V8 The Bayesian Network Representation

Our goal is to represent a joint distribution P over some set of random variables $X = \{ X_1, X_2, \ldots X_n \}$.

Even in the simplest case when these variables are binary-valued, a joint distribution requires the specification of $2^n - 1$ numbers – the probabilities of the 2^n different assignments of values $x_1, x_2, \ldots x_n$.

(Since their sum is 1, the number of free parameters is $2^n - 1$.)

For all but the smallest *n* the explicit representation of the joint distribution is unmanageable.

These problems could only be overcome after methods such as the Bayesian networks were developed.

Independent Random Variables

Let us start by considering a simple setting were we know that each X_i represents the outcome of a toss of coin *i*.

We typically assume that the different coin tosses are marginally independent, so that the distribution P will satisfy $(X_i \perp X_j)$ for any *i, j*.

More generally we assume that the distribution satisfies $(\mathbf{X} \perp \mathbf{Y})$ for any disjoint subsets of the variables **X** and **Y**.

Therefore $P(X_1, X_2, \ldots X_n) = P(X_1)\, P(X_2) \ldots P(X_n)$

If we use the standard representation of the joint distribution, this independence structure is obscured and the representation requires 2^n parameters.

Independent Random Variables

However, we can use a more natural set of parameters
for specifying this distribution.

If θ_i is the probability with which coin *i* lands heads,
the joint distribution *P* can be specified using the ***n* parameters** $\theta_1 \ldots \theta_n$.
These parameters implicitly specify the 2^n probabilities in the joint distribution.

For example, the probability that all of the coin tosses land heads is simply

$$\theta_1 \cdot \theta_2 \cdot \ldots \theta_n .$$

More generally, letting $\theta_{x_i} = \theta_i$ when $x_i = {x_i}^1$ and $\theta_{x_i} = 1 - \theta_i$ when $x_i = {x_i}^0$
we can define

$$P(x_1, \ldots, x_n) = \prod_i \theta_{x_i}$$

(This means that we have used for each coin the fact that the probabilities need to sum up to 1).

Independent Random Variables

This representation is limited, and there are many distributions that we cannot capture by choosing values for $\theta_1 \ldots \theta_n$.

This is quite obvious because the space of all joint distributions is $(2^n - 1)$- dimensional and we can in general not cover this by an *n*–dimensional manifold.

This only worked in this case of *n* independent random variables.

Conditional parametrization

Consider the problem faced by a company trying
to hire a recent college graduate.

The company‘s goal is to hire intelligent employees,
but there is no way to test intelligence directly.

Let‘s assume that the company has access to the student‘s SAT score.

Our probability space is now induced by the two random variables
`Intelligence (I)` and `SAT (S)`.

For simplicity, we assume that each of these takes 2 values:

- Val(I) = { i^1 , i^0 } which represent *high* and *low* intelligence
- Val(S) = { s^1 , s^0 } which represent the values *high* and *low* score.

Conditional parametrization

The joint distribution has 4 entries. E.g.

I	S	P(I,S)
i^0	s^0	0.665
i^0	s^1	0.035
i^1	s^0	0.06
i^1	s^1	0.24

There is an alternative and more natural way of representing the same joint distribution. Using the chain rule of conditional probabilites we have

$$P(I,S) = P(I)\,P(S|I)$$

Thus, instead of specifying the various joint entries P(I,S),
we can specify it in the form of P(I) and P(S|I).

Conditional parametrization

E.g. we can represent the previous joint distribution by the following 2 tables, one representing the prior distribution over I and the other the conditional probability distribution (CPD) of S given I:

i^0	i^1
0.7	0.3

I	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

Thus, a student of low intelligence is very unlikely to get a high SAT score (P (s^1 / i^0) = 0.05).

On the other hand, a student of high intelligence has a good chance to get a high SAT score (P (s^1 / i^1) = 0.8) but this is not certain.

Conditional parametrization

How can we parametrize this alternative representation?

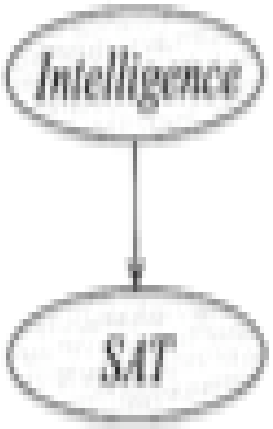
Here, we are using 3 binomial distributions,
one for P(I), and 2 for P(S| i^0) and P(S| i^1).

We can parametrize this representation using 3 independent parameters.

Our representation of the joint distribution as a 4-outcome multinomial
also required 3 parameters. → The new representation is not more compact.

The figure on the right shows
a simple Bayesian network
for this example.

Each of the 2 random variables I and S has a node,
and the edge from I to S represents the direction
of dependence in this model.



(a)

The student example: new variable `grade`

We will now assume that the company also has access to the student's `grade` G in some course.

Then, our probability space is the joint distribution of the 3 relevant random variables I, S, and G.

Assuming I and S as before and G takes on 3 values g^1 , g^2, g^3 representing the grades A, B and C, respectively. The joint distribution has 12 entries.

In this case, independence (of variables) does not help.

The student's intelligence is clearly correlated both with his/her SAT score and his/her grade.

The SAT score is also correlated with the grade.

The naive Bayes model

We expect that for our distribution P $P(g^1 | s^1) > P(g^1 | s^0)$

However, it is quite plausible that our distribution P
satisfies a conditional independence property:

If we know that the student has high intelligence, a high grade on the SAT
no longer gives us information about the student's performance in the class.

$$P(g | i^1, s^1) = P(g | i^1)$$

More generally, we may well assume that $P \models (S \perp G | I)$

This independence statement only holds if we assume
that the student's intelligence is the only reason
why his/her grade and SAT score might be correlated.

