example consisting of an outside stimulant, an extracellular signal, an inhibitor to the signal, a G protein coupled receptor, a G protein, and the cellular response. A Bayesian network can be constructed which expresses the relationships between variables. For example:

- The stimulant may or may not generate a signal.
- The concentration of the signal may effect the level of the inhibitor.
- Whether the signal binds with the receptor is dependent upon both the concentration of the signal and the level of the inhibitor.
- The G protein should become active if the receptor binds.
- An active G protein initiates a cascade of reactions that causes the cellular response.

Using this information, which variables depend on which other variables can be identified, and also which variables are conditionally independent. If two variables are independent given the state of a third variable, then they are said to be *conditionally independent*. For example, consider two independent tests for a disease, T_1 and T_2 . The tests are reasonably reliable, and a strong correlation is seen between T_1 and T_2 . If the result of test T_1 is positive, it becomes more likely that T_2 will also be positive. However, if it is known that the person has/hasn't got the disease, then the result of T_1 has no effect on the expected value of T_2 ; they become conditionally independent. The above relationships between the cell signalling variables can be expressed by the graph structure shown in Figure 1; nodes represent variables, and the directed

edges show the dependencies. (Feedback from the cellular response to the concentration of the extracellular signal (or inhibitor) would create a cyclic graph which is discussed later). Consider all the variables to be discrete, and to take the following possible values (and note the abbreviations introduced).

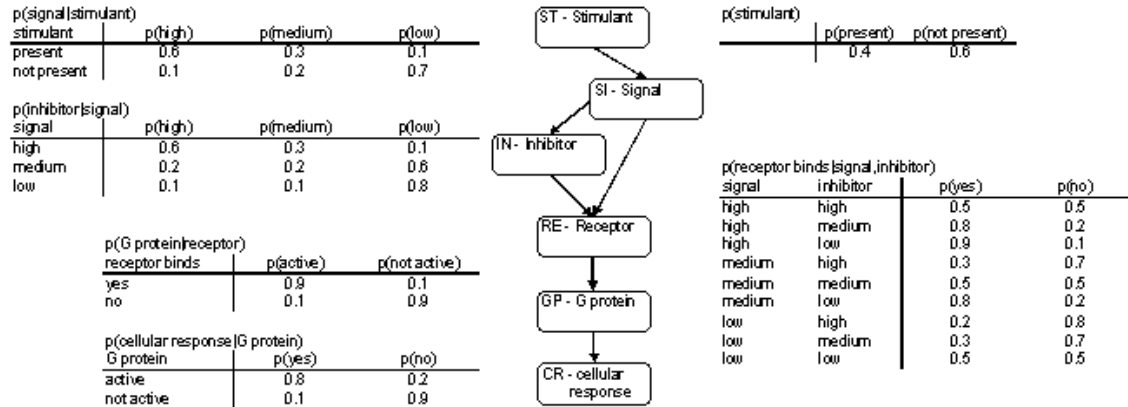


Figure 1: Bayesian network of the cell signalling pathway, and example CPTs

- ST - Stimulant: present/not present
- SI - Signal: high/medium/low
- IN - Inhibitor: high/medium/low
- RE - Receptor binds: yes/no
- GP - G protein: active/not active
- CR - Cellular response: yes/no

A model of the relationships between the variables can be built. In this discrete case, conditional probability tables (CPTs) can be formed to express the probability of the state of each variable given its parents (those it directly depends upon). For example, if the graph structure and CPTs of the Bayesian network are taken to be as defined in Figure 1, then the probability that the signal is high when the stimulant is present, $p(SI = high|ST = present) = 0.6$ and the probability that the receptor binds given that the signal is high and the inhibitor is low, $p(RE = yes|SI = high, IN = low) = 0.9$.

The joint probability distribution $p(ST, SI, IN, RE, GP, CR)$ can be expressed as a product of distributions over a smaller number of variables, through repeated application of the *product rule* of probability calculus

$$p(x, y) = p(x|y)p(y) \quad (1)$$

and by exploiting conditional independence relations described in the graph structure. Applying the product rule, and then conditional independence gives:

$$\begin{aligned} p(ST, SI, IN, RE, GP, CR) &= p(CR|ST, SI, IN, RE, GP)p(ST, SI, IN, RE, GP) \\ &= p(CR|GP)p(ST, SI, IN, RE, GP) \end{aligned}$$

Continuing in this way, the joint probability over all the variables can be expressed as:

$$p(ST, SI, IN, RE, GP, CR) = p(CR|GP)p(GP|RE)p(RE|SI, IN)p(IN|SI)p(SI|ST)p(ST)$$

In the case of Bayesian networks, consisting of a set of n nodes $\mathbf{x} = \{x_1, \dots, x_n\}$ organised in a directed acyclic graph (DAG), where each node x_i has parents $\mathbf{pa}(x_i)$, the joint probability distribution is compactly expressed as:

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i|\mathbf{pa}(x_i)) \quad (2)$$

The ability to express the joint probability in this way (exploiting conditional independencies) provides a concise representation in terms of simple component distributions (factors), thereby reducing the number of parameters to be estimated. In this example, to specify the full joint probability distribution as a conditional probability table would require 72 parameters, whereas by exploiting conditional independence only 24 are required. This may not seem that advantageous, however consider a network with 100 nodes, each taking 3 possible values. If the graph was fully connected, the full probability distribution would require over 10^{47} parameters¹, compared to only needing 1800 parameters if each node had only two parents ($100 \times 18 = 1800$). This demonstrates just how powerful conditional independence can be. Not only is the parameter space smaller, but the parameters are easier for an expert to estimate, since they involve fewer variables. Learning of the parameters from data is discussed below. First, inference in Bayesian networks is illustrated.

What is the probability of the G protein being active, given that the stimulant is present?

Given evidence about the state of a variable, or set of variables, the state of other variables can be inferred. For example, to find the probability that the G protein is active given that it has been observed that a stimulant is present, i.e. to find $p(GP = \text{active}|ST = \text{present})$, it is necessary to marginalise over the unknown parameters. This amounts to summing the probabilities of all routes through the graph, using the *sum rule*:

$$p(x) = \sum_y p(x, y) \quad (3)$$

where $p(x, y)$ may be expanded using the product rule (Equation 1). Thus:

$$\begin{aligned} p(GP = \text{active}|ST = \text{present}) &= \sum_x \sum_y \sum_z p(GP = \text{active}|RE = x) \\ &\quad p(RE = x|IN = y, SI = z)p(IN = y|SI = z) \\ &\quad p(SI = z|ST = \text{present}) \end{aligned}$$

¹In a fully connected directed acyclic graph there must be one node with $0, 1, \dots, n-1$ parents, thus the number of parameters is $\prod_{i=1}^{99} 2 \times 3^i = 2 \times \frac{1-3^{100}}{1-3} = 3^{100} - 1 \approx 5 \times 10^{47}$

which when evaluated with the conditional probabilities in Figure 1 equals 0.592. [$p(GP = active|ST = not\ present) = 0.5048$].

What is the probability the stimulant is present, given that the signal is high?

It is often of interest to calculate posterior probabilities such as the probability that the stimulant is present, given that the signal is high $p(ST = present|SI = high)$ for which *Bayes' rule* may be applied:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \quad (4)$$

Note also: $p(x) = \sum_y p(x|y)p(y)$

Thus:

$$\begin{aligned} p(ST = present|SI = high) &= \frac{p(SI=high|ST=present)p(ST=present)}{p(SI=high|ST=present)p(ST=present) + p(SI=high|ST=notpresent)p(ST=notpresent)} \\ &= \frac{0.6 \times 0.4}{0.6 \times 0.4 + 0.1 \times 0.6} = 0.8 \end{aligned}$$

So within this neat representation of a Bayesian network, inference is easy. Inferences can be made about the value of any variable(s), given evidence about the state of other variable(s). [For example, consider the prior probability that the stimulant is present $p(ST = present) = 0.4$. The inferred probability of the presence of a stimulant is dependent upon evidence about the other variables: $p(ST = present|GP = active) = 0.44$ and $p(ST = present|GP = not\ active) = 0.35$].

Models with continuous variables

For Bayesian networks which use continuous variables, conditional probability densities (CPDs) are used in a similar way to CPTs. Figure 2 presents a simple Bayesian network which introduces the concept of using continuous variables. A continuous node, B , with a discrete parent, A , (say, a variable with k states) could in effect model the continuous data with k Gaussian distributions. Thus given that A is in state a_i the likelihood of a value of B may be inferred, or alternatively, given a value b for variable B , the probability that variable A is in state a_i may be inferred. Linear regression may be used to fit the Gaussians (or other distributions) to the training data in order to minimise the decision error between the classes.

Learning for Bayesian networks

In essence a Bayesian network is used to model a probability distribution \mathbf{X} . A set of model parameters θ may be learned from the data in such a way that maximises the likelihood that the data came from \mathbf{X} . Given a set of observed training data, $D = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ consisting of N examples, it is useful to consider the likelihood of a model, $L(\theta)$, as the likelihood of seeing

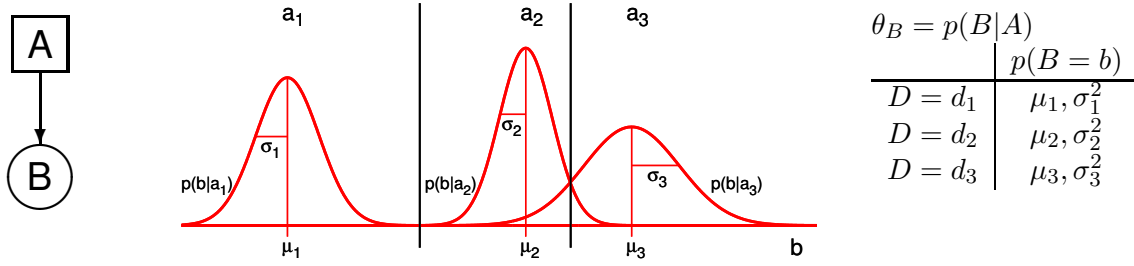


Figure 2: A simple Bayes net with a continuous node B , having a discrete parent A . The usual notation is to use squares for discrete nodes and circles for continuous nodes. θ_B is the parameter set which encodes the model for B in terms of three Gaussians - one for each of the three possible states of A . A mean μ_i and standard deviation σ_i are the parameters for the Gaussian distribution which models $p(b|a_i)$.

the data, given a model:

$$L(\theta) = p(D|\theta) = \prod_{i=1}^N p(\mathbf{x}_i|\theta) \quad (5)$$

In order to infer the likelihood of an example observation, \mathbf{x} , a joint probability of all the variables can be calculated (as in the previous section) as the product of the conditional probability distributions for each variable:

$$p(\mathbf{x}|\theta) = \prod_{i=1}^n p(x_i|\mathbf{pa}(x_i), \theta_i) \quad (6)$$

where $\mathbf{x} = \{x_1, \dots, x_n\}$ are the variables (and nodes in the BN), and the set of model parameters $\theta = \{\theta_1, \dots, \theta_n\}$, where θ_i is the set of parameters describing the distribution for the i th variable x_i which are used in conjunction with the known model structure given by $\mathbf{pa}(x_i)$ - the parents of x_i . Each parameter set θ_i may take a number of forms, commonly a CPT (conditional probability table) is used for discrete variables, and CPDs (such as Gaussian distributions) are used for continuous variables. Classification/regression models can be used to learn the parameters for each node in the network. For the example using CPTs in Figure 1, it is possible to learn the probabilities for these tables. For each node, the probability that the variable will be in each possible state (given its parents' states) could be calculated based on the frequency observed in a set of training data.

It is often useful/necessary to use a prior distribution for the model parameters. For multinomial sampling, a Dirichlet distribution is commonly used as a prior and can be thought of as adding pseudo-counts to the observed frequencies. If the sample size is large, the effect of the prior is small, however it can often be useful to allow a larger pseudo-count for classes with little or very uncertain data, ensuring certain configurations of variables are still possible. Without a prior, a configuration that was not seen in the training examples would be incorrectly assigned a zero probability of being drawn from \mathbf{X} .

The learning paradigm which aims to maximise $L(\theta)$ is called *maximum likelihood* (ML). This approximates the probability of a new example \mathbf{x} given the training data D as $p(\mathbf{x}|D) \approx$

$p(\mathbf{x}|\theta_{ML})$ where θ_{ML} is the maximum (log) likelihood model which aims to maximise $\ln p(D|\theta)$, i.e. $\theta_{ML} = \arg \max_{\theta} \ln p(D|\theta)$. This amounts to maximising the likelihood of the ‘data given model’. ML assumes a uniform prior. In order to consider other prior distributions, a *maximum a posteriori* (MAP) model can be used. This approximates the probability of a new example \mathbf{x} given the training data D as $p(\mathbf{x}|D) \approx p(\mathbf{x}|\theta_{MAP})$ where θ_{MAP} is the maximum a posteriori probability (likelihood of the ‘model given data’) which aims to maximise $\ln p(\theta|D)$, i.e. $\theta_{MAP} = \arg \max_{\theta} \ln p(\theta|D)$. This takes into account the prior, since through Bayes’ theorem: $p(\theta|D) = p(D|\theta)p(\theta)/p(D)$. Both ML and MAP produce a point estimate for θ . One of the powers of *Bayesian statistics* is not producing point estimates but is model averaging, which is considered in the next section.

Bayesian learning

For *Bayesian learning*, the parameters are considered to be latent variables and the key idea is to marginalise over these unknown parameters, rather than to make point estimates (which ML and MAP do). The computation of a full posterior distribution, or model averaging, avoids severe over-fitting and allows direct model comparison. Formulating Bayesian learning as an inference problem, the training examples in D can be considered as N independent observations of the distribution \mathbf{X} (Figure 3).

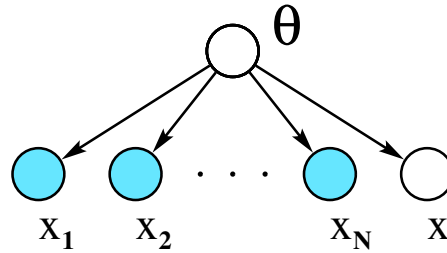


Figure 3: Bayesian learning is an inference problem. The shaded nodes \mathbf{x}_i represent the observed independent training data, \mathbf{x} the incomplete example observation for which the missing values are to be inferred, all of which are dependent upon the model θ .

The joint probability of our training data, the model and a new observation \mathbf{x} is:

$$p(D, \theta, \mathbf{x}) = p(\mathbf{x}|\theta)p(D|\theta)p(\theta) \quad (7)$$

Applying the sum rule (3):

$$p(\mathbf{x}, D) = \int p(D, \theta, \mathbf{x}) d\theta \quad (8)$$

Applying the product rule (1) to the left hand side, and substituting (7) for the joint probability on the right hand side, then dividing both sides by $p(D)$, gives the predictive distribution for \mathbf{x} :

$$p(\mathbf{x}|D) = \frac{1}{p(D)} \int p(\mathbf{x}|\theta)p(D|\theta)p(\theta) d\theta \quad (9)$$

$$= \int p(\mathbf{x}|\theta)p(\theta|D) d\theta \quad (10)$$

i.e. $p(\text{example}|\text{data}) = p(\text{example}|\text{model}) \times p(\text{model}|\text{data})$ over all models

This is computing a full Bayesian posterior. In order to do this, a prior distribution for the model parameters needs to be specified. There are many types of priors which may be used, and much debate about the choice of prior.

Learning from incomplete data

The parameters for Bayesian networks may be learned even when the training data set is incomplete, i.e. the values of some variables in some cases are unknown. Commonly, the Expectation-Maximisation (EM) algorithm is used, which estimates the missing values by computing the expected values and updating parameters using these expected values as if they were observed values.

