Monte Carlo Methods
Another Application: Gene Expression

Dieter W. Heermann

Heidelberg University

January 2, 2020

1 Bayesian Networks

2 Gene Expression Data

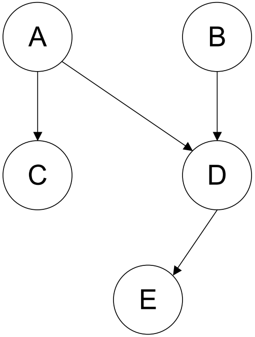
3 Bayesian Networks for Gene Expression Data
- Dynamic Bayesian Networks
- Monte Carlo

- Graphical display of dependence structure between multiple interacting qua[illegible]s (expression levels of different genes).
- Probabilistic semantics: Fits the stochastic nature of both the biological processes and noisy experiments. Capable of handling noise and estimating the confidence in the different features of the network.
- Due to lack of data: Extract features that are pronounced in the data rather than a single model that explains the data.
- Random variable X_i = measured expression level of gene i represented by nodes.
- Edges = regulatory interactions between genes.

- Define the functional form of the conditional distributions (e.g. multinomia
discrete variables, linear Gaussian for continuous variables).
    - Find the best network structure S
    - Given a network structure, find the best set of parameters for the conditional distributions (the most probable structure/parameter vector given the data)

Graphic representation of a joint distribution over a set of random variables A, B, C, D, E.

$$P(A, B, C, D, E) = P(A) * P(B) * P(C|A) * P(D|A, B) * P(E|D)$$



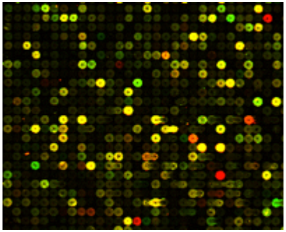
Example: Nodes represent gene expression while edges encode the interactions (cf. inhibition, activation)

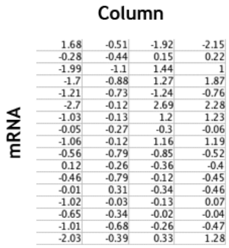
- Given a set of random variables $X = (X_1, ..., X_n)$, a Bayesian network is de as a pair $BN = (S, \theta)$, where
  - S is a directed acyclic graph (DAG), which is a graphical representation of the conditional independencies between variables in X
  - θ is the set of parameters for the conditional probability distributions of these variables.
  - In a Bayesian network, the probability of a state $x = (x_1, x_2, ..., x_n)$ is factored as

    $$P(x) = P(x_1|pa(x_1))P(x_2|pa(x_2)).x_n|pa(x_n)),$$

    where pa(x) denotes the parents of node x in the graph S

Consider a microarray (D_{ij}), whose rows ($D_{i.}$) correspond to genes and whose c[illegible]ns ($D_{.j}$) correspond to probes (tissue samples, experiments, etc.)



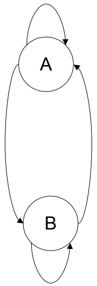


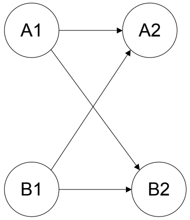
	Column			
mRNA	1.68	-0.51	-1.92	-2.15
	-0.28	-0.44	0.15	0.22
	-1.99	-1.1	1.44	1
	-1.7	-0.88	1.27	1.87
	-1.21	-0.73	-1.24	-0.76
	-2.7	-0.12	2.69	2.28
	-1.03	-0.13	1.2	1.23
	-0.05	-0.27	-0.3	-0.06
	-1.06	-0.12	1.16	1.19
	-0.56	-0.79	-0.85	-0.52
	0.12	-0.26	-0.36	-0.4
	-0.46	-0.79	-0.12	-0.45
	-0.01	0.31	-0.34	-0.46
	-1.02	-0.03	-0.13	0.07
	-0.65	-0.34	-0.02	-0.04
	-1.01	-0.68	-0.26	-0.47
	-2.03	-0.39	0.33	1.28

A real value is coming from one spot and tells if the concentration of a specific mRNA is higher(+) or lower(-) than the normal value

Dynamic Bayesian Networks I

- A Bayesian network should be a DAG (Direct Acyclic Graph).
- Random variable X_i = measured expression level of gene i. Arcs = regulatory interactions between genes.
- However, there are lots regulatory networks having directed cycles.
- Solve this by expanding into the time direction





Use DBN (Dynamic Bayesian Networks: BN with constraints on parents and children nodes) for sequential gene expression data

Dynamic Bayesian Networks II

- We are looking for a Bayesian network that is most probable given the data D (gene expression)

$$BN^* = \mathrm{argmax}_{\mathrm{BN}}\{P(BN|D)\}$$

  where

$$P(BN|D) = \frac{P(D|BN)P(BN)}{P(D)}$$

- There are many networks. An exhaustive search and scoring approach for the different models will not work in practice (the number of networks increases super-exponentially, $O(2^{n^{2}})$ for dynamic Bayesian networks)
- **Idea:** Sample the networks such that we eventually have sampled the most probable networks

Monte Carlo I

- Recall detailed balance condition for Monte Carlo

$$P(BN_{old}|D)P(BN_{old} \to BN_{new}|D) = P(BN_{new}|D)P(BN_{new} \to BN_{old}|D)$$