The naive Bayes model

By simple probabilistic reasoning we also have that

$$P(I,S,G) = P(S,G \mid I)\, P(I)$$

The previous eq. $P \models (S \perp G \mid I)$ implies that $P(S,G \mid I) = P(S \mid I)\, P(G \mid I)$

Hence, we get $P(I,S,G) = P(S \mid I)\, P(G \mid I)\, P(I)$

Thus, we have factorized the joint distribution P(I,S,G) as
a product of 3 **conditional probability distributions** (CPDs).

P(I) and P(S | I) can be re-used from p.7.

P(G | I) could, e.g., have the following form

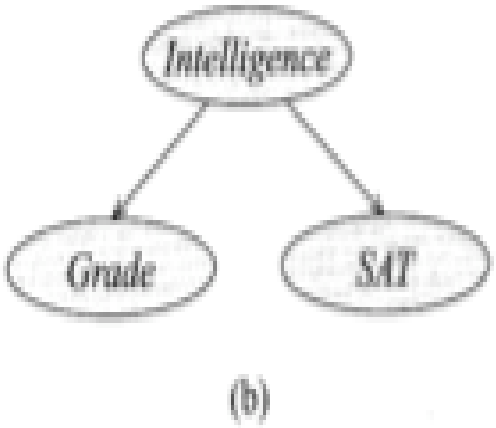
I	g^1	g^2	g^3
i^0	0.2	0.34	0.46
i^1	0.74	0.17	0.09

The naive Bayes model

Together, these 3 CPDs fully specify the joint distribution (assuming P |= (S ⊥ G | I)).

For example $P(i^1, s^1, g^2) = P(i^1)\, P(s^1 \mid i^1)\, P(g^2 \mid i^1) = 0.3 \cdot 0.8 \cdot 0.17 = 0.0408$

This probabilistic model is represented using the Bayesian network shown below.



(b)

In this case, the alternative parametrization is more compact than the joint.

The 3 binomial distributions P(I), $P(S \mid i^1)$ and $P(S \mid i^0)$ require 1 parameter each.
The 3-valued multinomial distributions $P(G \mid i^1)$ and $P(G \mid i^0)$ require 2 parameters each.
This makes 7 parameters, compared to the joint distribution with 12 entries, and thus 11 independent parameters.

Another advantage is the modularity. We could re-use the previous tables from p.7.

The naive Bayes model: general model

The naive Bayes model assumes that instances fall into one of a number of mutually exclusive and exhaustive **classes**.

Thus, we have a class variable C that takes on values in some set $\{c^1, c^2, \ldots, c^k\}$.

In our example, the class variable is the student's `intelligence` I and there are 2 classes *high* and *low*.

The model also includes some **features** $X_1, \ldots X_n$ whose values are typically observed.

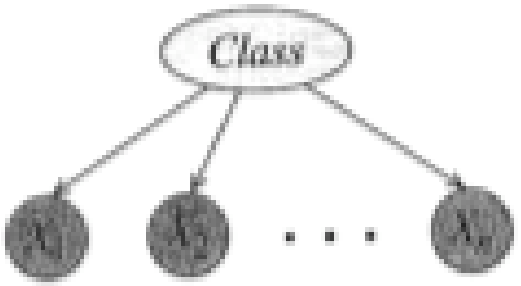
The **naive Bayes assumption** is that the features are conditionally independent given the instance's class.

In other words, within each class of instances, the different properties can be determined independently.

The naive Bayes model: general model

Formally we have $(X_i \perp X_j \mid C)$ for all i and $j \neq i$.

This model can be presented by



Here, darker oval represent variables that are always observed when the network is used.

We can show that the model **factorizes** as

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

We can represent the joint distribution by a prior distribution P(C) and a set of CPDs, one for each of the *n* finding variables.
The number of required parameters is linear in the number of variables, not exponential as for the explicit representation of the joint.

Side remark on naive Bayesian models

Naive Bayesian models were often used in the early days of medical diagnostics.

However, the model makes several strong assumptions that are generally not true, specifically that the patient can have at most one disease, and that given the patient‘s disease, the presence or absence of different symptoms, and the values of different tests, are all independent.

Experience showed that the model tends to overestimate the impact by certain evidence by „overcounting“ it. E.g. both high blood pressure and obesity are strong indicators of heart disease. But these 2 symptoms are themselves highly correlated.

It was found that the diagnostic performance of naive Bayesian models decreased as the number of features was increased.

→ more complex Bayesian models were developed.

Still, naive Bayesian models are useful in a variety of applications.

Bayesian Analysis of Protein-Protein Complexes

A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data

Ronald Jansen,[1]* Haiyuan Yu,[1] Dov Greenbaum,[1] Yuval Kluger,[1]
Nevan J. Krogan,[4] Sambath Chung,[1,2] Andrew Emili,[4]
Michael Snyder,[2] Jack F. Greenblatt,[4] Mark Gerstein[1,3]†

We have developed an approach using Bayesian networks to predict protein-protein interactions genome-wide in yeast. Our method naturally weights and combines into reliable predictions genomic features only weakly associated with interaction (e.g., messenger RNA coexpression, coessentiality, and colocalization). In addition to de novo predictions, it can integrate often noisy, experimental interaction data sets. We observe that at given levels of sensitivity, our predictions are more accurate than the existing high-throughput experimental data sets. We validate our predictions with TAP (tandem affinity purification) tagging experiments. Our analysis, which gives a comprehensive view of yeast interactions, is available at genecensus.org/intint.

Science 302 (2003) 449

Noisy Data — Clear Statements?