EM is used to find local maxima for MAP or ML configurations. EM begins with a particular parameter configuration $\hat{\theta}$ (possibly random) and iteratively applies the expectation and maximisation steps, until convergence:

- **E-step.** The expected values of the missing data are inferred to form D_c - the most likely complete dataset given the current model parameter configuration.
- **M-step.** The configuration of $\hat{\theta}$ which maximises $p(\hat{\theta}|D_c)$ is found (for MAP).

Using EM to find a point estimate for the model parameters can be efficient to calculate and gives good results when learning from incomplete data or for network structures with hidden nodes. With large sample sizes the effect of the prior $p(\theta)$ becomes small, and ML is often used instead of MAP in order to simplify the calculation.

A number of sampling methods have been used to estimate the (full) posterior distribution of the model parameters in the presence of incomplete data. Monte Carlo methods, such as *Gibbs sampling*, are extremely accurate, and require lots of computation, often taking a long time to converge, and become intractable as the sample size grows. *Gaussian approximation* is often accurate for relatively large samples, and is more efficient than Monte-Carlo methods. It is based on the fact that the posterior distribution $p(\theta|D)$ which is proportional to $p(D|\theta) \times p(\theta)$ can often be approximated as a Gaussian distribution. With more training data, the Gaussian peak becomes sharper, and tends to the maximum a posteriori configuration θ_{MAP} .

Structure learning

So far, only the learning of parameters of a Bayesian network of known structure has been considered. Sometimes the structure of the network may be unknown and this may also be learnt from training data. One approach to learning structure is to use a *search* to find a 'good' structure. This may be done by starting with an initial network with no connectivity and adding parents to each node, measuring the accuracy of the resulting network at each stage or alternatively an initial guess of the structure may be made and this may then be updated

through modifications such as the addition or removal of edges. This could be achieved through an optimisation process such as simulated annealing.

There are two common approaches used to decide on a 'good' structure. The first is to test whether the conditional independence assertions implied by the structure of the network are satisfied by the data. The second approach is to assess the degree to which the resulting structure explains the data (as described for learning the parameters of the network). In this case a penalty is required to prevent the selection of complex structures as these will have a higher likelihood. For example, using ML without a penalty function would produce a completely connected network, implying no simplification of the factors.

The computation of a full posterior distribution over the parameter space and the model structure space is intractable. *Markov chain Monte Carlo* (MCMC) methods (such as the Metropolis-Hastings algorithm) are used to obtain a set of 'good' sample networks from the posterior distribution $p(\mathcal{S}, \theta | D)$, where \mathcal{S} is a possible model structure. This is particularly useful in the bioinformatics domain, where data D may be sparse and the posterior distribution $p(\mathcal{S}, \theta | D)$ diffuse, and therefore much better represented as averaged over a set of model structures, than through choosing a single model structure.

Dynamic Bayesian networks

An essential feature of many biological systems is feedback. For example, in the simple cell signalling pathway presented at the start of this article, it may be that the strength of the extra-cellular signal is dependent upon the cellular response (once successful, the signal becomes blocked). This would create a feedback loop (cyclic graph). In order to combat this problem, the network may be rolled out in time, to create a *dynamic Bayesian network* where there are connections between time slices and each node is present in each slice. Hidden Markov models (HMMs) are a special case of these.

Dynamic Bayesian networks have been used for inferring genetic regulatory interactions from microarray data. Data from a few dozen time points during a cell cycle is a very small amount of data on which to train a dynamic Bayesian network. Husmeier has recently used MCMC on simulated data of microarray experiments in order to assess the network inference performance with different training set size, priors and sampling strategies (Husmeier *et al.*, 2005).

Causality

Often the really interesting problems involve the learning of causal relationships (Pearl, 2000), such as protein-signalling networks (Sachs *et al.*, 2005). In order to discover the underlying causal model, more than just structure learning is needed, since many network structures are equivalent. (In Markov equivalent network structures the nodes may be dependent upon each other in different ways, but produce the same results). In order to identify a variable which exhibits a causal influence over another variable, particular patterns of dependency of a

third variable must be observed, in the context of interventions (fixing the values of particular variables). This allows the directionality of the causal relation to be determined.

Software

A variety of software is used for Bayesian inference. Three commonly used packages are:

- Bayes Net Toolkit for Matlab (<http://bnt.sourceforge.net/>)
- Probabilistic Network Library (<http://sourceforge.net/projects/openpnl>)
- OpenBUGS (<http://mathstat.helsinki.fi/openbugs/>)

Example code for inference in the cell signalling pathway example in Matlab for use with the Bayes Net Toolkit is available from <http://www.comp.leeds.ac.uk/chrisn/research/cellsig/>


Summary: Bayesian networks for computational biology

Many applications in computational biology have taken advantage of Bayesian networks or more generally, probabilistic graphical models. These include: protein modelling, systems biology; gene expression analysis, networks and pathway modelling; biological data integration; protein protein interaction and functional annotation; DNA sequence analysis; genetics and phylogeny linkage analysis (Beaumont and Rannala, 2004). Bayesian networks and probabilistic graphical models use results from graph theory which allow lucid expression of probability theory. Bayesian networks coupled with Bayesian learning provide a robust framework in which to combine domain knowledge and data, in order to make inferences about states of unknown variables. Learning in Bayesian networks may use a point estimate of the parameters, or use Bayesian statistics to average over possible model structures and parameters to provide an estimate of the posterior distribution of the variables, which avoids over-fitting to the data, which may be noisy, limited, incomplete and uncertain.


References

- Beaumont MA and Rannala B (2004) The Bayesian revolution in genetics. *Nature Reviews Genetics*, 5, 251–261.
- Biedermann A and Taroni F (in press, 2005) A probabilistic approach to the joint evaluation of firearm evidence and gunshot residues. *Forensic Science International*.
- Burnside ES (2005) Bayesian networks: Computer-assisted diagnosis support in radiology. *Academic Radiology*, 12, 422–430.
- Friedman N (2004) Inferring cellular networks using probabilistic graphical models. *Science*, 303, 799–805.
- Heckerman D (1998) A tutorial on learning with Bayesian networks. In Jordan MI, ed., *Learning in Graphical Models*. Kluwer Academic, 301–354.
- Husmeier D, Dybowski R, and (Eds) SR (2005) *Probabilistic Modeling in Bioinformatics and Medical Informatics*. Springer.
- Imoto S, Higuchi T, Goto H, Tashiro K, Kuhara S, and Miyano S (2003) Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. In *IEEE Computational Systems Bioinformatics (CSB'03)*.
- Murphy KP and Mian S (1999) Modelling gene expression data using dynamic Bayesian networks. Tech. rep., Dept. of Computer Science, University of California, Berkeley, CA.
- Needham CJ, Bradford JR, Bulpitt AJ, and Westhead DR (2006) Inference in Bayesian networks. *Nature Biotechnology*, 24, 51–53.
- Pearl J (2000) *Causality: models, reasoning and inference*. Cambridge.
- Sachs K, Perez O, Pe'er D, Lauffenburger DA, and Nolan GP (2005) Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308, 523–529.
- Troyanskaya OG, Dolinski K, Owen AB, Altman RB, and Botstein D (2003) A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *saccharomyces cerevisiae*). *PNAS*, 100, 8348–8353.

14th Annual International Conference On Intelligent Systems For Molecular Biology



ISMB 2006 Fortaleza, Brazil
August 6-10, 2006
and 2nd Annual AB3C Conference: X-Meeting



Bayesian networks for bioinformatics

An introduction to inference and learning


Dr Chris Needham
Computing
The University of Leeds, UK
chrism@comp.leeds.ac.uk

Dr James Bradford
Cellular and Molecular Biology
The University of Leeds, UK
j.r.bradford@leeds.ac.uk

ISMB 2006, Fortaleza, Brazil, August 6-10, 2006

Timetable

Time	Topic
45 mins	Introduction to Bayesian statistics
40 mins	Bayesian networks: representation and inference
35 mins	Learning from data
30 mins	coffee
30 mins	More advanced concepts
30 mins	Examples section
30 mins	Discussion




Fortaleza, Brazil
August 6-10, 2006

2

Introduction to Bayesian statistics

- Principles of learning from data
- Other machine learning approaches
- Probability: Classical vs Bayesian
- Probability theory
- Bayesian inference




Fortaleza, Brazil
August 6-10, 2006

3

Classification

- “Classification is hard” —Janet Thornton, ISMB05
- Don’t just want machine learning methods that classify well...
- ...we want to form an interpretable model.
- Yes/No is not enough, need to know why the decision was made.



Fortaleza, Brazil
August 6-10, 2006

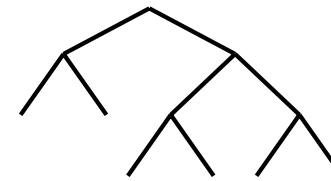
4

What is machine learning?

- Why do we want to learn from data?
- What problems can we tackle?

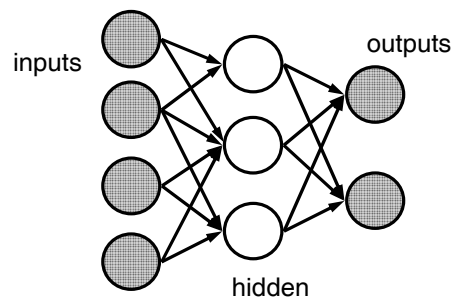
Decision Trees

- Gini index: $i(N) = 1 - \sum_j P^2(w_j)$
- 20 questions?



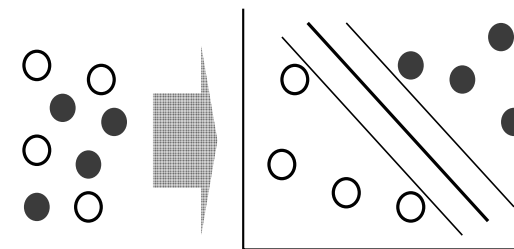
Neural networks

- 'Black box'



SVMs

Data x_i is transformed by a non-linear mapping $\phi(\cdot)$ to a high dimensional space where the $y_i = \phi(x_i)$ are linearly separable.



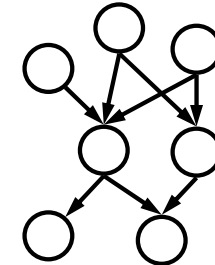
Drawbacks

- Many of you use machine learning algorithms...

What's wrong with them?

Bayesian networks

- A framework for explaining causal relationships consisting of a set of variables connected by a set of directed edges
- Probability calculus is used to describe the probabilistic relationship of each variable with its parents



Bayesian networks

- Combine domain knowledge and data
- Avoid over-fitting of data
- Handle incomplete datasets
- Allow learning about causal relationships

first some probability theory...

Bayesian Probability

- A classical probability is a physical property of the world
- The Bayesian probability of an event X is a person's degree of belief in that event
- Important difference: Do not need repeated trials in order to assign a Bayesian probability
- What is the probability that Brazil will win the world cup in 2010 ?

Probability assignment

- Probability assessment is the process of measuring a degree of belief and can be done in a number of ways:
 - probability wheel
 - ball drawing gambles

Your boss will give you an extra \$1000 if you:

A – Write 3 Journal papers this year

B – Choose a red marble from a bag of 100 marbles, with n red marbles

At what n would event A and B be equally likely?

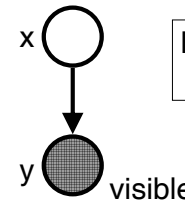


13

Probability Calculus

Product rule: $p(x,y) = p(y|x)p(x)$

Sum rule: $p(x) = \sum_y p(x,y)$



$$\text{Bayes' rule: } p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

$$p(x|y) = \frac{p(y|x)p(x)}{\sum_x p(y|x)p(x)}$$



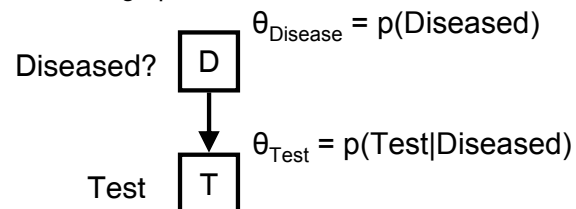
useful because often $p(y|x)$ easy to find, whereas $p(x|y)$ hard to assess

14

Have you got a disease?

- You've tested positive for a disease!
- What is the probability you have the disease?
- It depends on accuracy and sensitivity of the test and background (prior) probability of the disease.

In our probabilistic graphical models notation:



15

Let $p(\text{test is positive} \mid \text{you have the disease}) = 0.95$
 Suppose false positive rate is 5%: $p(T=\text{pos} \mid D=\text{false}) = 0.05$
 and 1% of population have the disease, $p(D=\text{true}) = 0.01$

Diseased? D		$\theta_{\text{Disease}} = p(D)$		T	F
				0.01	0.99
Test T		$\theta_{\text{Test}} = p(T D)$		pos	neg
				T	F
				0.95	0.05
				0.05	0.95

$$\begin{aligned}
 P(D=\text{true} \mid T=\text{pos}) &= \frac{P(T=\text{pos} \mid D=\text{true}) * P(D=\text{true})}{P(T=\text{pos} \mid D=\text{true}) * P(D=\text{true}) + P(T=\text{pos} \mid D=\text{false}) * P(D=\text{false})} \quad (\text{Bayes' Rule}) \\
 &= 0.95 * 0.01 / (0.95 * 0.01 + 0.05 * 0.99) = 0.0095 / 0.0590 = \mathbf{0.161}
 \end{aligned}$$



16

Disease example (cont.)

- So probability of having the disease given you have tested positive is 16%
- Low?
- Of 100 people, we expect only 1 of them to have the disease, but we expect 5 to test positive (5%)
- So, of the 6 people who tested positive, we only expect 1 of them to actually have the disease. Indeed $1/6 \approx 0.16$
- [Using multiple independent tests increases the posterior probability]

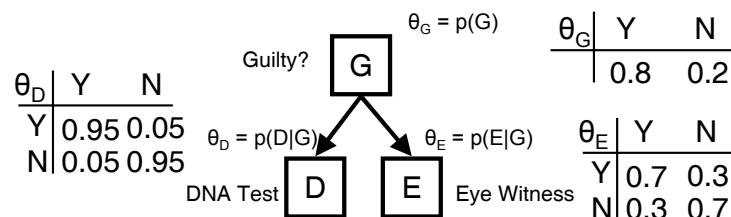
Guilty or not guilty?

- After opening statements, a jury believes there is an 80% probability that a suspect may be guilty
- Two pieces of evidence are presented, an eye witness report and a DNA test result
- Prior studies have shown that the reliability of an eye-witness report is 70% and the DNA test 95%
- The eye witness has identified the subject as the guilty party, while the DNA test indicates the suspect is innocent
- What would be the revised probability that the jury should believe the suspect to be guilty?

$$p(G,D,E) = p(G|D,E)p(D,E) \quad \text{--product rule}$$

$$p(G,D,E) = p(G)p(D|G)p(E|G) \quad \text{--from graph/CI*}$$

$$\Rightarrow p(G|D,E) = p(D|G)p(E|G)p(G)/p(D)p(E)$$



What's the probability of guilty, given witness says guilty, and DNA not guilty?

$$p(G=Y|D=N,E=Y) = p(D=N|G=Y)p(E=Y|G=Y)p(G=Y) / p(D=N)p(E=Y)$$

$$= 0.05 * 0.7 * 0.8 / (0.05*0.8 + 0.95*0.2) * (0.7*0.8 + 0.3*0.2)$$

$$= 0.028 / (0.23*0.62)$$

$$= 0.196$$

$p(\text{guilty} | \text{evidence}) = 20\%$

Summary

- Discussed some methods of machine learning and their limitations
- Introduced graphical models and probability theory
- Made lots of promises about Bayes nets

Bayesian networks: representation and inference

- Joint probability distributions
- Bayesian networks
- Conditional independence
- Compact representation
- Conditional probability distributions
- Inference in Bayesian networks
- Calculating posterior probabilities

Joint Probability Distributions

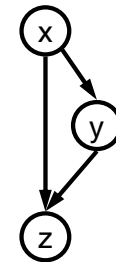
- Given a set of n variables, $X = \{x_1, \dots, x_n\}$, we want to form the joint probability distribution $p(X) = p(x_1, \dots, x_n)$
- Using this we can capture the relationships between sets of variables
- And perform inference of unknown values, such as $p(x_i | x_j, x_k)$

What are Bayesian networks?

