Using Past to Predict Future – Bayesian Networks and Medical Data

Jaroslaw 'Jaric' Zola

jzola@buffalo.edu · http://www.score-group.org/

SCoRe – Scalable Computing Research Group

Department of Computer Science and Engineering
Department of Biomedical Informatics
University at Buffalo, SUNY

Simple Exercise

- Suppose you are taking “Statistics” this semester, what are your chances of getting “A”? $P(Stat = A) = \dots$

Simple Exercise

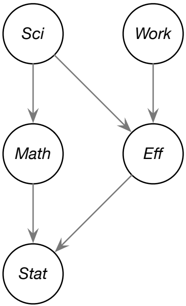
- Suppose you are taking "Statistics" this semester, what are your chances of getting "A"? $P(Stat = A) = \ldots$
- Which factors are important to answer this question?

Simple Exercise

- How much effort you will put into the course, *Eff*
- Your grade in math, *Math*
- Your overall workload, *Work*
- How much you like science *Sci*

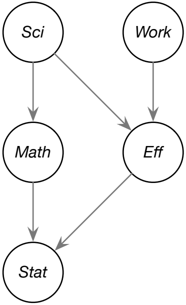
Simple Exercise

- Intuitively, we could plot dependencies between our variables as follows:



Simple Exercise

- Intuitively, we could plot dependencies between our variables as follows:



Sci	Like	Dislike
	0.75	0.25

Work	Low	High
	0.4	0.6

Sci \ *Math*	A	B	C
Like	0.50	0.25	0.25
Dislike	0.10	0.40	0.50

Sci Work \ *Eff*	Low	High
Like Low	0.15	0.85
Like High	0.25	0.75
Dislike Low	0.50	0.50
Dislike High	0.40	0.60

Math Eff \ *Stat*	A	B	C	Fail
A Low	0.5	0.5	0	0
A High	0.85	0.15	0	0
B Low	0.10	0.65	0.25	0
B High	0.15	0.70	0.15	0
C Low	0	0.25	0.35	0.40
C High	0.01	0.39	0.40	0.20

Simple Exercise

- We can describe our joint probability as:

$$P(Sci, Work, Math, Eff, Stat) = \\ P(Sci)P(Work)P(Math|Sci)P(Eff|Sci, Work)P(Stat|Math, Eff)$$

What Are Bayesian Networks?

- Class of Probabilistic Graphical Models
- Efficient and intuitive way to encode conditional independencies
- Formally: (G, P) where $G = (\mathcal{X}, E)$ is a DAG of conditional independencies and P is a probability over $\mathcal{X}$

Bayesian Networks Primer

- Suppose that $\mathcal{X} = \{X_1, \ldots, X_n\}$
- From the chain rule of probability:

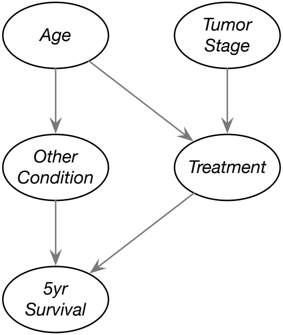
$$P(X_1, \ldots, X_n) = P(X_1)P(X_2|X_1)P(X_3|X_2, X_1) \ldots P(X_n|X_{n-1}, \ldots, X_1)$$

- BN $(G = (\mathcal{X}, E), P)$ provides much more efficient factorization:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i))$$

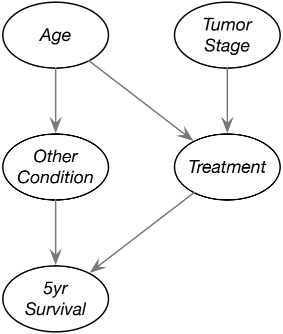
where $Pa(X_i)$ are parents of X_i in G

Power of Bayesian Networks



What is $P(Survival = yes|Stage = 0)$?

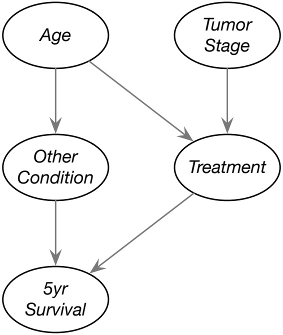
Power of Bayesian Networks



Which treatment to choose?

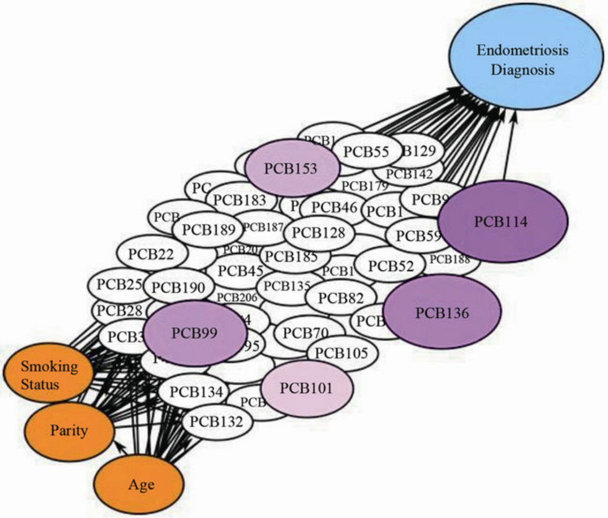
$$\underset{t}{\operatorname{argmax}} P(Survival = yes|Evidence, Treatment = t)$$

Power of Bayesian Networks



Which probabilities are needed to answer queries of interest?