For **yeast**: ~ 6000 proteins → ~18 million potential interactions

rough estimates: ≤ 100000 interactions occur

→ 1 true positive for ca. 200 potential candidates = **0.5%**

→ **decisive** experiment must have **accuracy** << 0.5% false positives

But **different experiments** detect different interactions

For yeast: 80000 interactions known,

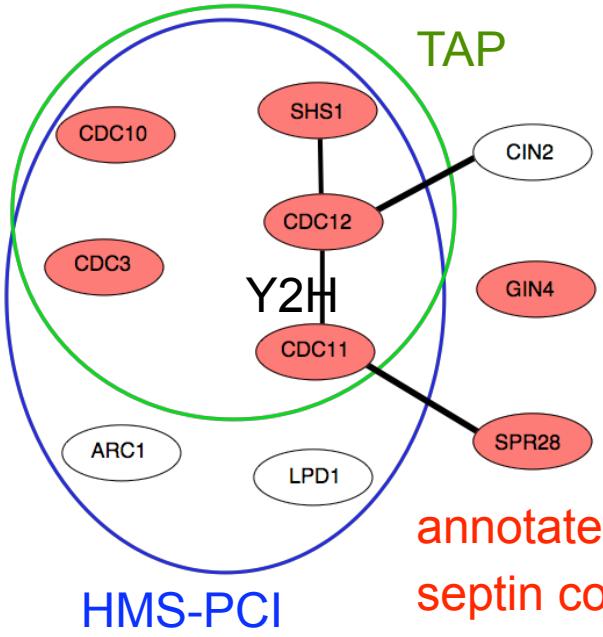
only 2400 found in > 1 experiment

Y2H: → many false positives

(up to 50% errors)

Co-expression: → gives indications at best

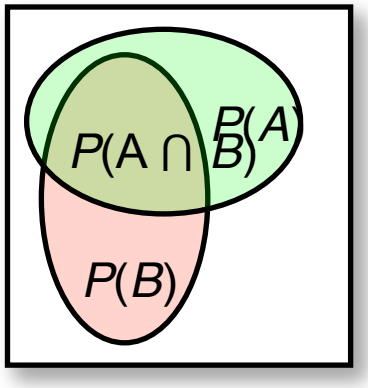
Combine weak indicators = ???



see: von Mering (2002)

Review: Conditional Probabilities

Joint probability for "*A* and *B*":



$$P(A \cap B) = P(A|B)\,P(B) = P(B|A)\,P(A)$$

Solve for conditional probability for "*A* when *B* is true"
→ Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} = \frac{P(B|A)}{P(B)}\,P(A)$$

P(*A*) = prior probability (marginal prob.) for "*A*" → no prior knowledge about *A*

P(*B*) = prior probability for "*B*" → normalizing constant

P(*B* | *A*) = conditional probability for "*B* given *A*"

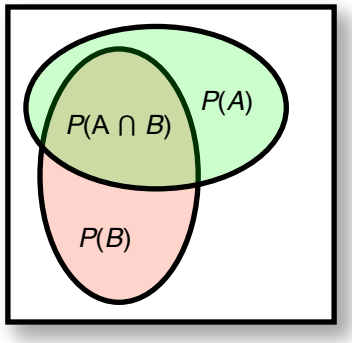
P(*A* | *B*) = posterior probability for "*A* given *B*"

→ Use information about *B* to improve knowledge about *A*

What are the Odds?

Express Bayes theorem

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} = \frac{P(B|A)}{P(B)}\,P(A)$$



in terms of odds:

• Also consider case "*A* does not apply": $P(\bar{A}|B) = \frac{P(B|\bar{A})}{P(B)}\,P(\bar{A})$

• odds for *A* when we know about *B*
(we will interpret *B* as information or features):

$$O(A|B) = \frac{P(A|B)}{P(\bar{A}|B)} = \frac{P(B|A)}{P(B|\bar{A})}\,\frac{P(A)}{P(\bar{A})} = \Lambda(A|B)\,O(A)$$

posterior odds for *A* — likelihood ratio — prior odds for *A*

$\Lambda(A \mid B) \rightarrow$ by how much does our knowledge about *A* improve?

2 types of Bayesian Networks

Encode conditional dependencies between evidences

= "*A* depends on *B*"
with the conditional probability *P(A | B)*

Evidence nodes can have a variety of types: numbers, categories, …

(1) **Naive Bayesian network**
→ independent odds

$$O(A|B,C) = \Lambda(A|B)\,\Lambda(A|C)\,O(A)$$

(2) **Fully connected Bayesian network**
→ table of joint odds

	B	!B
C	0.3	0.16
!C	0.4	0.14

$\Leftrightarrow \Lambda(A|B,C)$

Improving the Odds

Is a given protein pair **AB a complex** (from all that we know)?

$$O_{post}(\text{Complex}|f_1, f_2, \ldots) = \Lambda(\text{Complex}|f_1, f_2, \ldots)\, O_{prior}(\text{Complex})$$

likelihood ratio: **improvement** of the odds when we know about features f_1, f_2, …

prior odds for a random pair AB to be a complex

Idea: determine from known complexes and use for prediction of new complexes

estimate (somehow)

Features used by Jansen et al (2003):

- 4 experimental data sets of complexes
- mRNA co-expression profiles
- biological functions annotated to the proteins (GO, MIPS)
- essentiality for the cell

Gold Standard Sets

To determine $\Lambda(\text{Complex}|f_1, f_2, \ldots) = \dfrac{P(f_1, f_2, \ldots|\text{Complex})}{P(f_1, f_2, \ldots|\text{no Complex})}$

→ use two data sets with **known** features f_1, f_2, … for **training**

Requirements for training data:

i) should be independent of the data serving as evidence

ii) large enough for good statistics

iii) free of systematic bias

Gold Standard Positive Set (GP):

8250 complexes from the hand-curated MIPS catalog of protein complexes
(MIPS stands for Munich Information Center for Protein Sequences)

Gold Standard Negative Set (GN):

2708746 (non-)complexes formed by proteins from different cellular compartments
(assuming that such protein pairs likely do not interact)

Prior Odds

$$O_{prior}(\text{Complex}) = \frac{P(\text{Complex})}{P(\text{no Complex})} = \frac{P(\text{Complex})}{1 - P(\text{Complex})}$$

Jansen et al:

- estimated ≥ 30000 existing complexes in yeast
- 18 Mio. possible complexes → P(Complex) ≈ 1/600

→ O_{prior} = 1/600

→ The odds are 600 : 1 against picking a complex at random

→ expect 50% good hits (TP > FP) with $\Lambda \approx 600$

Note: O_{prior} is mostly an educated guess

Essentiality

Test whether both proteins are essential (E) for the cell or not (N)

→ we expect that for protein complexes, EE or NN should occur more often

pos/neg: # of gold standard positives/ negatives with essentiality information

$$L(\text{Ess}) = \frac{P(\text{Ess} \mid \text{pos})}{P(\text{Ess} \mid \text{neg})}$$

Essentiality	pos	neg	P(Ess|pos)	P(Ess|neg)	L(Ess)
EE	1114	81924	5,18E-01	1,43E-01	3,6
NE	624	285487	2,90E-01	4,98E-01	0,6
NN	412	206313	1,92E-01	3,60E-01	0,5
sum	2150	573724	1,00	1,00	

possible values of the feature

overlap of gold standard sets with feature values

probabilities for each feature value

likelihood ratios

$\frac{1114}{2150}$ = 0,518

$\frac{0.19}{0.36}$ = 0,5

mRNA Co-Expression

Publicly available expression data from

- the Rosetta compendium
- the yeast cell cycle

Expression correlation		# protein pairs	Gold standard overlap					P(exp|pos)	P(exp|neg)	L
			pos	*neg*	sum(*pos*)	sum(*neg*)	sum(*pos*)/ sum(*neg*)			
Values	0.9	678	16	45	16	45	0.36	2.10E-03	1.68E-05	124.9
	0.8	4,827	137	563	153	608	0.25	1.80E-02	2.10E-04	85.5
	0.7	17,626	530	2,117	683	2,725	0.25	6.96E-02	7.91E-04	88.0
	0.6	42,815	1,073	5,597	1,756	8,322	0.21	1.41E-01	2.09E-03	67.4
	0.5	96,650	1,089	14,459	2,845	22,781	0.12	1.43E-01	5.40E-03	26.5
	0.4	225,712	993	35,350	3,838	58,131	0.07	1.30E-01	1.32E-02	9.9
	0.3	529,268	1,028	83,483	4,866	141,614	0.03	1.35E-01	3.12E-02	4.3
	0.2	1,200,331	870	183,356	5,736	324,970	0.02	1.14E-01	6.85E-02	1.7
	0.1	2,575,103	739	368,469	6,475	693,439	0.01	9.71E-02	1.38E-01	0.7
	0	9,363,627	894	1,244,477	7,369	1,937,916	0.00	1.17E-01	4.65E-01	0.3
	-0.1	2,753,735	164	408,562	7,533	2,346,478	0.00	2.15E-02	1.53E-01	0.1
	-0.2	1,241,907	63	203,663	7,596	2,550,141	0.00	8.27E-03	7.61E-02	0.1
	-0.3	484,524	13	84,957	7,609	2,635,098	0.00	1.71E-03	3.18E-02	0.1
	-0.4	160,234	3	28,870	7,612	2,663,968	0.00	3.94E-04	1.08E-02	0.0
	-0.5	48,852	2	8,091	7,614	2,672,059	0.00	2.63E-04	3.02E-03	0.1
	-0.6	17,423	-	2,134	7,614	2,674,193	0.00	0.00E+00	7.98E-04	0.0
	-0.7	7,602	-	807	7,614	2,675,000	0.00	0.00E+00	3.02E-04	0.0
	-0.8	2,147	-	261	7,614	2,675,261	0.00	0.00E+00	9.76E-05	0.0
	-0.9	67	-	12	7,614	2,675,273	0.00	0.00E+00	4.49E-06	0.0
Sum		18,773,128	7,614	2,675,273	-	-	-	1.00E+00	1.00E+00	1.0

Jansen et al, Science 302 (2003) 449

Biological Function

Use MIPS function catalog and Gene Ontology function annotations

- determine functional class shared by the two proteins; small values (1-9)

Indicate highest MIPS function or GO BP similarity

- count how many of the 18 Mio potential pairs share this classification

MIPS function similarity		# protein pairs	Gold standard overlap					P(MIPS|pos)	P(MIPS|neg)	L
			pos	neg	sum(pos)	sum(neg)	sum(pos)/ sum(neg)			
Values	1 -- 9	6,584	171	1,094	171	1,094	0.16	2.12E-02	8.33E-04	25.5
	10 -- 99	25,823	584	4,229	755	5,323	0.14	7.25E-02	3.22E-03	22.5
	100 -- 1000	88,548	688	13,011	1,443	18,334	0.08	8.55E-02	9.91E-03	8.6
	1000 -- 10000	255,096	6,146	47,126	7,589	65,460	0.12	7.63E-01	3.59E-02	21.3
	10000 -- Inf	5,785,754	462	1,248,119	8,051	1,313,579	0.01	5.74E-02	9.50E-01	0.1
Sum		6,161,805	8,051	1,313,579	-	-	-	1.00E+00	1.00E+00	1.0

GO biological process similarity		# protein pairs	Gold standard overlap					P(GO|pos)	P(GO|neg)	L
			pos	neg	sum(pos)	sum(neg)	sum(pos)/ sum(neg)			
Values	1 -- 9	4,789	88	819	88	819	0.11	1.17E-02	1.27E-03	9.2
	10 -- 99	20,467	555	3,315	643	4,134	0.16	7.38E-02	5.14E-03	14.4
	100 -- 1000	58,738	523	10,232	1,166	14,366	0.08	6.95E-02	1.59E-02	4.4
	1000 -- 10000	152,850	1,003	28,225	2,169	42,591	0.05	1.33E-01	4.38E-02	3.0
	10000 -- Inf	2,909,442	5,351	602,434	7,520	645,025	0.01	7.12E-01	9.34E-01	0.8
Sum		3,146,286	7,520	645,025	-	-	-	1.00E+00	1.00E+00	1.0