- Bayesian networks encode the probabilistic relationships between variables
- Nodes represent variables $X = \{x_1, \dots, x_n\}$
- Edges represent relationships between variables
- A directed acyclic graph (DAG) is formed

Joint Probability Distributions

- Consider $p(x, y, z)$
- By successive application of the product rule:
- $$p(x, y, z) = p(y, z | x) p(x)$$
$$= p(z | x, y) p(y | x) p(x)$$



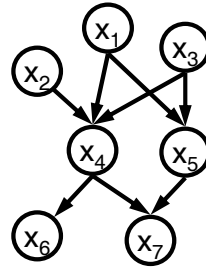
Joint Probability Distributions

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i | pa_i)$$

where pa_i are the parents of x_i

DAG: directed acyclic graph

$$p(x_1, \dots, x_7) = p(x_1)p(x_2)p(x_3) \\ p(x_4|x_1, x_2, x_3) \\ p(x_5|x_1, x_3) \\ p(x_6|x_4)p(x_7|x_4, x_5)$$



Conditional Independence

- If two variables are independent given the state of a third variable, then they are said to be *conditionally independent*

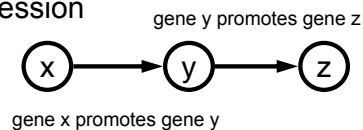
$$p(x, y | z) = p(x | z)p(y | z)$$

- Conditional Independence in Bayesian networks allows us to find variables that are independent and make the models of manageable size.

Serial connections

- Evidence transmitted unless state of variable in connection is known

e.g. Gene expression



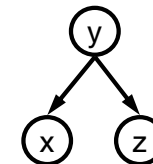
Y unknown: evidence of level of x effects level of z

Y known: the level of z depends only on y, and is conditionally independent of x

Diverging connections

- Evidence transmitted unless connection is instantiated

e.g. Transcription factor Y turns two genes X and Z on

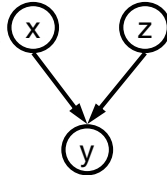


Y unknown: evidence gene x is on effects state of z

Y known: the state of z depends only on y, and is conditionally independent of x

Converging connections

- Evidence transmitted only if variable in connection or one of its children receives evidence



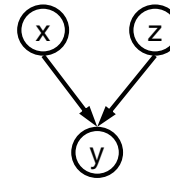
Genes X and Z promote gene Y

Y unknown: evidence of expression level of gene x does not help to infer the expression level of z
 -- *x and z are conditionally independent*

Y known: evidence of expression level of gene x does help to infer the expression level of z

$$p(x,z|y) \neq p(x|y)p(z|y)$$

Converging connections (example)



$$p(x,y,z) = p(y|x,z)p(x)p(z)$$

$$p(x,z) = p(x)p(z)$$

$$p(x,z|y) \neq p(x|y)p(z|y)$$

- Pixel colour in an image
 x = lighting colour, y = image colour, z = surface colour

Simple cell signalling pathway

- Consider a simple example consisting of :
 - a **stimulant**,
 - an extracellular **signal**,
 - an **inhibitor** of the signal,
 - a G protein-coupled **receptor**,
 - a **G protein** and the **cellular response**.

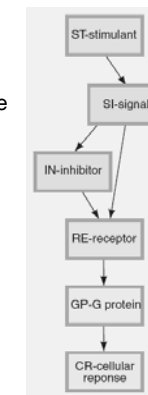
How do you model this pathway?

Prior knowledge

A Bayesian network can be constructed that expresses the relationships between variables

the concentration of the **signal** may affect the level of the **inhibitor**

the **G protein** should become active if the **receptor** binds



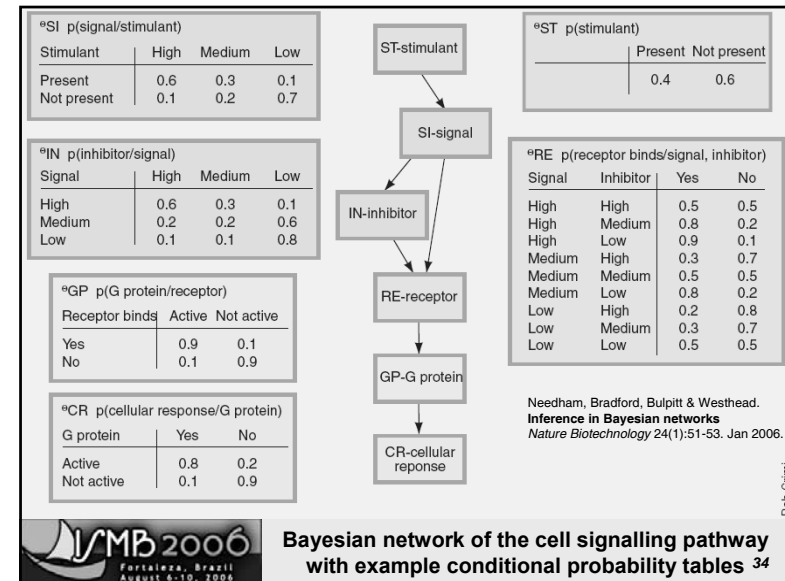
the **stimulant** may or may not generate a **signal**

whether the **signal** binds with the **receptor** depends on the concentrations of both the **signal** and the **inhibitor**

an active **G protein** initiates a cascade of reactions that causes the **cellular response**

Conditional probabilities

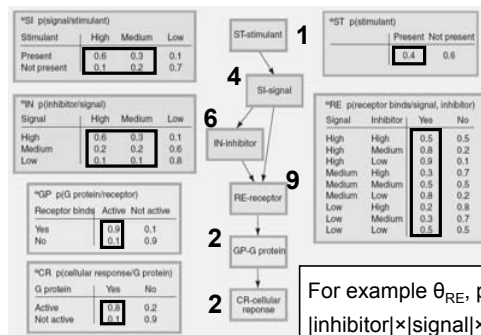
- Now we have a network structure, we need to know the conditional probability distributions θ_i
- These are much easier to specify, since they involve fewer variables and don't involve estimating posterior probabilities.
- For example we only need to know
 - $p(\text{G protein is active} \mid \text{receptor binds})$
- rather than
 - $p(\text{G protein is active} \mid \text{receptor binds, inhibitor is high, signal is medium, stimulant is not present})$
- We can learn these from data (next section)



Compactly expressing the JPD

$$p(\text{ST}, \text{SI}, \text{IN}, \text{RE}, \text{GP}, \text{CR}) = p(\text{CR}|\text{GP})p(\text{GP}|\text{RE})p(\text{RE}|\text{SI}, \text{IN})p(\text{IN}|\text{SI})p(\text{SI}|\text{ST})p(\text{ST})$$

full JPD has $(2 \times 3 \times 3 \times 2 \times 2 \times 2) - 1 = 143$ free parameters



Bayes Net JPD has
24 free parameters
(1+4+6+9+2+2)

free parameters for each
CPD $p(x|y_1, \dots, y_n)$ are:

$$|y_1| \times \dots \times |y_n| \times (|x| - 1)$$

For example $\theta_{\text{RE}}, p(\text{receptor} \mid \text{inhibitor}, \text{signal})$
 $|\text{inhibitor}| \times |\text{signal}| \times (|\text{receptor}| - 1) = 3 \times 3 \times 1 = 9$

How much smaller is the model?

$p(\text{G protein active} \mid \text{stimulant present})$

- Using the sum rule $p(x) = \sum_y p(x, y)$ and the product rule $p(x, y) = p(x|y)p(y)$ we marginalise over the unknown variables:

$$\begin{aligned}
 p(\text{GP} = \text{active} \mid \text{ST} = \text{present}) &= \sum_x \sum_y \sum_z p(\text{GP} = \text{active} \mid \text{RE} = x) p(\text{RE} = x \mid \text{IN} = y, \text{SI} = z) \\
 &\quad p(\text{IN} = y \mid \text{SI} = z) p(\text{SI} = z \mid \text{ST} = \text{present}) \\
 &= 0.592
 \end{aligned}$$

$$p(\text{GP} = \text{active} \mid \text{ST} = \text{not present}) = 0.5048$$

Inference in Bayesian networks

p(stimulant present | signal high)

- It's often of interest to calculate posterior probabilities – we use *Bayes' rule*

p(ST = present | SI = high) =

$$\frac{p(SI = high | ST = present)p(ST = present)}{p(SI = high | ST = present)p(ST = present) + p(SI = high | ST = absent)p(ST = absent)}$$

$$= \frac{0.6 \times 0.4}{(0.6 \times 0.4) + (0.1 \times 0.6)} = 0.8$$

Bayes' rule: $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$



Calculating posterior probabilities 37

Why have you come to ISMB?

- What factors influence people's decision to attend ISMB (or not)?

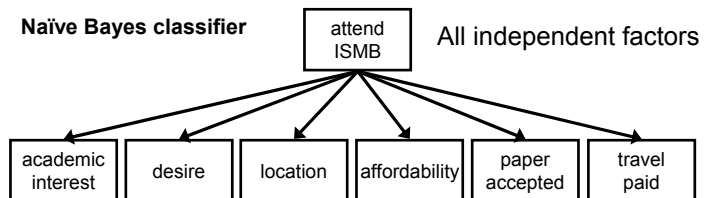
Academic interest	Location
Desire (beach?)	Cost/Affordability
Travel paid	Paper in

- Which of these are independent
- Can we draw a Bayesian network?

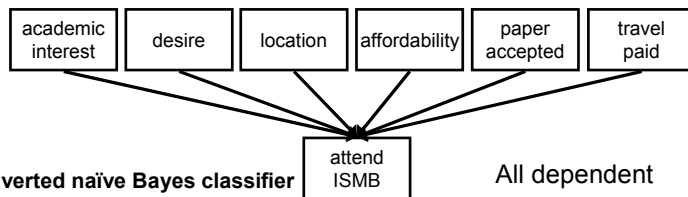


38

Naïve Bayes classifier

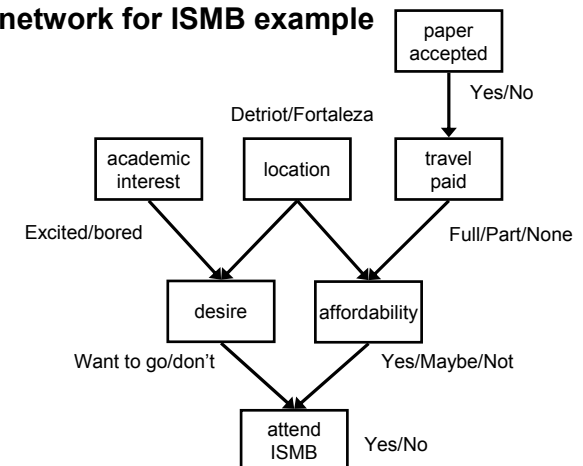


Inverted naïve Bayes classifier



39

Bayesian network for ISMB example



40

Summary

- Joint probability distributions
- Basic concepts of Bayesian networks
- Representation
- Inference
- Incorporating prior knowledge

Learning from data

- Learning model parameters from data
- Parameter priors
- Continuous variables as well as discrete
- Point estimates: maximum likelihood (ML), maximum *a posteriori* (MAP) estimates
- Bayesian learning – model averaging

Probabilistic terminology

- Prior $p(\theta)$
 - the prior probability assigned to a parameter, or to an event, in advance of any empirical evidence
- Posterior $p(\theta|D)$
 - the probability assigned to a parameter, or to an event, on the basis of its observed frequency in a sample, calculated from a prior probability by Bayes' rule
- Data set $D = \{x_n\}$, $n = 1, \dots, N$

Models with discrete variables

$p(a,b)$	b_1	b_2	b_3
a_1	1/9	1/9	1/9
a_2	1/9	1/9	1/9
a_3	1/9	1/9	1/9

Prior model parameters θ

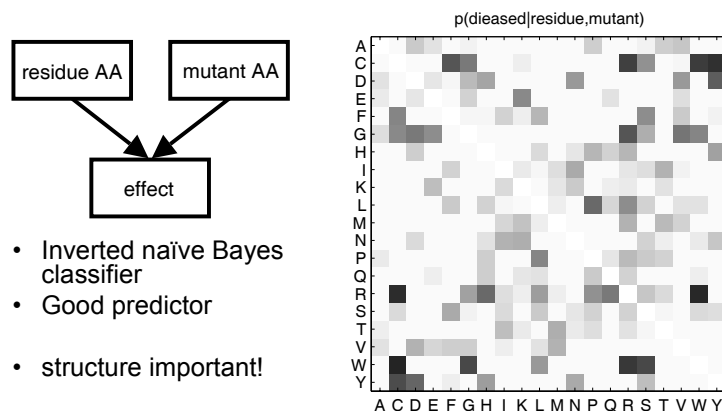
prior (model no data) often uniform uninformative
Dirichlet priors are used.

$p(a,b)$	b_1	b_2	b_3
a_1	0.30	0.10	0.01
a_2	0.02	0.05	0.15
a_3	0.10	0.24	0.03

Posterior model parameters θ

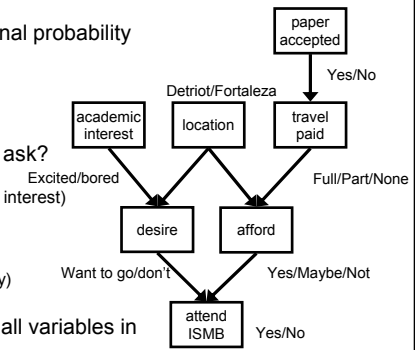
posterior (model given data) can be formed from pseudo-counts of observed frequencies in example training data

SNP prediction from Amino Acids



Example (ISMB BNet revisited)

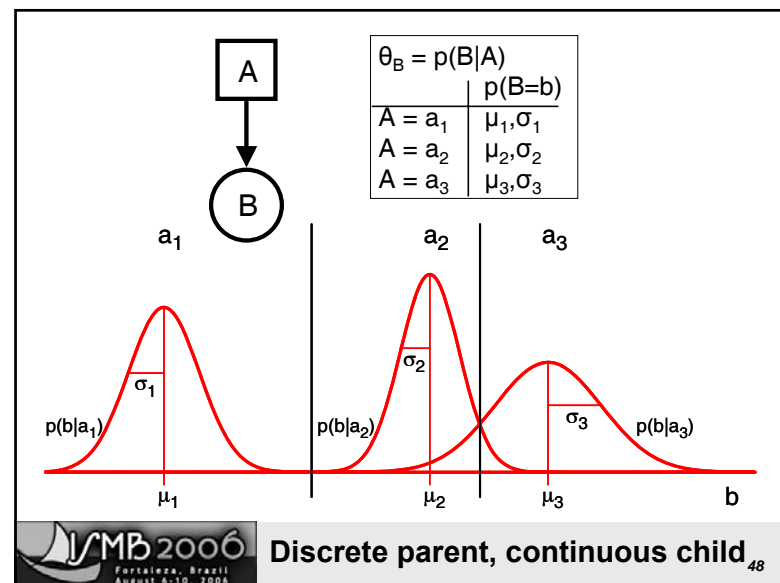
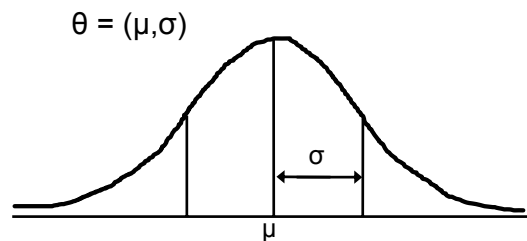
- How can we construct conditional probability tables for this example?
 - Using frequency counts
- How does it work?