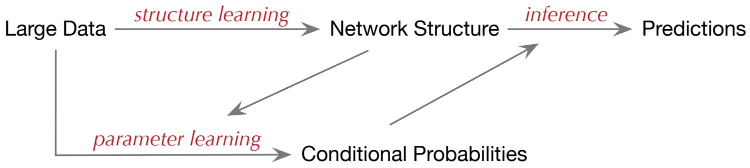
Real Networks Are Complicated



Important Questions

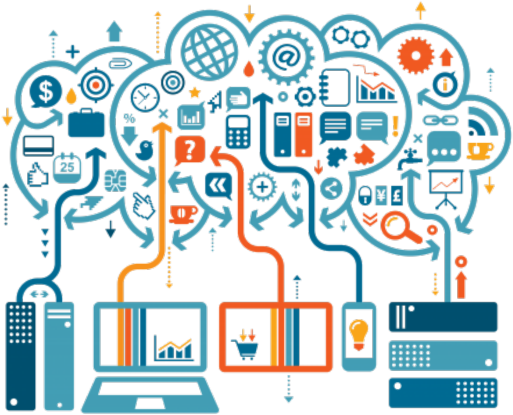
- Where all these probabilities come from?
- How do we build our network?

Bayesian Networks Workflow



Where Probabilities Come From?

Data!



Structure Learning

- Structure is a graph that best explains our data
  $Score(G) = P(G|D)$

$$Score(G) = \frac{P(D|G)P(G)}{P(D)}$$

- We want to find a graph with the highest *Score*

Structure Learning

- How many graphs (DAGs) with n variables?

Structure Learning

- How many graphs (DAGs) with n variables?

n	y
1	

Structure Learning

- How many graphs (DAGs) with n variables?

n	y
1	1
2	

Structure Learning

- How many graphs (DAGs) with n variables?

n	y
1	1
2	3
3	

Structure Learning

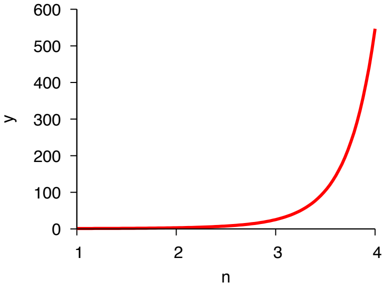
- How many graphs (DAGs) with n variables?

n	y
1	1
2	3
3	25
4	

Structure Learning

- How many graphs (DAGs) with n variables?

n	y
1	1
2	3
3	25
4	543
5	29,281
6	3,781,503



$$y(n) = \sum_{i=1}^{n} (-1)^{(i+1)} \binom{n}{i} 2^{i(n-i)} y(n-i)$$

Structure Learning

- Search space grows super-exponentially, and our problem is *NP*-hard
- If we can decompose *Score* as

$$Score(G) = \sum_{i=1}^{n} s(X_i, Pa(X_i))$$

  then:

  1. We can use DAG to order our variables
  2. We can disregard ordering of parents
  3. This reduces the search space to 2^n

- We still need a better approach!

Modern Computers Are Parallel

- 1993: Connection Machine (CM-5), $50,000,000
  1024 cores, 130 Gflop/s
- 2015: Intel Core i7, $1,000-$2,000
  4-8 cores, 80-160 Gflop/s

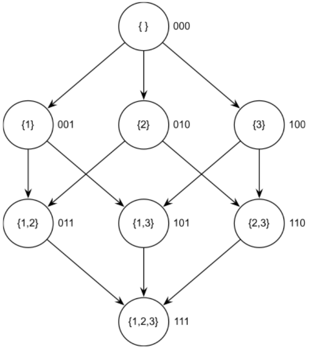
Modern Computers Are Parallel

- 1993: Connection Machine (CM-5), $50,000,000
  1024 cores, 130 Gflop/s
- 2015: Intel Core i7, $1,000-$2,000
  4-8 cores, 80-160 Gflop/s

The Joy of Parallel Computing

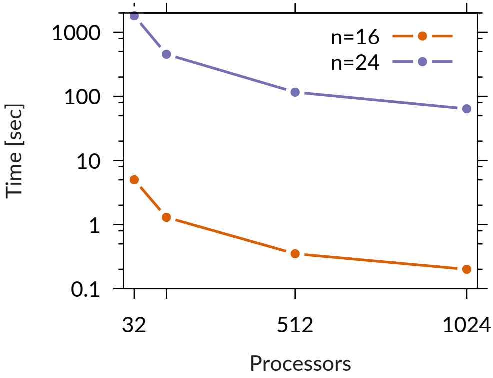
Parallel Structure Learning

- We consider all subsets $A \subseteq \{X_1, \ldots, X_n\}$ in increasing size
- For A we find best parents of X_i from $A - \{X_i\}$



Parallel Structure Learning

- Data with $m = 500$ observations

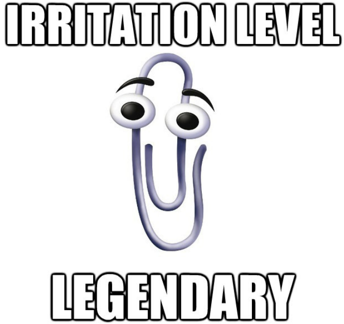


Applications of Bayesian Networks

- Clinical decision/support systems
- Gene networks and genes epistasis
- Recommender systems
- Diagnostic systems

Applications of Bayesian Networks

- Clinical decision/support systems
- Gene networks and genes epistasis
- Recommender systems
- Diagnostic systems
- And Microsoft clippy...



Things to Remember

- By learning from data we can make predictions about most likely outcomes
- Bayesian networks help to organize and use joint probabilities
- Parallel computing helps to tackle intractable problems

Questions?

http://www.score-group.org/
http://www.jzola.org/