Jansen et al, Science 302 (2003) 449

Experimental Data Sets

In vivo pull-down:	Gavin et al, *Nature* **415** (2002) 141	31304 pairs
	Ho et al, *Nature* **415** (2002) 180	25333 pairs
HT-Y2H:	Uetz et al, *Nature* **403** (2000) 623	981 pairs
	Ito et al, *PNAS* **98** (2001) 4569	4393 pairs

4 experiments on overlapping PP pairs
→ 2^4 = 16 categories — fully connected Bayes network

Gavin (g)	Ho (h)	Uetz (u)	Ito (i)	# protein pairs	Gold-standard overlap					P(g,h,u,i | pos)	P(g,h,u,i | neg)	L
					pos	neg	sum(pos)	sum(neg)	sum(pos)/ sum(neg)			
1	1	1	0	16	6	0	6	0	-	7.27E-04	0.00E+00	-
1	0	0	1	53	26	2	32	2	16.0	3.15E-03	7.38E-07	4268.3
1	1	1	1	11	9	1	41	3	13.7	1.09E-03	3.69E-07	2955.0
1	0	1	1	22	6	1	47	4	11.8	7.27E-04	3.69E-07	1970.0
1	1	0	1	27	16	3	63	7	9.0	1.94E-03	1.11E-06	1751.1
1	0	1	0	34	12	5	75	12	6.3	1.45E-03	1.85E-06	788.0
1	1	0	0	1920	337	209	412	221	1.9	4.08E-02	7.72E-05	529.4
0	1	1	0	29	5	5	418	227	1.8	6.06E-04	1.85E-06	328.3
0	1	1	1	16	1	1	413	222	1.9	1.21E-04	3.69E-07	328.3
0	1	0	1	39	3	4	421	231	1.8	3.64E-04	1.48E-06	246.2
0	0	1	1	123	6	23	427	254	1.7	7.27E-04	8.49E-06	85.7
1	0	0	0	29221	1331	6224	1758	6478	0.3	1.61E-01	2.30E-03	70.2
0	0	1	0	730	5	112	1763	6590	0.3	6.06E-04	4.13E-05	14.7
0	0	0	1	4102	11	644	1774	7234	0.2	1.33E-03	2.38E-04	5.6
0	1	0	0	23275	87	5563	1861	12797	0.1	1.05E-02	2.05E-03	5.1
0	0	0	0	2702284	6389	2695949	8250	2708746	0.0	7.74E-01	9.95E-01	0.8

Jansen et al, Science 302 (2003) 449

Statistical Uncertainties

Gavin (g)	Ho (h)	Uetz (u)	Ito (i)	# protein pairs	Gold		P(g,h,u,i | pos)	P(g,h,u,i | neg)	L
					pos	neg			
1	1	1	0	16	6	0	7.27E-04	0.00E+00	-
1	0	0	1	53	26	2	3.15E-03	7.38E-07	4268.3
1	1	1	1	11	9	1	1.09E-03	3.69E-07	2955.0
1	0	1	1	22	6	1	7.27E-04	3.69E-07	1970.0
1	1	0	1	27	16	3	1.94E-03	1.11E-06	1751.1
1	0	1	0	34	12	5	1.45E-03	1.85E-06	788.0

1) $L(1111) < L(1001)$

statistical uncertainty: $\Delta N = \sqrt{N+1}$

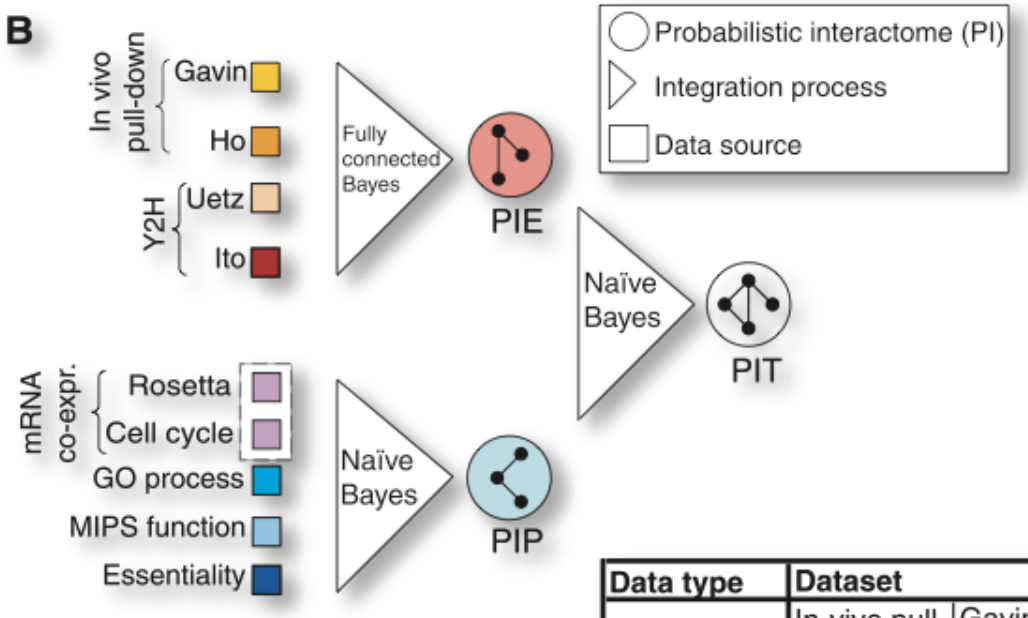
Overlap with all experiments is smaller → larger uncertainty

2) $L(1110)$ = NaN?