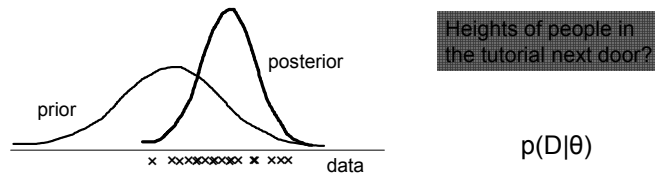
- What questions do we need to ask?
 - $p(\text{travel paid} | \text{paper accepted})$
 - $p(\text{desire to go} | \text{location, academic interest})$
 - $p(\text{academic interest})$
 - $p(\text{paper accepted})$
 - $p(\text{affordability} | \text{location, travel paid})$
 - $p(\text{attend ISMB} | \text{desire, affordability})$
- Compare to building JPD over all variables in this way!
- Later we will consider how to do this in the presence of incomplete data

Continuous data

- prior $p(\theta)$ – estimate of model parameters



Learning model parameters



- How do we fit a model to data?
- Do we measure how well the data fits the model? or how well the model fits the data?
- Given training data, how do we predict a new example?

Maximum Likelihood estimate

- Likelihood function (for independent observations)

$$L(\theta) = p(D|\theta) = \prod_{i=1}^N p(x_i|\theta)$$

- θ_{ML} is maximum likelihood model parameters

$$\theta_{ML} = \arg \max_{\theta} \ln p(D|\theta)$$

(data given model)

- Predictive distribution

$$p(x|D) \approx p(x|\theta_{ML})$$

MAP estimate

- θ_{MAP} is maximum posterior model parameters

$$\theta_{MAP} = \arg \max_{\theta} \ln p(\theta|D)$$

(model given data) $p(\theta|D) = p(D|\theta)p(\theta)$

- Predictive distribution

$$p(x|D) \approx p(x|\theta_{MAP})$$

Bayesian learning paradigm

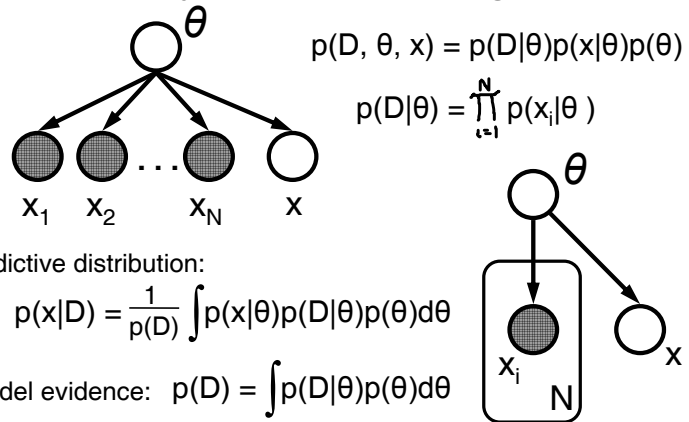
Predictive distribution

$$p(x|D) = \int p(x|\theta)p(\theta|D)d\theta$$

(x given model)(model given data)(over all models)

- Key idea is to marginalize over unknown parameters, rather than make point estimates
 - avoids the over-fitting of ML and MAP
 - allows direct model comparison
- Parameters are now latent variables
- Bayesian learning is an inference problem!

Bayesian Learning



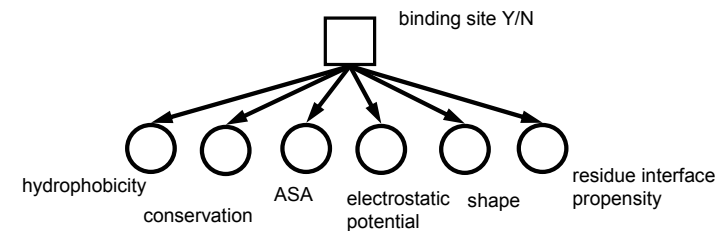
Summary

- Methods for learning model parameters
- Benefits of Bayesian learning
- Avoids over-fitting

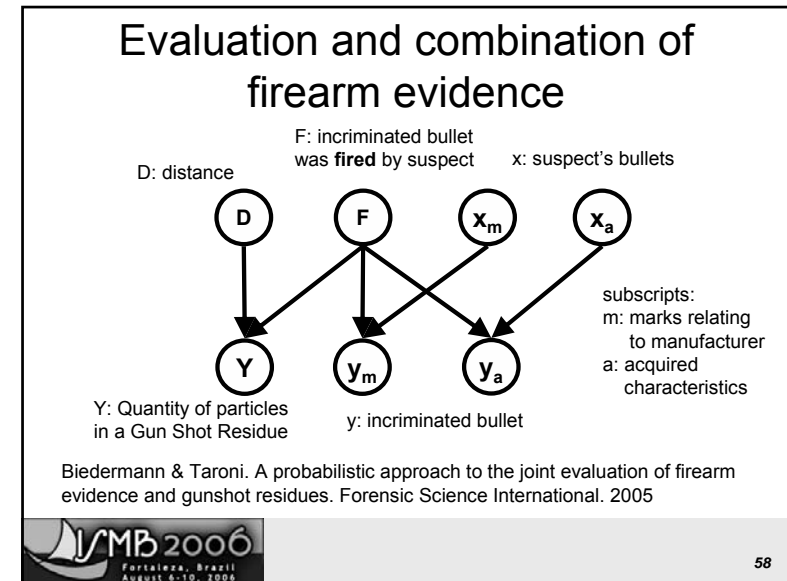
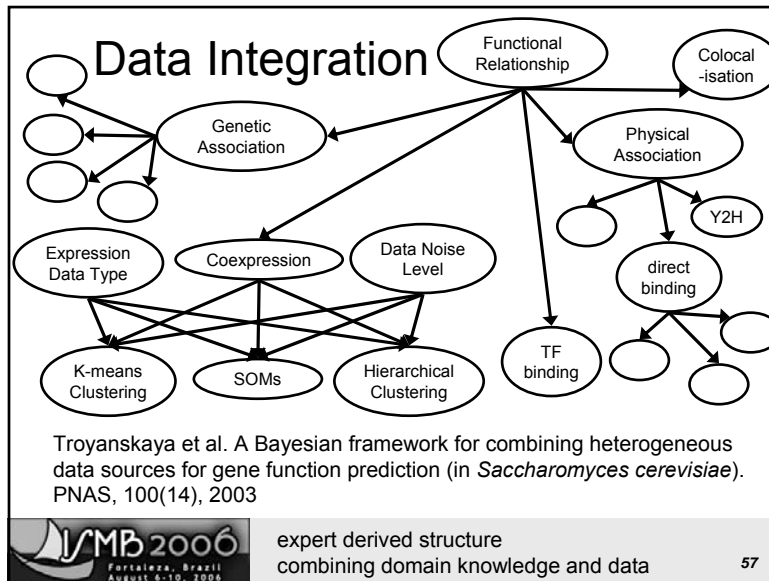
Review of Bayes Nets

- Binding site prediction
- Data integration for gene function prediction
- Evaluation of firearm evidence
- Medical decisions
- Gene cluster analysis

Binding site prediction



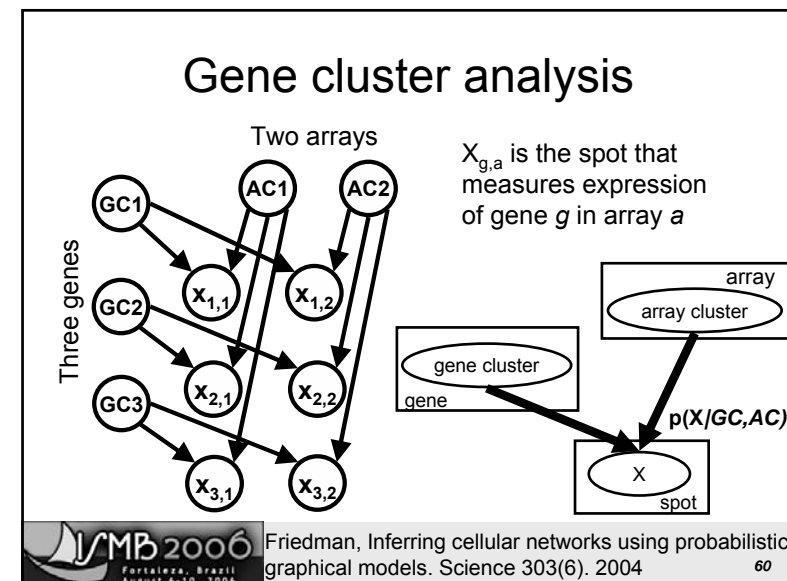
- Naïve Bayes classifier is excellent predictor of binding site patches on protein surfaces.



Medical Decisions

- Radiologists have an overwhelming task of integrating over a breadth of relevant and diverse data
- Breast disease diagnosis factors:
 - age, HRT, family history, calcifications (in a variety of patterns), mass attributes, asymmetric densities

Burnside. Bayesian Networks: Computer-assisted diagnosis Support in Radiology. Academic Radiology 12(4). 2005



More advanced concepts

- Learning from incomplete data
- Markov chain Monte Carlo methods
- Structure learning
- Dynamic Bayesian Networks
- Hidden Markov Models
- Latent variables
- Causality

Learning from incomplete data

- Parameters can be learned even when some variables are unknown in some cases
- Commonly the Expectation-Maximisation algorithm is used.

EM estimates the missing values by computing the expected values and updating the parameters using these expected values as if they were observed values

The EM algorithm

- EM finds local maxima for MAP or ML
- Starts with $\hat{\theta}$, a parameter configuration (random)
- Iteratively applies the expectation and maximisation steps until convergence
- **E-step.** The expected values of the missing data are inferred to form D_c – the most likely complete dataset given the current model parameters
- **M-step.** The configuration of $\hat{\theta}$ which maximises $p(\hat{\theta} | D_c)$ is found (for MAP)

Sampling methods

- Sampling methods have been used to estimate the full posterior distribution of the model parameters in the presence of incomplete data
- Monte Carlo methods such as **Gibbs sampling** are extremely accurate (but require lots of computation, take a long time to converge and become intractable as the sample size grows)
- **Gaussian approximation** is based on the fact that $p(\theta|D) \propto p(D|\theta)p(\theta)$ can be approximated as Gaussian distribution. With more training data the Gaussian peak becomes sharper $\rightarrow \theta_{MAP}$

Structure learning

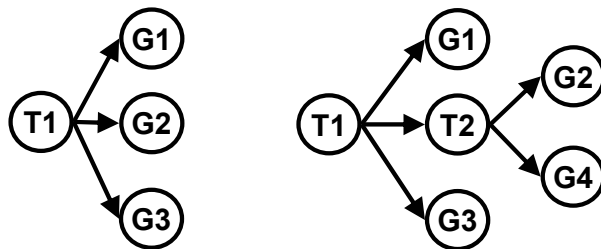
- We've seen that we can combine knowledge about the domain with data
 - i.e. get an expert to design a network structure based on known relationships/ independencies between the variables
- We can also learn the structure of the model!
 - search for good structures which capture the interactions between the variables, whilst maintaining a compact model

Structure Learning

- Greedy search
 - Iteratively: add, reverse or delete an edge
 - Score the structure S^h
- Score functions
 - Full Bayesian posterior
 - BIC score function

$$\ln p(D|S^h) \approx \ln p(D|\theta_s^{ML}, S^h) - \frac{1}{2} d \ln N$$

Learning Cellular Networks



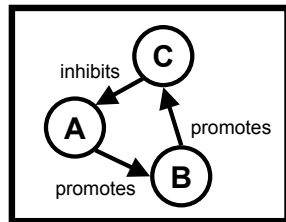
Review article: Friedman, Inferring cellular networks using probabilistic graphical models. Science 303(6). 2004

Inferring genetic networks

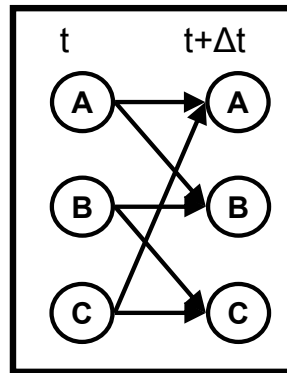
- Constructing a genetic network from microarray gene expression data by using a Bayesian network.
 - a gene corresponds to a node (random variable)
 - gene regulations are shown by directed edges
 - gene interactions are modelled by the conditional distribution of each gene
- Incorporate prior knowledge from protein-protein interactions, protein-DNA interactions, gene networks and literature
- Analysis of *Saccharomyces cerevisiae* gene expression data newly obtained by disrupting 100 genes, mainly transcription factors.

Dynamic Bayesian networks (DBNs)

- Expression levels of genes A, B, C



Static model – not a BN



Dynamic Bayesian network



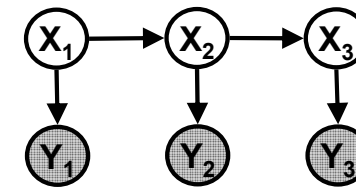
Murphy & Mian. Modelling Gene Expression Data using Dynamic Bayesian Networks. Tech Report. 1999.

69

Modelling the state of variable X , as a Markov process, with a DBN:



Hidden Markov Models (HMMs)



HMMs can be represented as Dynamic Bayesian networks, with hidden variables.

t doesn't have to be time
HMMs are often used for sequence alignment, where hidden state is INSERT, DELETE, or MATCH, and t is the next position in the sequence.

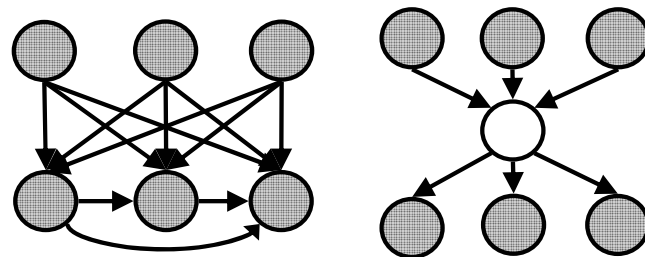
White nodes unobserved. Shaded nodes observed.