- Let us look at

$$P(BN|D) = \frac{P(D|BN)P(BN)}{P(D)}$$

- Assume $P(BN)$ is uniformly distributed *(We could incorporate knowledge)*
- Choose next BN with probability $P(BN_{\text{new}})$
- Accept the new BN with the following Metropolis-Hastings accept/rejection criterion:

$$\begin{aligned}
P &= \min\left\{1, \frac{P(BN_{\text{new}}|D)P(BN_{new} \to BN_{old}|D)}{P(BN_{\text{old}}|D)P(BN_{old} \to BN_{new}|D)}\right\} \\
&= \min\left\{1, \frac{P(D|BN_{\text{new}})P(BN_{\text{new}})P(D)}{P(D|BN_{\text{old}})P(BN_{\text{old}})P(D)}\right\} \\
&= \min\left\{1, \frac{P(D|BN_{\text{new}})P(BN_{\text{new}})}{P(D|BN_{\text{old}})P(BN_{\text{old}})}\right\} \\
&= \min\left\{1, \frac{P(D|BN_{\text{new}})}{P(D|BN_{\text{old}})}\right\}
\end{aligned}$$

Discrete model I

- Even though the amount of mRNA or protein levels, for example, can vary in a scale that is most conveniently modeled as continuous, we can still model the system by assuming that it operates with functionally discrete states
  - **activated** / **not activated** (2 states)
  - **under expressed** / **normal** / **over expressed** (3 states)
- Discretization of data values can be used to compromise between the
  - averaging out of noise
  - accuracy of the model
  - complexity/accuracy of the model/parameter learning
- Qualitative models can be learned even when the quality of the data is not sufficient for more accurate model classes

- Let N_{ijk} be the number of times we observe variable/node i in state k given parent node configuration j
- Summarize the number of total number of observations for variable i with parent node configuration j,

$$N_{ij} = \sum_{k=\mathbf{1}}^{r_i} N_{ijk}$$

- Since our states are discrete we use a multinomial distribution

Discrete model II

- the ML estimate of multinomial probabilities is obtained by the normalized counts

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}}$$

- A convenient prior distribution to choose for the parameters θ is given by the Dirichlet distribution

$$(\theta_{ij\mathbf{1}}, ..., \theta_{ijr_i}) \sim \text{Dirichlet}(\alpha_{ij\mathbf{1}}, ..., \alpha_{ijr_i})$$

- The Dirichlet distribution has PDF

$$f(\theta_{ij1}, ...\theta_{ijr_i}; \alpha_{ij1}, ...\alpha_{ijr_i}) = \frac{1}{B(\alpha_{ij})} \prod_{i=1}^{r_i} \theta_{ijr_i}^{\alpha_{ijr_i}-1}$$

  with $\theta_{ijr_i} \geq 0, \sum_i \theta_{ijr_i} = 1$ and hyperparameters $\alpha_{ijr_i} \geq 0, \alpha_{ij} = \sum_k \alpha_{ijr_i}$

- The normalization constant, the Beta function, can be expressed using the gamma function

$$B(\alpha_{ij}) = \frac{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijr_i})}{\Gamma(\alpha_{ij})}$$

Discrete model III

- The convenience arises from the fact that the distribution is conjugate to the multinomial distribution, i.e., if $P(\theta)$ is Dirichlet and $P(X|\theta)$ is multinomial, then $P(\theta|X)$ is Dirichlet as well
- The multinomial distribution is given (for $N_{ij} = \sum_k N_{ijk}$) by

$$f(N_{ij1}, ..., N_{ijr_i} | N_{ij}, \theta_{ij1}, ..., \theta_{ijr_i}) = \frac{N_{ij}!}{N_{ij1}! \cdots N_{ijr_i}!} \theta_{ij1}^{N_{ij1}} \cdots \theta_{ijr_i}^{N_{ijr_i}}$$

  and is the distribution of observations in r_i classes if N_{ij} observations are selected as outcomes of independent selection from the classes with probabilities $\theta_{ijk}, k = 1, ...r_i$

Structural Properties I

- In order to get reliable results we can focus on features that can be inferred
  - for example, we can define a feature, an indicator variable f with value 1 if and only if the structure of the model contains a path between nodes A and B
  - Looking at a set of models S with a good fit we can approximate the posterior probability of feature f by

$$P(f|D) = \sum_S f(S)P(S|D)$$

- With gene regulatory networks, one can look for only the most significant edges based on the scoring

- A Markov chain is defined over Bayesian nets so that it approaches a steady-state distribution as it is being run, and the probabilities of the states (networks) correspond to their posterior probability
- Individual nets are created as states in the chain and after (assumed) convergence, samples S_i are taken
- Posterior probability of an edge can then be approximated with

$$P(f(S)|D) \approx \frac{1}{n} \sum_{i=1}^{n} f(S_i)$$

Structural Properties II

- To work out the Monte Carlo Method to generate networks we first have to compute $P(D|S)$

$$\begin{aligned}
P(D|S) &= \int_\theta P(D|\theta, S)P(\theta|S)d\theta \\
&= \ldots \\
&= \prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})}\prod_{k=1}^{r_i}\frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}
\end{aligned}$$

- Monte Carlo moves: ADD, REMOVE, REVERSE edge in network