Use conservative lower bound → assume 1 overlap with GN
→ $L(1110) \geq 1970$

Jansen et al, Science 302 (2003) 449

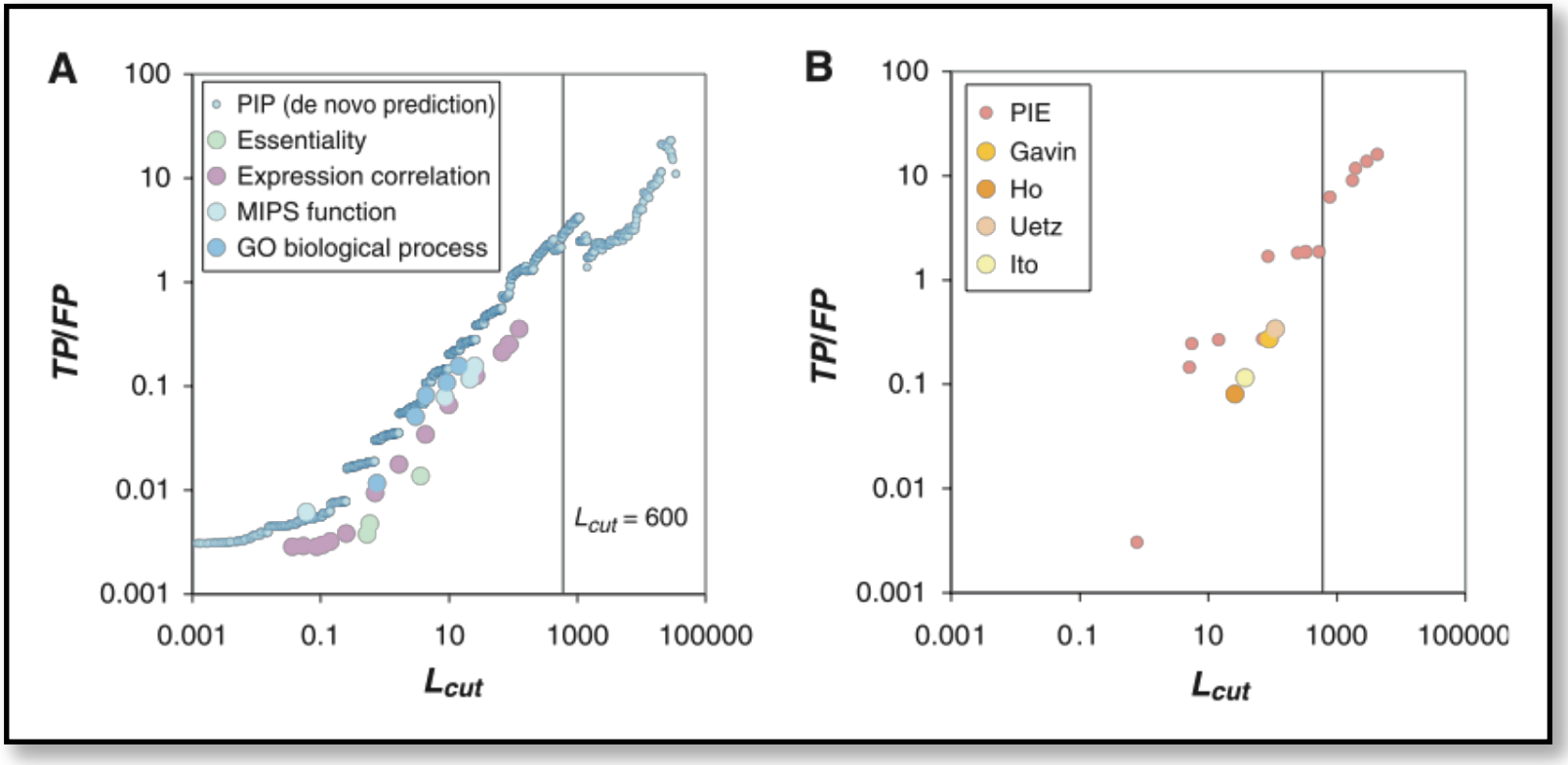
Overview



Data type	Dataset		# protein pairs	Used for ...
Experimental interaction data	In-vivo pull-down	Gavin et al.	31,304	Integration of experimental interaction data (PIE)
		Ho et al.	25,333	
	Yeast two-hybrid	Uetz et al.	981	
		Ito et al.	4,393	
Other genomic features	mRNA Expression	Rosetta compendium	19,334,806	De novo prediction (PIP)
		Cell cycle	17,467,005	
	Biological function	GO biological process	3,146,286	
		MIPS function	6,161,805	
	Essentiality		8,130,528	
Gold standards	Positives	Proteins in the same MIPS complex	8,250	Training & testing
	Negatives	Proteins separated by localization	2,708,746	