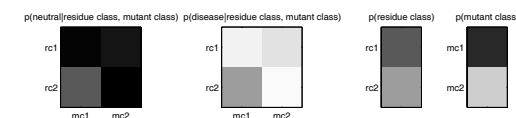
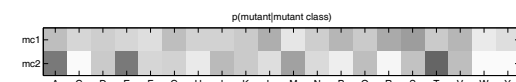
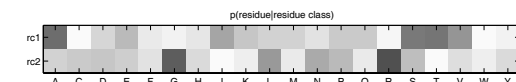
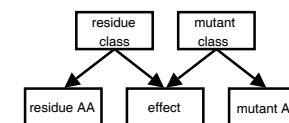
70

Latent (hidden) variables

- Latent variables can be added to models to capture additional information or reduce model size through expert knowledge



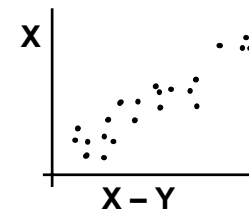
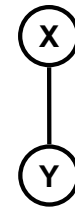
71



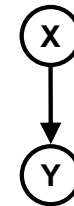
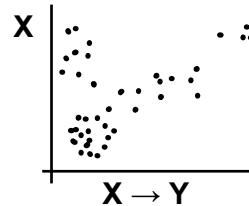
A latent variable model 72

Causality

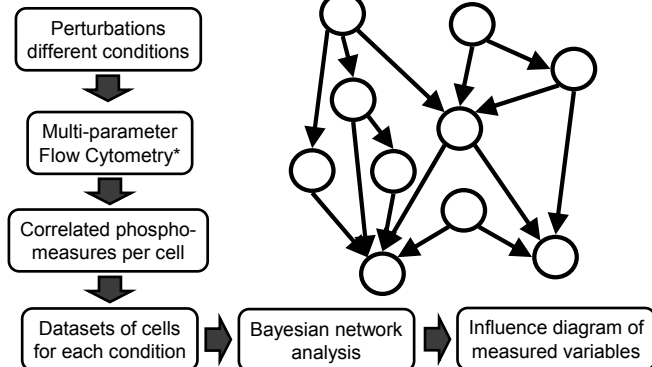
- The learning of causal relations is somewhat trickier
- We'd like to determine what effects what, rather than just what's related
- To do this requires the combination of expert knowledge, and interventions



No inhibition
X inhibited
Y inhibited



Causal protein-signalling networks



* measures 11 phosphoproteins and phospholipids in individual cells in each perturbation

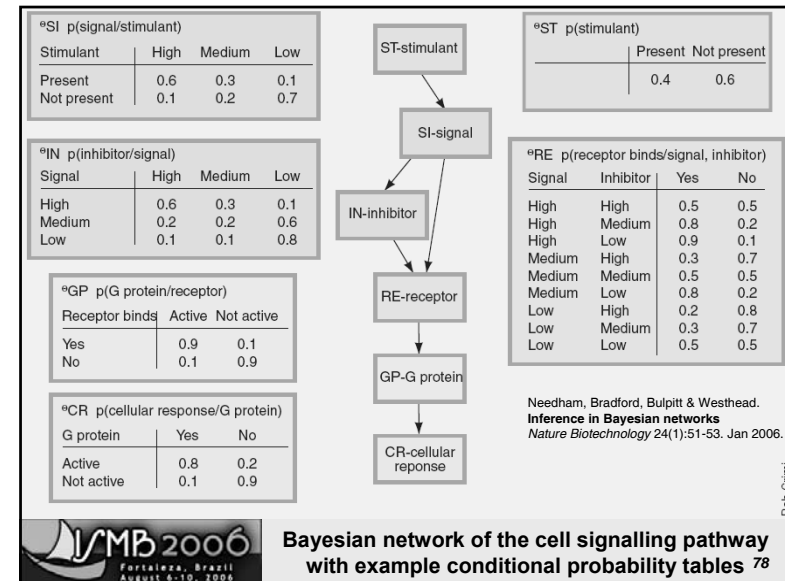
Sachs *et al.* Causal protein signalling networks derived from multi-parameter single-cell data. Science 308(5721) 2005:75

Summary

- Handling incomplete data
- Structure learning
- Learning causal relationships

Examples section

- The simple cell signalling example from earlier, example in Matlab
- An application of Bayesian networks to learning parameters and structures from data for predicting functional consequences of missense mutations



Cell signalling pathway example in Matlab

We have 6 discrete nodes

```
N = 6;
discrete_nodes = 1:N;
```

We will name them for ease of use

```
ST = 1; SI = 2; IN = 3; RE = 4; GP = 5; CR = 6;
```

And construct a DAG

```
dag = zeros(N,N);
dag(ST,SI) = 1;
dag(SI,[IN,RE]) = 1;
dag(IN,RE) = 1;
dag(RE,GP) = 1;
dag(GP,CR) = 1;
dag
```

```
dag =
0 1 0 0 0 0
0 0 1 1 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
0 0 0 0 0 0
```

Define the nodes sizes (SI and IN each take 3 values: high, med, low)

```
node_sizes = [2 3 3 2 2 2];
```

Initialise the Bayesian Network

```
bnet = mk_bnet(dag, node_sizes, 'discrete', discrete_nodes,
'names', {'Stimulant','Signal','Inhibitor','Receptor','G protein','Cell Res'});
```

Define the Conditional Probability Tables

```
bnet.CPD{ST} = tabular_CPD(bnet, ST, [0.4 0.6]);
bnet.CPD{SI} = tabular_CPD(bnet, SI, [0.6 0.1 0.3 0.2 0.1 0.7]);
bnet.CPD{IN} = tabular_CPD(bnet, IN, [0.6 0.2 0.1 0.3 0.2 0.1 0.1 0.6 0.8]);
bnet.CPD{RE} = tabular_CPD(bnet, RE, [0.5 0.3 0.2 0.8 0.5 0.3 0.9 0.8 0.5
0.5 0.7 0.8 0.2 0.5 0.7 0.1 0.2 0.5]);
bnet.CPD{GP} = tabular_CPD(bnet, GP, [0.9 0.1 0.1 0.9]);
bnet.CPD{CR} = tabular_CPD(bnet, CR, [0.8 0.1 0.2 0.9]);
```

We choose the Junction Tree algorithm for Inference

```
engine = jtree_inf_engine(bnet);
```

The CPT for p(stimulant)

```
ans = 0.4000
      0.6000
```

The CPT for p(signal|stimulant)

```
ans = 0.6000 0.3000 0.1000
      0.1000 0.2000 0.7000
```

The CPT for p(inhibitor|signal)

```
ans = 0.6000 0.3000 0.1000
      0.2000 0.2000 0.6000
      0.1000 0.1000 0.8000
```

The CPT for p(G protein|receptor)

```
ans = 0.9000 0.1000
      0.1000 0.9000
```

The CPT for p(receptor|inhibitor,signal)

```
ans(:,1) = 0.5000 0.8000 0.9000
           0.3000 0.5000 0.8000
           0.2000 0.3000 0.5000
```

```
ans(:,2) = 0.5000 0.2000 0.1000
           0.7000 0.5000 0.2000
           0.8000 0.7000 0.5000
```

the first table above shows the conditional probabilities when the receptor binds, and the second when the receptor does not bind

The CPT for p(cell res|G protein)

```
ans = 0.8000 0.2000
      0.1000 0.9000
```



81

Now we can make inferences! e.g. What is p(G protein|Stimulant=present) ?

We set the evidence to nothing (a blank cell array)

```
evidence = cell(1,N);
```

We add to evidence that ST was present (1)

```
evidence{ST} = 1;
```

We pass this evidence to the inference engine

```
[engine, loglik] = enter_evidence(engine, evidence);
```

We get the marginal probabilities for GP for the given evidence

```
marg = marginal_nodes(engine, GP);
marg.T
```

```
ans = 0.5920    p(GP = active | ST = present) = 0.5920
      0.4080    p(GP = not active | ST = present) = 0.4080
```



82

Similarly, what is the probability that the G Protein is active if the Stimulant not present? i.e. p(G protein|Stimulant=not present)

We set the evidence to nothing (a blank cell array)

```
evidence = cell(1,N);
```

We add to evidence that ST was not present (2)

```
evidence{ST} = 2;
```

We pass this evidence to the inference engine

```
[engine, loglik] = enter_evidence(engine, evidence);
```

We get the marginal probabilities for GP for the given evidence

```
marg = marginal_nodes(engine, GP);
marg.T
```

```
ans = 0.5048    p(GP = active | ST = not present) = 0.5048
      0.4952    p(GP = not active | ST = not present) = 0.4952
```



83

BNT functionality

- The Bayes Net Toolbox for Matlab supports many conditional probability distributions, inference engines, methods for parameter learning, and some structure learning.
- It is free open source code and is available from <http://bnt.sourceforge.net/>



84

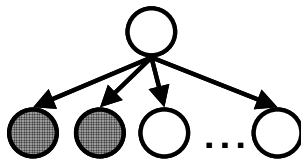
Missense mutations

- A single nucleotide polymorphism (SNP) is a mutation (insertion, deletion or substitution) observed in the genomic DNA of individuals of the same species.
- When the SNP results in an amino acid substitution in the protein product of the gene, it is called a *missense mutation*.
- A missense mutation can have various phenotypic effects. Here, we aim to predict whether a missense mutation has an effect or no effect on protein function.

Attributes

effect	Effect of mutation on functionality
ac	Solvent accessible area of native AA
rac	Accessibility relative to maximum accessibility in training set
bf	Normalised B-factor of native AA
nbf	Normalised B-factor of structural neighbourhood of native AA
bur	Mutant AA is charged AA at buried site
trn	Mutant AA occurs at glycine or proline in a turn
hlx	Mutant AA occurs in helical region and involves glycine or proline
ifc	Native AA is near subunit interface
BOTH nrent	Phylogenetic entropy of structural neighbourhood of native AA
rent	Normalised phylogenetic entropy of native AA
cnsd	Native AA is at conserved position in phylogenetic profile
ncnsd	Native AA is near conserved position in phylogenetic profile
uslaa	Mutant AA is not in phylogenetic profile
uslby	Mutant AA is not in the smallest AA class that includes the phylogenetic profile

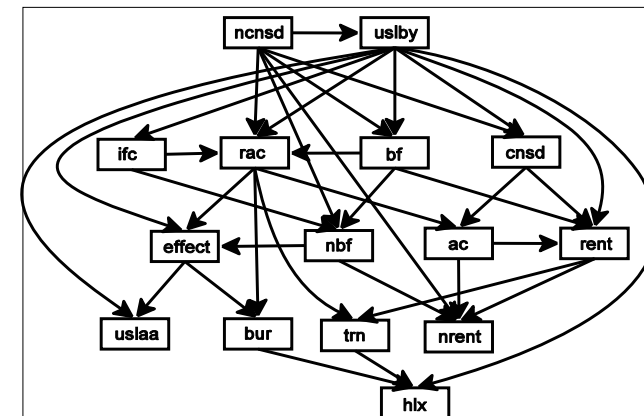
Naïve Bayes classifier



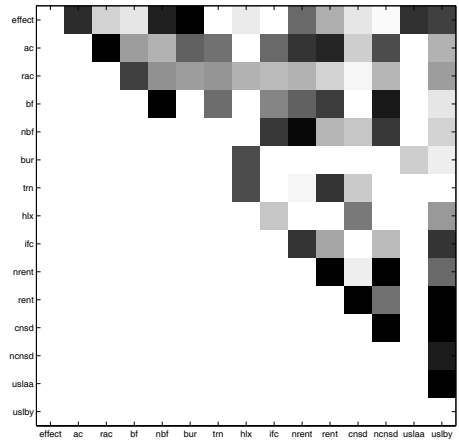
- Overall error rates 20%
- Area under ROC = 0.80

- performs well when evolutionary information is hidden,
- but poorly when structural information hidden

Learned network structure S



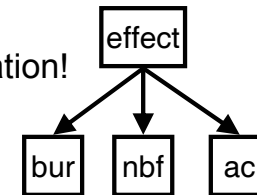
Posterior distribution of edges in learned structures



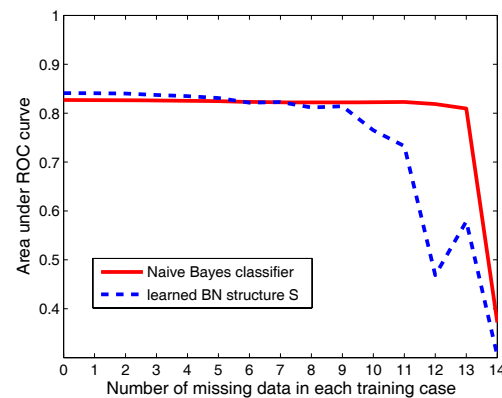
A simplified Bayesian network

- Three structural descriptors:
 - solvent accessible area of the native amino acid
 - whether the amino acid is charged at the buried site
 - the flexibility of its structural neighbourhood

- No evolutionary information!
- Same performance!



Learning from incomplete data



Conclusions/Recap

This application has shown that Bayesian networks

- Generalise well to new data
- Parameters can be learned from incomplete datasets
- Predictions can be made with missing data (through marginalising over the unknown variables)
- Structure learning can produce good compact models (compared to big fully connected graphs)
- A naïve Bayes' classifier is excellent at integrating information

Discussion

Bayesian networks for bioinformatics

An introduction to inference and learning

Many thanks must also go to
Dr Andrew Bulpitt & Prof David Westhead
for their contributions to this tutorial

14th Annual International Conference On Intelligent Systems For Molecular Biology

ISMB 2006 Fortaleza, Brazil
August 6-10, 2006
and 2nd Annual AB3C Conference: X-Meeting

Bayesian networks for bioinformatics

An introduction to inference and learning

Dr Chris Needham
Computing
The University of Leeds, UK
chrin@comp.leeds.ac.uk

Dr James Bradford
Cellular and Molecular Biology
The University of Leeds, UK
j.r.bradford@leeds.ac.uk

ISMB 2006 Fortaleza, Brazil August 6-10, 2006

Timetable

Time	Topic
45 mins	Introduction to Bayesian statistics
40 mins	Bayesian networks: representation and inference
35 mins	Learning from data
30 mins	coffee
30 mins	More advanced concepts
30 mins	Examples section
30 mins	Discussion

ISMB 2006 Fortaleza, Brazil
August 6-10, 2006

58

Introduction to Bayesian statistics

- Principles of learning from data
- Other machine learning approaches
- Probability: Classical vs Bayesian
- Probability theory
- Bayesian inference

ISMB 2006 Fortaleza, Brazil
August 6-10, 2006

59

Classification

- “Classification is hard” —Janet Thornton, ISMB05
- Don’t just want machine learning methods that classify well...
- ...we want to form an interpretable model.
- Yes/No is not enough, need to know why the decision was made.

ISMB 2006 Fortaleza, Brazil
August 6-10, 2006

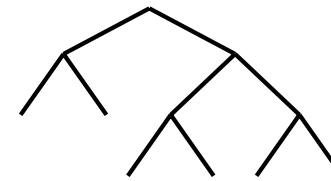
60

What is machine learning?

- Why do we want to learn from data?
- What problems can we tackle?

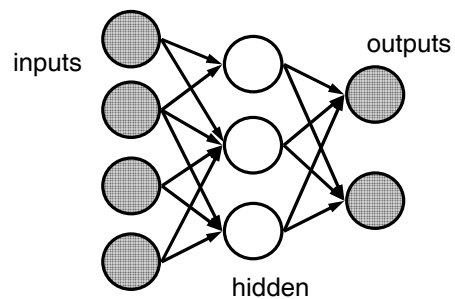
Decision Trees

- Gini index: $i(N) = 1 - \sum_j P^2(w_j)$
- 20 questions?



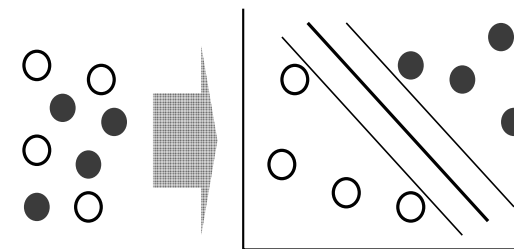
Neural networks

- 'Black box'



SVMs

Data x_i is transformed by a non-linear mapping $\phi(\cdot)$ to a high dimensional space where the $y_i = \phi(x_i)$ are linearly separable.



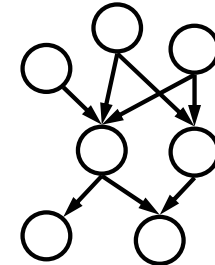
Drawbacks

- Many of you use machine learning algorithms...

What's wrong with them?

Bayesian networks

- A framework for explaining causal relationships consisting of a set of variables connected by a set of directed edges
- Probability calculus is used to describe the probabilistic relationship of each variable with its parents



Bayesian networks

- Combine domain knowledge and data
- Avoid over-fitting of data
- Handle incomplete datasets
- Allow learning about causal relationships

first some probability theory...

Bayesian Probability

- A classical probability is a physical property of the world
- The Bayesian probability of an event X is a person's degree of belief in that event
- Important difference: Do not need repeated trials in order to assign a Bayesian probability
- What is the probability that Brazil will win the world cup in 2010 ?

Probability assignment

- Probability assessment is the process of measuring a degree of belief and can be done in a number of ways:
 - probability wheel
 - ball drawing gambles

Your boss will give you an extra \$1000 if you:

A – Write 3 Journal papers this year

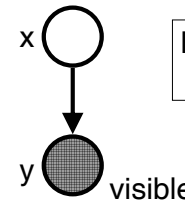
B – Choose a red marble from a bag of 100 marbles, with n red marbles

At what n would event A and B be equally likely?

Probability Calculus

Product rule: $p(x,y) = p(y|x)p(x)$

Sum rule: $p(x) = \sum_y p(x,y)$



Bayes' rule: $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$

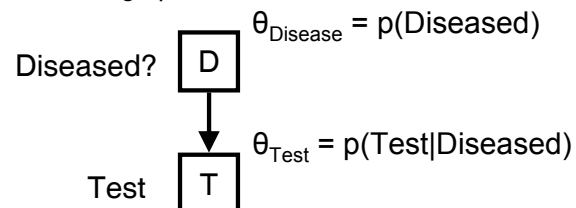
$$p(x|y) = \frac{p(y|x)p(x)}{\sum_x p(y|x)p(x)}$$

useful because often $p(y|x)$ easy to find, whereas $p(x|y)$ hard to assess

Have you got a disease?

- You've tested positive for a disease!
- What is the probability you have the disease?
- It depends on accuracy and sensitivity of the test and background (prior) probability of the disease.

In our probabilistic graphical models notation:



Let $p(\text{test is positive} \mid \text{you have the disease}) = 0.95$
 Suppose false positive rate is 5%: $p(T=\text{pos} \mid D=\text{false}) = 0.05$
 and 1% of population have the disease, $p(D=\text{true}) = 0.01$

$\theta_{\text{Disease}} = p(D)$			
Diseased?		T	F
		0.01	0.99

$\theta_{\text{Test}} = p(T D)$			
Test		pos	neg
T	0.95	0.05	
F	0.05	0.95	

$$\begin{aligned}
 P(D=\text{true} \mid T=\text{pos}) &= \frac{P(T=\text{pos} \mid D=\text{true}) * P(D=\text{true})}{P(T=\text{pos} \mid D=\text{true}) * P(D=\text{true}) + P(T=\text{pos} \mid D=\text{false}) * P(D=\text{false})} \quad (\text{Bayes' Rule}) \\
 &= 0.95 * 0.01 / (0.95 * 0.01 + 0.05 * 0.99) = 0.0095 / 0.0590 = \mathbf{0.161}
 \end{aligned}$$

Disease example (cont.)

- So probability of having the disease given you have tested positive is 16%
- Low?
- Of 100 people, we expect only 1 of them to have the disease, but we expect 5 to test positive (5%)
- So, of the 6 people who tested positive, we only expect 1 of them to actually have the disease. Indeed $1/6 \approx 0.16$
- [Using multiple independent tests increases the posterior probability]

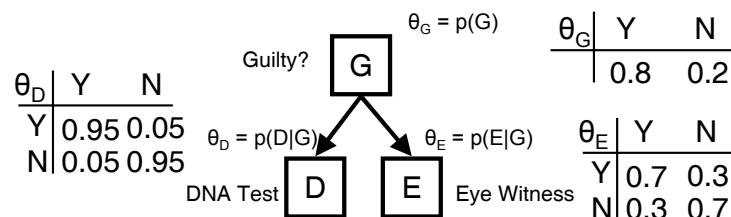
Guilty or not guilty?

- After opening statements, a jury believes there is an 80% probability that a suspect may be guilty
- Two pieces of evidence are presented, an eye witness report and a DNA test result
- Prior studies have shown that the reliability of an eye-witness report is 70% and the DNA test 95%
- The eye witness has identified the subject as the guilty party, while the DNA test indicates the suspect is innocent
- What would be the revised probability that the jury should believe the suspect to be guilty?

$$p(G,D,E) = p(G|D,E)p(D,E) \quad \text{--product rule}$$

$$p(G,D,E) = p(G)p(D|G)p(E|G) \quad \text{--from graph/CI*}$$

$$\Rightarrow p(G|D,E) = p(D|G)p(E|G)p(G)/p(D)p(E)$$



What's the probability of guilty, given witness says guilty, and DNA not guilty?

$$p(G=Y|D=N,E=Y) = p(D=N|G=Y)p(E=Y|G=Y)p(G=Y) / p(D=N)p(E=Y)$$

$$= 0.05 * 0.7 * 0.8 / (0.05*0.8+0.95*0.2) * (0.7*0.8+0.3*0.2)$$

$$= 0.028/(0.23*0.62)$$

$$= 0.196$$

$p(\text{guilty} | \text{evidence}) = 20\%$

Summary

- Discussed some methods of machine learning and their limitations
- Introduced graphical models and probability theory
- Made lots of promises about Bayes nets

Bayesian networks: representation and inference

- Joint probability distributions
- Bayesian networks
- Conditional independence
- Compact representation
- Conditional probability distributions
- Inference in Bayesian networks
- Calculating posterior probabilities

Joint Probability Distributions

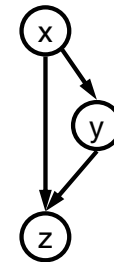
- Given a set of n variables, $X = \{x_1, \dots, x_n\}$, we want to form the joint probability distribution $p(X) = p(x_1, \dots, x_n)$
- Using this we can capture the relationships between sets of variables
- And perform inference of unknown values, such as $p(x_i | x_j, x_k)$

What are Bayesian networks?

- Bayesian networks encode the probabilistic relationships between variables
- Nodes represent variables $X = \{x_1, \dots, x_n\}$
- Edges represent relationships between variables
- A directed acyclic graph (DAG) is formed

Joint Probability Distributions

- Consider $p(x, y, z)$
- By successive application of the product rule:
- $$p(x, y, z) = p(y, z | x) p(x)$$
$$= p(z | x, y) p(y | x) p(x)$$



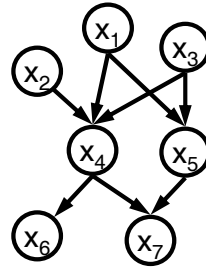
Joint Probability Distributions

$$p(x_1, \dots, x_n) = \prod_{i=1}^n p(x_i | pa_i)$$

where pa_i are the parents of x_i

DAG: directed acyclic graph

$$p(x_1, \dots, x_7) = p(x_1)p(x_2)p(x_3) \\ p(x_4|x_1, x_2, x_3) \\ p(x_5|x_1, x_3) \\ p(x_6|x_4)p(x_7|x_4, x_5)$$



Conditional Independence

- If two variables are independent given the state of a third variable, then they are said to be *conditionally independent*

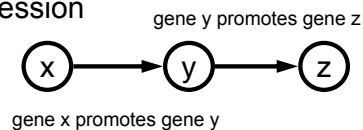
$$p(x, y | z) = p(x | z)p(y | z)$$

- Conditional Independence in Bayesian networks allows us to find variables that are independent and make the models of manageable size.

Serial connections

- Evidence transmitted unless state of variable in connection is known

e.g. Gene expression



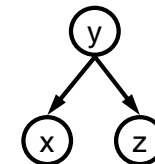
Y unknown: evidence of level of x effects level of z

Y known: the level of z depends only on y, and is conditionally independent of x

Diverging connections

- Evidence transmitted unless connection is instantiated

e.g. Transcription factor Y turns two genes X and Z on

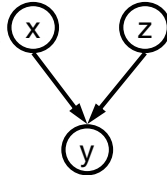


Y unknown: evidence gene x is on effects state of z

Y known: the state of z depends only on y, and is conditionally independent of x

Converging connections

- Evidence transmitted only if variable in connection or one of its children receives evidence



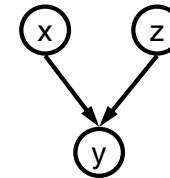
Genes X and Z promote gene Y

Y unknown: evidence of expression level of gene x does not help to infer the expression level of z
 -- *x and z are conditionally independent*

Y known: evidence of expression level of gene x does help to infer the expression level of z

$$p(x,z|y) \neq p(x|y)p(z|y)$$

Converging connections (example)



$$p(x,y,z) = p(y|x,z)p(x)p(z)$$

$$p(x,z) = p(x)p(z)$$

$$p(x,z|y) \neq p(x|y)p(z|y)$$

- Pixel colour in an image
 x = lighting colour, y = image colour, z = surface colour

Simple cell signalling pathway

- Consider a simple example consisting of :
 - a **stimulant**,
 - an extracellular **signal**,
 - an **inhibitor** of the signal,
 - a G protein-coupled **receptor**,
 - a **G protein** and the **cellular response**.

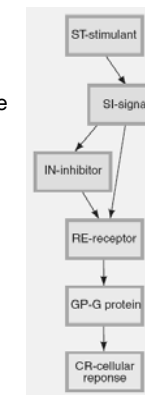
How do you model this pathway?

Prior knowledge

A Bayesian network can be constructed that expresses the relationships between variables

the concentration of the **signal** may affect the level of the **inhibitor**

the **G protein** should become active if the **receptor** binds



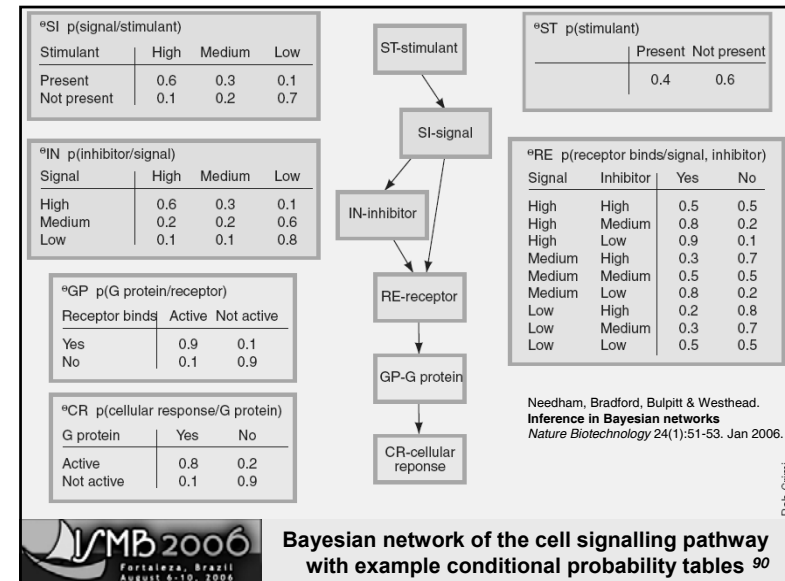
the **stimulant** may or may not generate a **signal**

whether the **signal** binds with the **receptor** depends on the concentrations of both the **signal** and the **inhibitor**

an active **G protein** initiates a cascade of reactions that causes the **cellular response**

Conditional probabilities

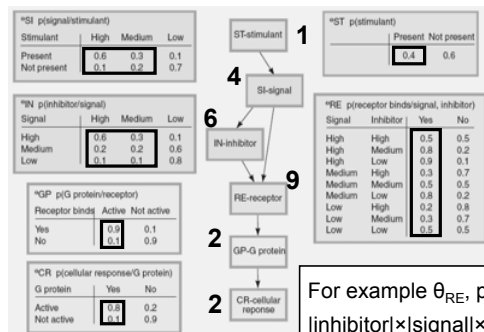
- Now we have a network structure, we need to know the conditional probability distributions θ_i
- These are much easier to specify, since they involve fewer variables and don't involve estimating posterior probabilities.
- For example we only need to know
 - $p(\text{G protein is active} \mid \text{receptor binds})$
- rather than
 - $p(\text{G protein is active} \mid \text{receptor binds, inhibitor is high, signal is medium, stimulant is not present})$
- We can learn these from data (next section)



Compactly expressing the JPD

$$p(\text{ST}, \text{SI}, \text{IN}, \text{RE}, \text{GP}, \text{CR}) = p(\text{CR}|\text{GP})p(\text{GP}|\text{RE})p(\text{RE}|\text{SI}, \text{IN})p(\text{IN}|\text{SI})p(\text{SI}|\text{ST})p(\text{ST})$$

full JPD has $(2 \times 3 \times 3 \times 2 \times 2 \times 2) - 1 = 143$ free parameters



$p(\text{G protein active} \mid \text{stimulant present})$

- Using the sum rule $p(x) = \sum_y p(x, y)$ and the product rule $p(x, y) = p(x|y)p(y)$ we marginalise over the unknown variables:

$$\begin{aligned}
 p(\text{GP} = \text{active} \mid \text{ST} = \text{present}) &= \sum_x \sum_y \sum_z p(\text{GP} = \text{active} \mid \text{RE} = x) p(\text{RE} = x \mid \text{IN} = y, \text{SI} = z) \\
 &\quad p(\text{IN} = y \mid \text{SI} = z) p(\text{SI} = z \mid \text{ST} = \text{present}) \\
 &= 0.592
 \end{aligned}$$

$$p(\text{GP} = \text{active} \mid \text{ST} = \text{not present}) = 0.5048$$

p(stimulant present | signal high)

- It's often of interest to calculate posterior probabilities – we use *Bayes' rule*

p(ST = present | SI = high) =

$$\frac{p(SI = high | ST = present)p(ST = present)}{p(SI = high | ST = present)p(ST = present) + p(SI = high | ST = absent)p(ST = absent)}$$

$$= \frac{0.6 \times 0.4}{(0.6 \times 0.4) + (0.1 \times 0.6)} = 0.8$$

Bayes' rule: $p(x|y) = \frac{p(y|x)p(x)}{p(y)}$



Calculating posterior probabilities 93

Why have you come to ISMB?

- What factors influence people's decision to attend ISMB (or not)?

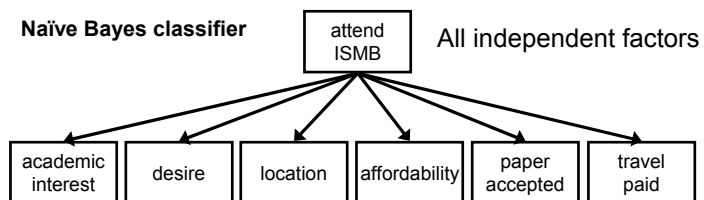
Academic interest	Location
Desire (beach?)	Cost/Affordability
Travel paid	Paper in

- Which of these are independent
- Can we draw a Bayesian network?

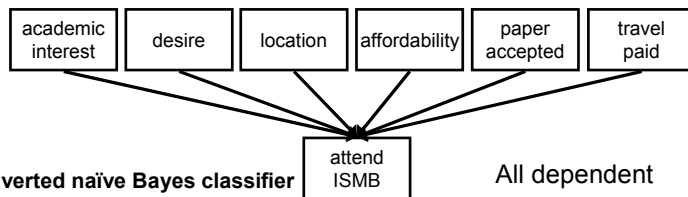


94

Naïve Bayes classifier

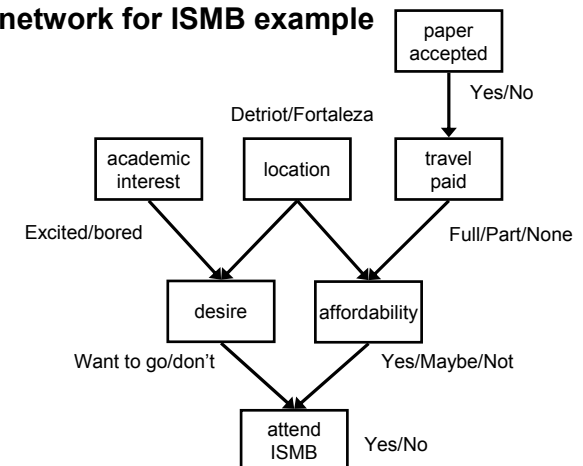


Inverted naïve Bayes classifier



95

Bayesian network for ISMB example



96

Summary

- Joint probability distributions
- Basic concepts of Bayesian networks
- Representation
- Inference
- Incorporating prior knowledge

Learning from data

- Learning model parameters from data
- Parameter priors
- Continuous variables as well as discrete
- Point estimates: maximum likelihood (ML), maximum *a posteriori* (MAP) estimates
- Bayesian learning – model averaging

Probabilistic terminology

- Prior $p(\theta)$
 - the prior probability assigned to a parameter, or to an event, in advance of any empirical evidence
- Posterior $p(\theta|D)$
 - the probability assigned to a parameter, or to an event, on the basis of its observed frequency in a sample, calculated from a prior probability by Bayes' rule
- Data set $D = \{x_n\}$, $n = 1, \dots, N$

Models with discrete variables

$p(a,b)$	b_1	b_2	b_3
a_1	1/9	1/9	1/9
a_2	1/9	1/9	1/9
a_3	1/9	1/9	1/9

Prior model parameters θ

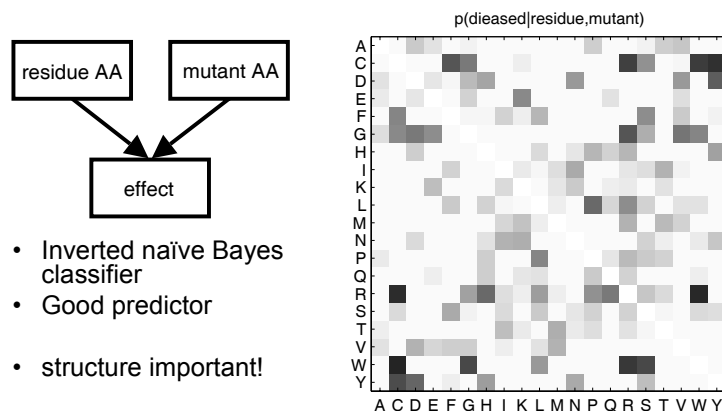
prior (model no data) often uniform uninformative
Dirichlet priors are used.