Jansen et al, Science 302 (2003) 449

Performance of complex prediction

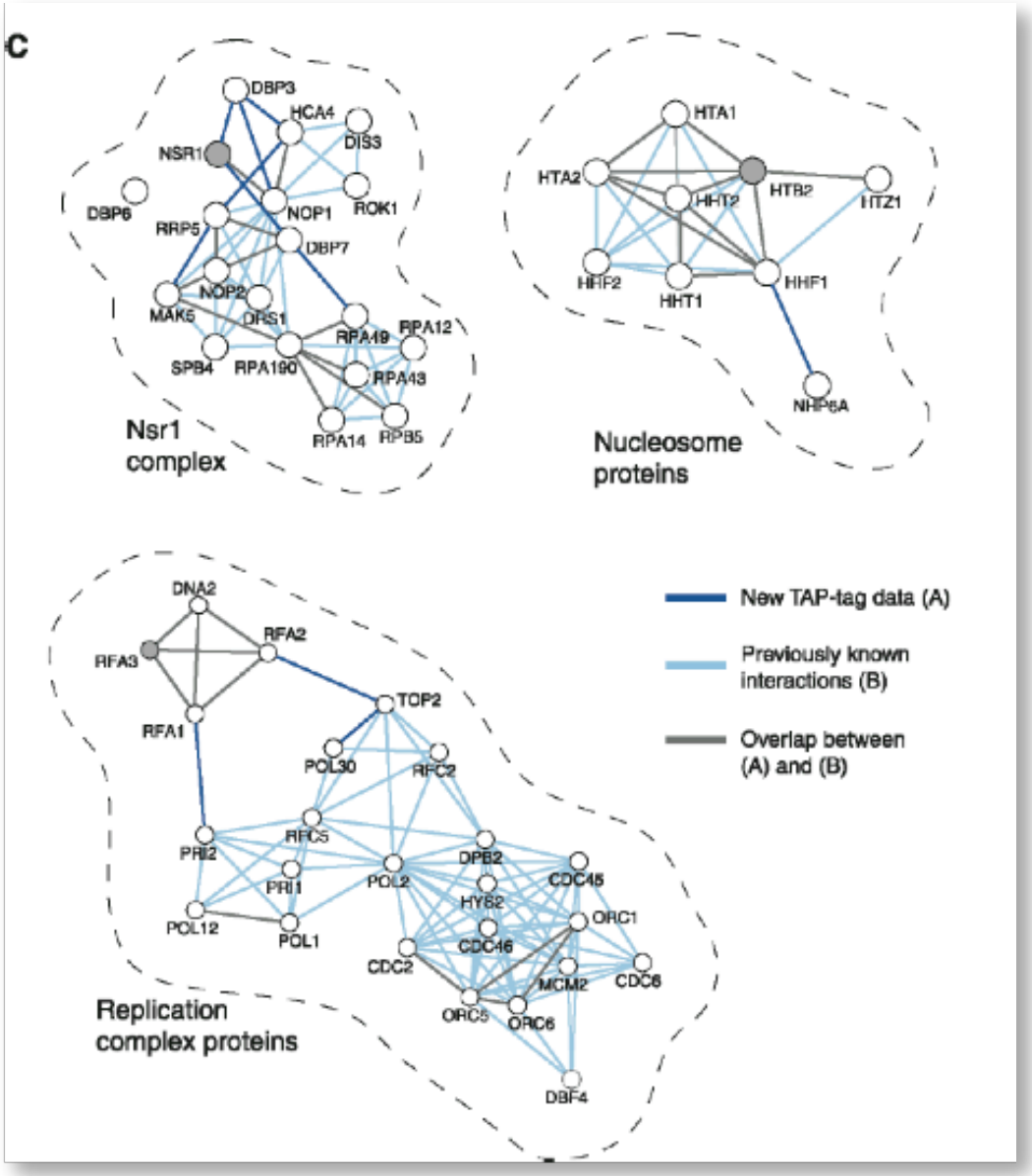


Re-classify Gold standard complexes:
Ratio of true positives to false positives
→ None of the evidences alone was enough

$$\frac{TP}{FP}(L_{cut}) = \frac{\sum_{L>L_{cut}} \mathrm{pos}(L)}{\sum_{L>L_{cut}} \mathrm{neg}(L)}$$

Jansen et al, Science 302 (2003) 449

Verification of Predicted Complexes

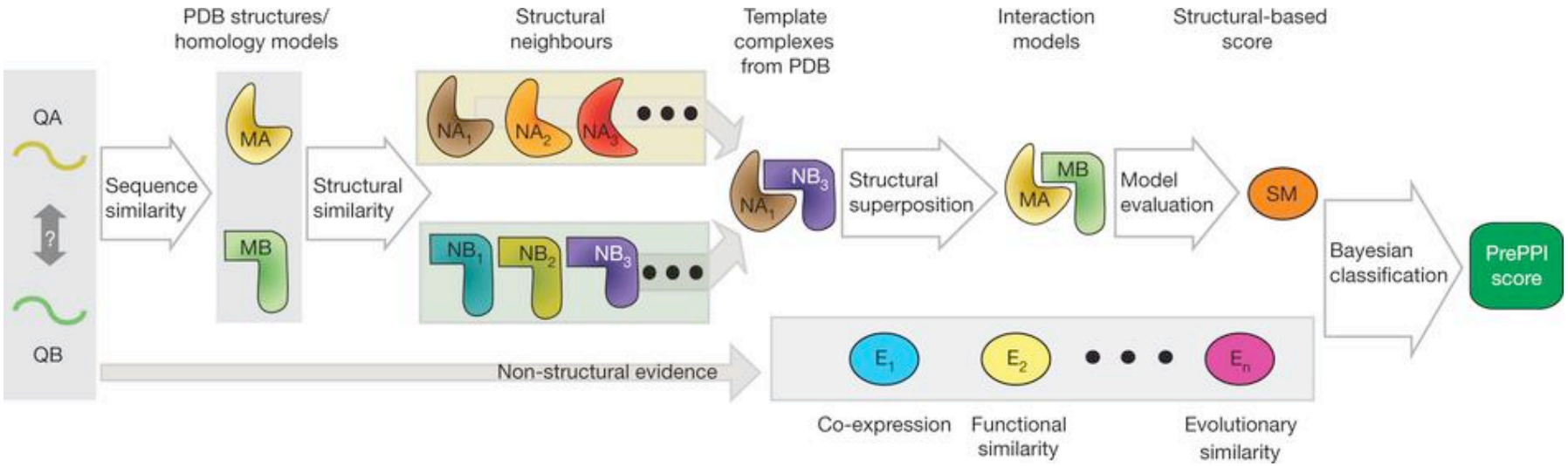


Compare predicted complexes with available experimental evidence and directed new TAP-tag experiments

→ use directed experiments to verify new predictions (more efficient)

Jansen et al, Science 302 (2003) 449

Follow-up work: PrePPI (2012)



Given a pair of query proteins that potentially interact (QA, QB), representative structures for the individual subunits (MA, MB) are taken from the PDB, where available, or from homology model databases.