$p(a,b)$	b_1	b_2	b_3
a_1	0.30	0.10	0.01
a_2	0.02	0.05	0.15
a_3	0.10	0.24	0.03

Posterior model parameters θ

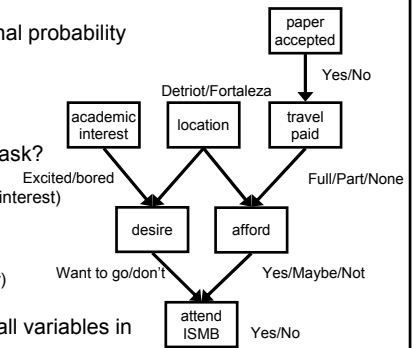
posterior (model given data) can be formed from pseudo-counts of observed frequencies in example training data

SNP prediction from Amino Acids



Example (ISMB BNet revisited)

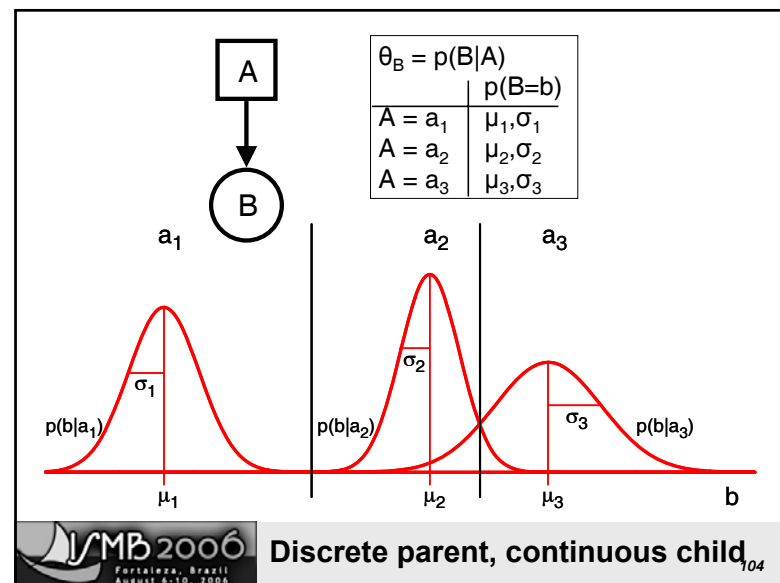
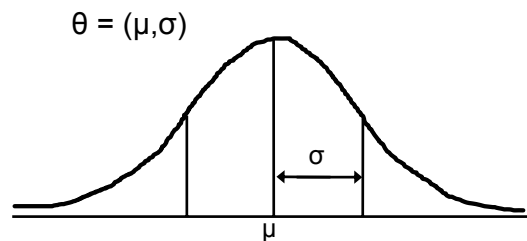
- How can we construct conditional probability tables for this example?
 - Using frequency counts
- How does it work?



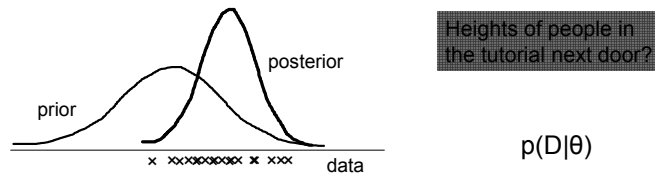
- What questions do we need to ask?
 - $p(\text{travel paid} | \text{paper accepted})$
 - $p(\text{desire to go} | \text{location, academic interest})$
 - $p(\text{academic interest})$
 - $p(\text{paper accepted})$
 - $p(\text{affordability} | \text{location, travel paid})$
 - $p(\text{attend ISMB} | \text{desire, affordability})$
- Compare to building JPD over all variables in this way!
- Later we will consider how to do this in the presence of incomplete data

Continuous data

- prior $p(\theta)$ – estimate of model parameters



Learning model parameters



- How do we fit a model to data?
- Do we measure how well the data fits the model? or how well the model fits the data?
- Given training data, how do we predict a new example?

Maximum Likelihood estimate

- Likelihood function (for independent observations)

$$L(\theta) = p(D|\theta) = \prod_{i=1}^N p(x_i|\theta)$$

- θ_{ML} is maximum likelihood model parameters

$$\theta_{ML} = \arg \max_{\theta} \ln p(D|\theta)$$

(data given model)

- Predictive distribution

$$p(x|D) \approx p(x|\theta_{ML})$$

MAP estimate

- θ_{MAP} is maximum posterior model parameters

$$\theta_{MAP} = \arg \max_{\theta} \ln p(\theta|D)$$

(model given data) $p(\theta|D) = p(D|\theta)p(\theta)$

- Predictive distribution

$$p(x|D) \approx p(x|\theta_{MAP})$$

Bayesian learning paradigm

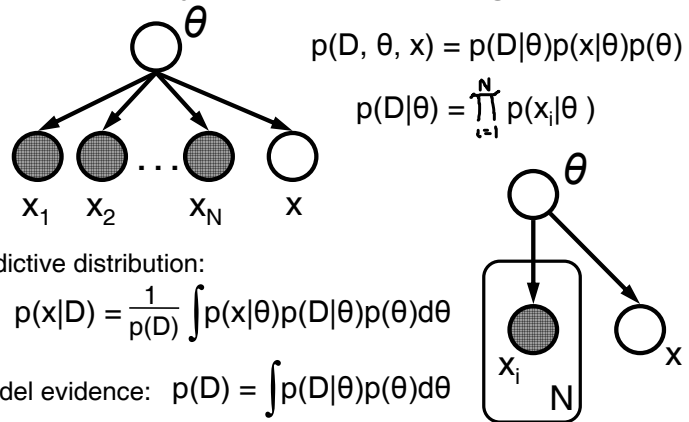
Predictive distribution

$$p(x|D) = \int p(x|\theta)p(\theta|D)d\theta$$

(x given model)(model given data)(over all models)

- Key idea is to marginalize over unknown parameters, rather than make point estimates
 - avoids the over-fitting of ML and MAP
 - allows direct model comparison
- Parameters are now latent variables
- Bayesian learning is an inference problem!

Bayesian Learning



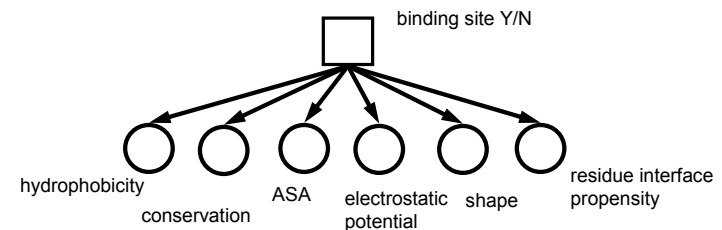
Summary

- Methods for learning model parameters
- Benefits of Bayesian learning
- Avoids over-fitting

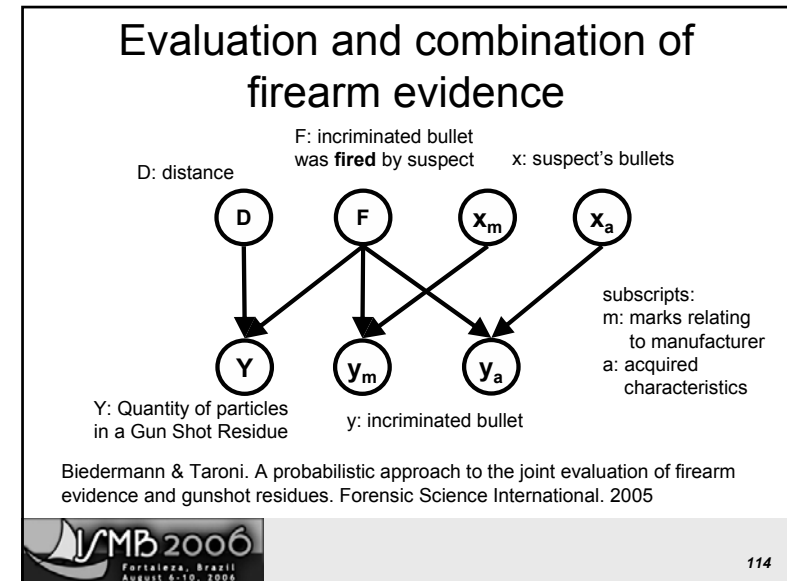
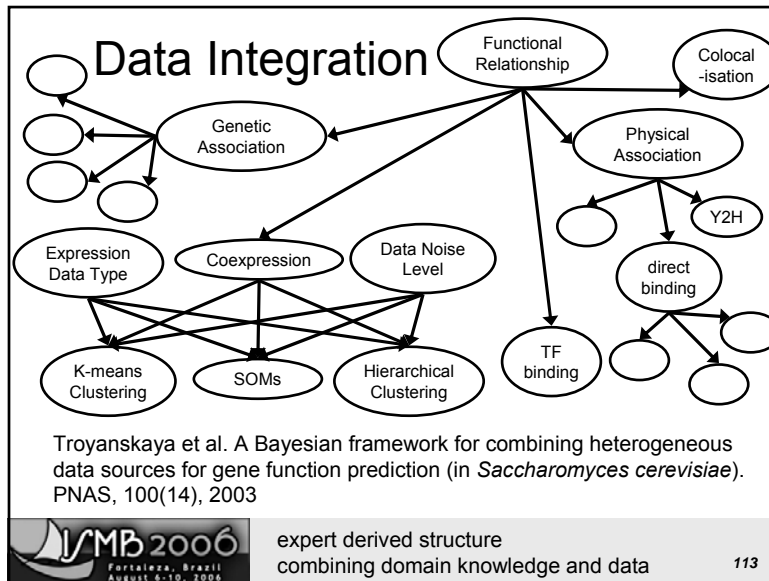
Review of Bayes Nets

- Binding site prediction
- Data integration for gene function prediction
- Evaluation of firearm evidence
- Medical decisions
- Gene cluster analysis

Binding site prediction



- Naïve Bayes classifier is excellent predictor of binding site patches on protein surfaces.



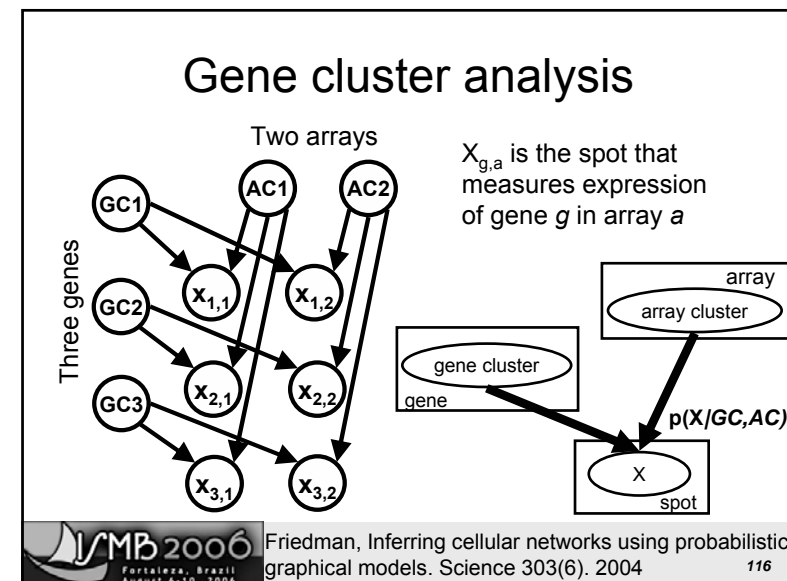
Medical Decisions

- Radiologists have an overwhelming task of integrating over a breadth of relevant and diverse data
- Breast disease diagnosis factors:
 - age, HRT, family history, calcifications (in a variety of patterns), mass attributes, asymmetric densities

Burnside. Bayesian Networks: Computer-assisted diagnosis Support in Radiology. Academic Radiology 12(4). 2005

I/MB 2006
Fortaleza, Brazil
August 6-10, 2006

115



More advanced concepts

- Learning from incomplete data
- Markov chain Monte Carlo methods
- Structure learning
- Dynamic Bayesian Networks
- Hidden Markov Models
- Latent variables
- Causality

Learning from incomplete data

- Parameters can be learned even when some variables are unknown in some cases
- Commonly the Expectation-Maximisation algorithm is used.

EM estimates the missing values by computing the expected values and updating the parameters using these expected values as if they were observed values

The EM algorithm

- EM finds local maxima for MAP or ML
- Starts with $\hat{\theta}$, a parameter configuration (random)
- Iteratively applies the expectation and maximisation steps until convergence
- **E-step.** The expected values of the missing data are inferred to form D_c – the most likely complete dataset given the current model parameters
- **M-step.** The configuration of $\hat{\theta}$ which maximises $p(\hat{\theta} | D_c)$ is found (for MAP)

Sampling methods

- Sampling methods have been used to estimate the full posterior distribution of the model parameters in the presence of incomplete data
- Monte Carlo methods such as **Gibbs sampling** are extremely accurate (but require lots of computation, take a long time to converge and become intractable as the sample size grows)
- **Gaussian approximation** is based on the fact that $p(\theta|D) \propto p(D|\theta)p(\theta)$ can be approximated as Gaussian distribution. With more training data the Gaussian peak becomes sharper $\rightarrow \theta_{MAP}$

Structure learning

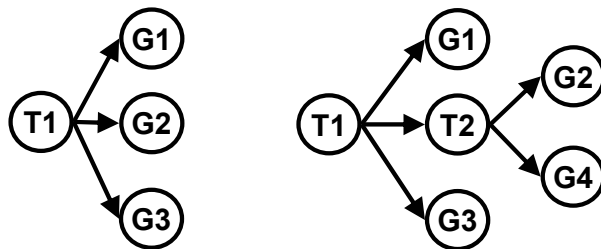
- We've seen that we can combine knowledge about the domain with data
 - i.e. get an expert to design a network structure based on known relationships/ independencies between the variables
- We can also learn the structure of the model!
 - search for good structures which capture the interactions between the variables, whilst maintaining a compact model

Structure Learning

- Greedy search
 - Iteratively: add, reverse or delete an edge
 - Score the structure S^h
- Score functions
 - Full Bayesian posterior
 - BIC score function

$$\ln p(D|S^h) \approx \ln p(D|\theta_s^{ML}, S^h) - \frac{1}{2} d \ln N$$

Learning Cellular Networks



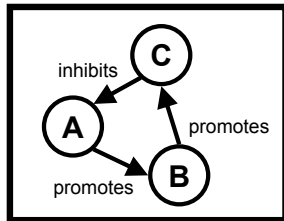
Review article: Friedman, Inferring cellular networks using probabilistic graphical models. Science 303(6). 2004

Inferring genetic networks

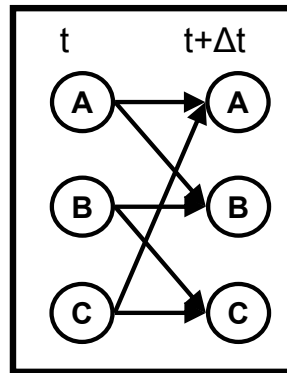
- Constructing a genetic network from microarray gene expression data by using a Bayesian network.
 - a gene corresponds to a node (random variable)
 - gene regulations are shown by directed edges
 - gene interactions are modelled by the conditional distribution of each gene
- Incorporate prior knowledge from protein-protein interactions, protein-DNA interactions, gene networks and literature
- Analysis of *Saccharomyces cerevisiae* gene expression data newly obtained by disrupting 100 genes, mainly transcription factors.

Dynamic Bayesian networks (DBNs)

- Expression levels of genes A, B, C



Static model – not a BN



Dynamic Bayesian network



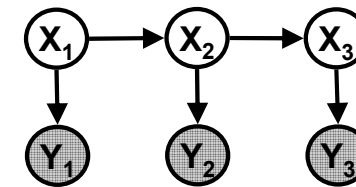
Murphy & Mian. Modelling Gene Expression Data using Dynamic Bayesian Networks. Tech Report. 1999.

125

Modelling the state of variable X , as a Markov process, with a DBN:



Hidden Markov Models (HMMs)



HMMs can be represented as Dynamic Bayesian networks, with hidden variables.

t doesn't have to be time
HMMs are often used for sequence alignment, where hidden state is INSERT, DELETE, or MATCH, and t is the next position in the sequence.

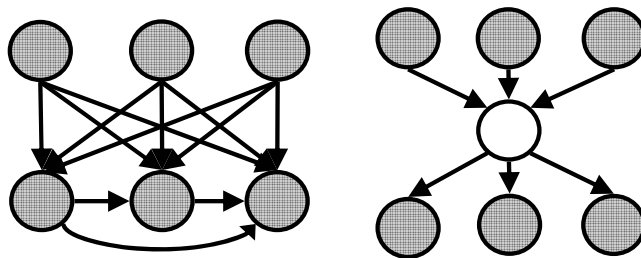
White nodes unobserved. Shaded nodes observed.



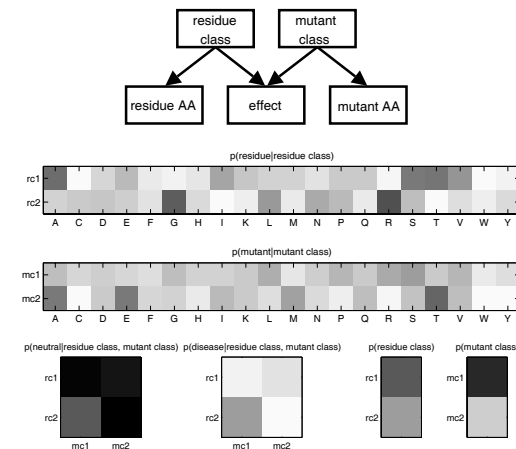
126

Latent (hidden) variables

- Latent variables can be added to models to capture additional information or reduce model size through expert knowledge



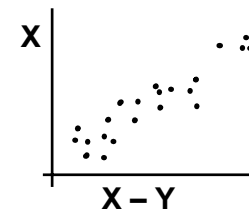
127



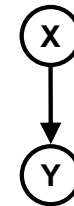
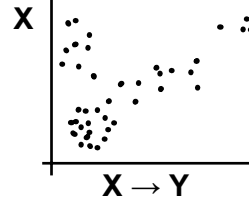
A latent variable model₁₂₈

Causality

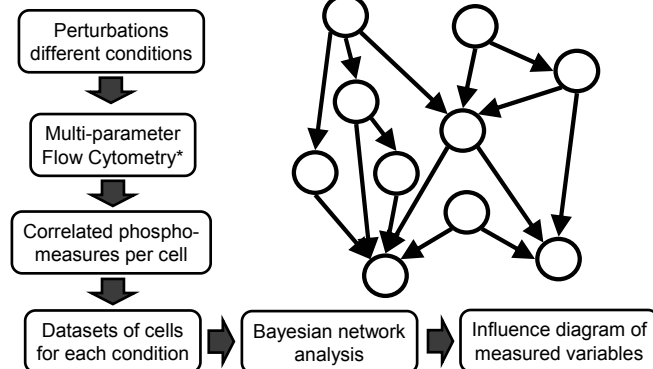
- The learning of causal relations is somewhat trickier
- We'd like to determine what effects what, rather than just what's related
- To do this requires the combination of expert knowledge, and interventions



No inhibition
 X inhibited
 Y inhibited



Causal protein-signalling networks



* measures 11 phosphoproteins and phospholipids in individual cells in each perturbation

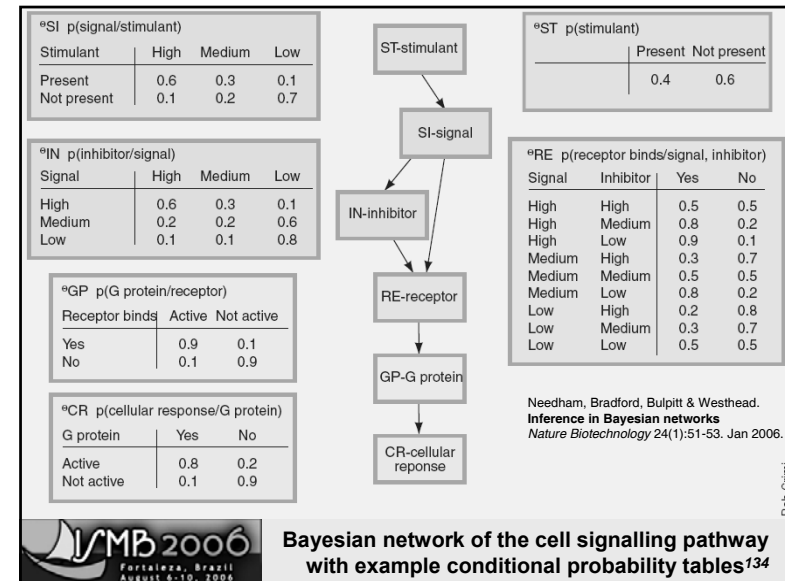
Sachs *et al.* Causal protein signalling networks derived from multi-parameter single-cell data. Science 308(5721) 200531

Summary

- Handling incomplete data
- Structure learning
- Learning causal relationships

Examples section

- The simple cell signalling example from earlier, example in Matlab
- An application of Bayesian networks to learning parameters and structures from data for predicting functional consequences of missense mutations



Cell signalling pathway example in Matlab

We have 6 discrete nodes

```
N = 6;
discrete_nodes = 1:N;
```

We will name them for ease of use

```
ST = 1; SI = 2; IN = 3; RE = 4; GP = 5; CR = 6;
```

And construct a DAG

```
dag = zeros(N,N);
dag(ST,SI) = 1;
dag(SI,[IN,RE]) = 1;
dag(IN,RE) = 1;
dag(RE,GP) = 1;
dag(GP,CR) = 1;
dag
```

```
dag =
0 1 0 0 0 0
0 0 1 1 0 0
0 0 0 1 0 0
0 0 0 0 1 0
0 0 0 0 0 1
0 0 0 0 0 0
```

Define the nodes sizes (SI and IN each take 3 values: high, med, low)

```
node_sizes = [2 3 3 2 2 2];
```

Initialise the Bayesian Network

```
bnet = mk_bnet(dag, node_sizes, 'discrete', discrete_nodes,
'names', {'Stimulant','Signal','Inhibitor','Receptor','G protein','Cell Res'});
```

Define the Conditional Probability Tables

```
bnet.CPD{ST} = tabular_CPD(bnet, ST, [0.4 0.6]);
bnet.CPD{SI} = tabular_CPD(bnet, SI, [0.6 0.1 0.3 0.2 0.1 0.7]);
bnet.CPD{IN} = tabular_CPD(bnet, IN, [0.6 0.2 0.1 0.3 0.2 0.1 0.1 0.6 0.8]);
bnet.CPD{RE} = tabular_CPD(bnet, RE, [0.5 0.3 0.2 0.8 0.5 0.3 0.9 0.8 0.5
0.5 0.7 0.8 0.2 0.5 0.7 0.1 0.2 0.5]);
bnet.CPD{GP} = tabular_CPD(bnet, GP, [0.9 0.1 0.1 0.9]);
bnet.CPD{CR} = tabular_CPD(bnet, CR, [0.8 0.1 0.2 0.9]);
```

We choose the Junction Tree algorithm for Inference

```
engine = jtree_inf_engine(bnet);
```

The CPT for p(stimulant)

```
ans = 0.4000  
0.6000
```

The CPT for p(signal|stimulant)

```
ans = 0.6000 0.3000 0.1000  
0.1000 0.2000 0.7000
```

The CPT for p(inhibitor|signal)

```
ans = 0.6000 0.3000 0.1000  
0.2000 0.2000 0.6000  
0.1000 0.1000 0.8000
```

The CPT for p(G protein|receptor)

```
ans = 0.9000 0.1000  
0.1000 0.9000
```

The CPT for p(receptor|inhibitor,signal)

```
ans(:,:,1) = 0.5000 0.8000 0.9000  
0.3000 0.5000 0.8000  
0.2000 0.3000 0.5000
```

```
ans(:,:,2) = 0.5000 0.2000 0.1000  
0.7000 0.5000 0.2000  
0.8000 0.7000 0.5000
```

the first table above shows the conditional probabilities when the receptor binds, and the second when the receptor does not bind

The CPT for p(cell res|G protein)

```
ans = 0.8000 0.2000  
0.1000 0.9000
```



137

Now we can make inferences! e.g. What is p(G protein|Stimulant=present) ?

We set the evidence to nothing (a blank cell array)

```
evidence = cell(1,N);
```

We add to evidence that ST was present (1)

```
evidence{ST} = 1;
```

We pass this evidence to the inference engine

```
[engine, loglik] = enter_evidence(engine, evidence);
```

We get the marginal probabilities for GP for the given evidence

```
marg = marginal_nodes(engine, GP);  
marg.T
```

```
ans = 0.5920 p(GP = active | ST = present) = 0.5920  
0.4080 p(GP = not active | ST = present) = 0.4080
```



138

Similarly, what is the probability that the G Protein is active if the Stimulant not present? i.e. p(G protein|Stimulant=not present)

We set the evidence to nothing (a blank cell array)

```
evidence = cell(1,N);
```

We add to evidence that ST was not present (2)

```
evidence{ST} = 2;
```

We pass this evidence to the inference engine

```
[engine, loglik] = enter_evidence(engine, evidence);
```

We get the marginal probabilities for GP for the given evidence

```
marg = marginal_nodes(engine, GP);  
marg.T
```

```
ans = 0.5048 p(GP = active | ST = not present) = 0.5048  
0.4952 p(GP = not active | ST = not present) = 0.4952
```



139

BNT functionality

- The Bayes Net Toolbox for Matlab supports many conditional probability distributions, inference engines, methods for parameter learning, and some structure learning.
- It is free open source code and is available from <http://bnt.sourceforge.net/>



140

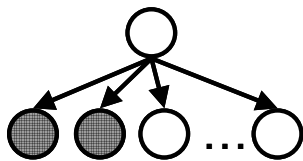
Missense mutations

- A single nucleotide polymorphism (SNP) is a mutation (insertion, deletion or substitution) observed in the genomic DNA of individuals of the same species.
- When the SNP results in an amino acid substitution in the protein product of the gene, it is called a *missense mutation*.
- A missense mutation can have various phenotypic effects. Here, we aim to predict whether a missense mutation has an effect or no effect on protein function.

Attributes

	effect	Effect of mutation on functionality
Structural	ac	Solvent accessible area of native AA
	rac	Accessibility relative to maximum accessibility in training set
	bf	Normalised B-factor of native AA
	nbf	Normalised B-factor of structural neighbourhood of native AA
	bur	Mutant AA is charged AA at buried site
	trn	Mutant AA occurs at glycine or proline in a turn
	hlx	Mutant AA occurs in helical region and involves glycine or proline
	ifc	Native AA is near subunit interface
BOTH	nrent	Phylogenetic entropy of structural neighbourhood of native AA
Evolutionary	rent	Normalised phylogenetic entropy of native AA
	cnsd	Native AA is at conserved position in phylogenetic profile
	ncnsd	Native AA is near conserved position in phylogenetic profile
	uslaa	Mutant AA is not in phylogenetic profile
	uslby	Mutant AA is not in the smallest AA class that includes the phylogenetic profile

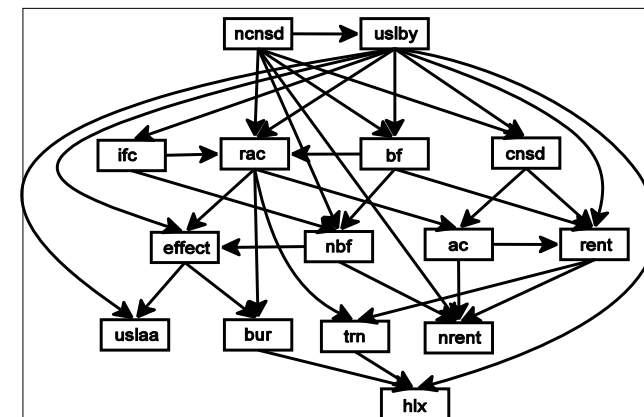
Naïve Bayes classifier



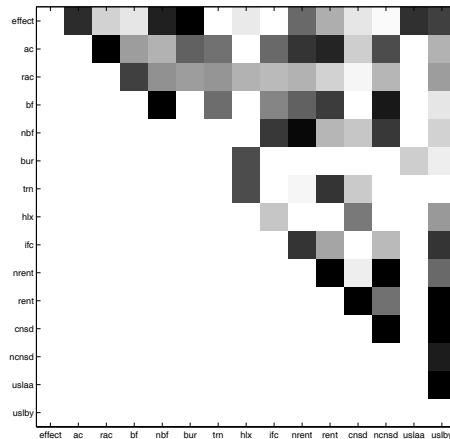
- Overall error rates 20%
- Area under ROC = 0.80

- performs well when evolutionary information is hidden,
- but poorly when structural information hidden

Learned network structure S



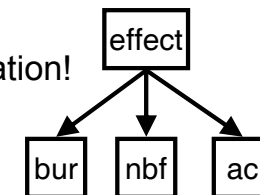
Posterior distribution of edges in learned structures



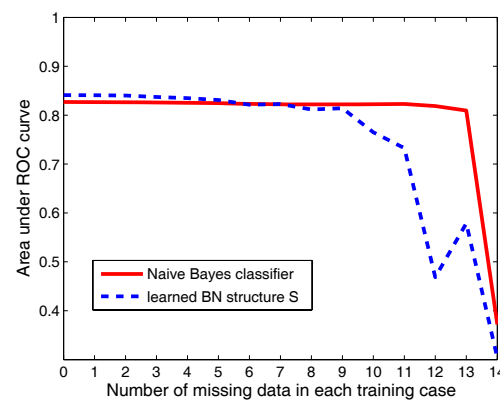
A simplified Bayesian network

- Three structural descriptors:
 - solvent accessible area of the native amino acid
 - whether the amino acid is charged at the buried site
 - the flexibility of its structural neighbourhood

- No evolutionary information!
- Same performance!



Learning from incomplete data



Conclusions/Recap

This application has shown that Bayesian networks

- Generalise well to new data
- Parameters can be learned from incomplete datasets
- Predictions can be made with missing data (through marginalising over the unknown variables)
- Structure learning can produce good compact models (compared to big fully connected graphs)
- A naïve Bayes' classifier is excellent at integrating information

Discussion

Bayesian networks for bioinformatics

An introduction to inference and learning

Many thanks must also go to
Dr Andrew Bulpitt & Prof David Westhead
for their contributions to this tutorial