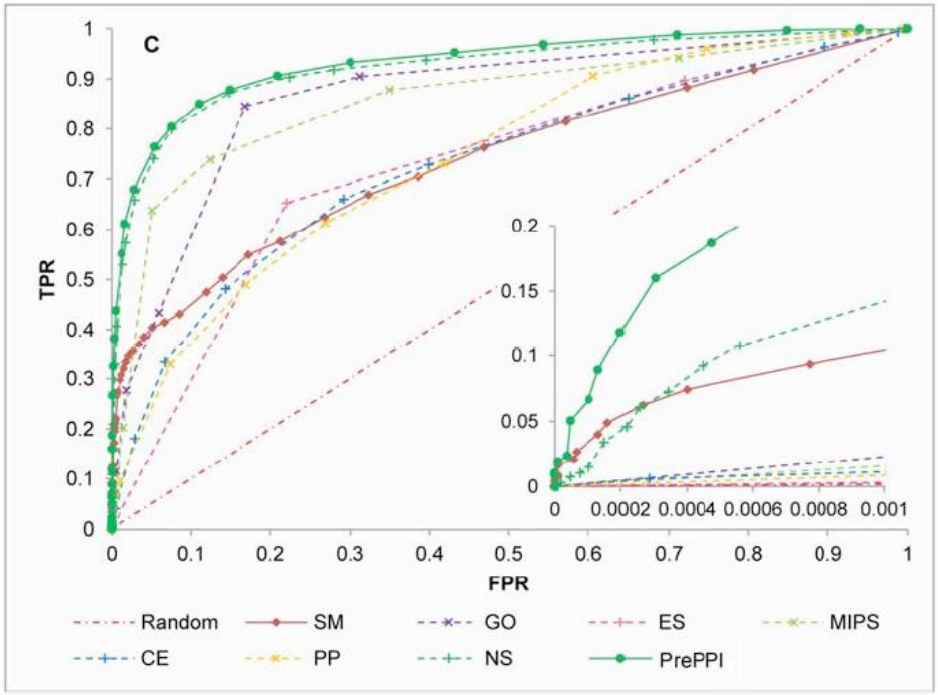
For each subunit we find both close and remote structural neighbours. A 'template' for the interaction exists whenever a PDB or PQS structure contains a pair of interacting chains (for example, NA_1–NB_3) that are structural neighbours of MA and MB, respectively. A model is constructed by superposing the individual subunits, MA and MB, on their corresponding structural neighbours, NA_1 and NB_3.

We assign 5 empirical-structure-based scores to each interaction model and then calculate a likelihood for each model to represent a true interaction by combining these scores using a Bayesian network trained on the HighConfidence and the NonInteracting interaction reference sets.
We finally combine the structure-derived score (SM) with non-structural evidence associated with the query proteins (for example, co-expression, functional similarity) using a naive Bayesian classifier.

Zhang et al, Nature (2012) 490, 556–560

Results of PrePPI

Receiver-operator characteristics (ROC) for predicted yeast complexes.



Examined features:
- structural modeling (SM),
- GO similarity,
- protein essentiality (ES) relationship,
- MIPS similarity,
- co-expression (CE),
- phylogenetic profile (PP) similarity.

Also listed are 2 combinations:
- NS for the integration of all non-structure clues, i.e. GO, ES, MIPS, CE, and PP, and
- PrePPI for all structural and non-structure clues).

This gave 30.000 high-confidence PP interactions for yeast and 300.000 for human.

Zhang et al, Nature (2012) 490, 556–560

Bayesian networks (BN)

Similar to the naive Bayes models, Bayesian networks (BN) also exploit conditional independence properties of the distribution in order to allow a compact and natural representation.

However, they are not restricted to representing distributions satisfying the same strong independence assumptions.

The core of the BN representation is a **directed acyclic graph** (DAG) G whose nodes are the random variables in our domain and whose edges correspond to the direct influence of one node on another

Revised student example as BN

Consider a slightly more complex scenario.

The student‘s grade now depends not only on his/her intelligence but also on the `difficulty` of the course, represented by the random variable D.

Val(D) = {easy, hard}

Our student then asks his professor for a recommendation letter.

The professor is absentminded (typical of professors) and never remembers the names of her students (not typical).

She can look at his/her grade, and writes the letter based only on the grade. The `quality of the letter` is a random variable L.

Val (L) = {strong, weak}

Revised student example: random variables

We therefore now have 5 random variables:

- The student‘s intelligence I
- The course difficulty D
- The grade G
- The student‘s SAT score S
- The quality of the recommendation letter L

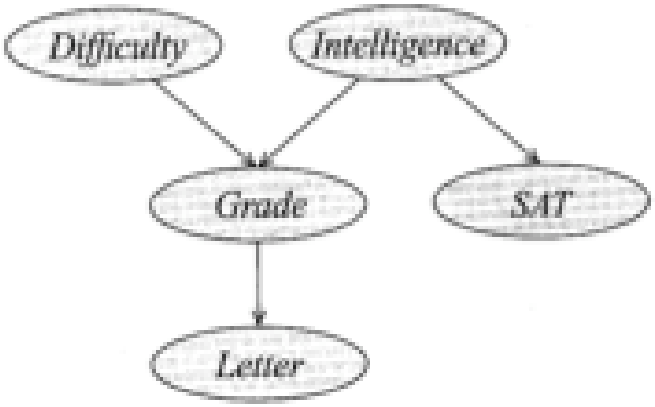
All of the variables except G are binary-values. G has 3 possible values.

Hence, the joint distribution has $2 \times 2 \times 2 \times 2 \times 3 = 48$ entries

Student example revised as BN

The most natural network structure (DAG) for this example may be the one below.

The course difficulty and the student‘s intelligence are determined independently, and before any of the other variables of the model.



The grade depends on both these factors.

The SAT score depends only on the student‘s intelligence.

The quality of the professor‘s recommendation letter depends (by assumption) only on the student‘s grade.

Intuitively, each variable depends directly only on its parents.

Student example revised as BN

The second component of the BN representation is a set of **local probability models** that represent the nature of the dependence of each variable on its parents.

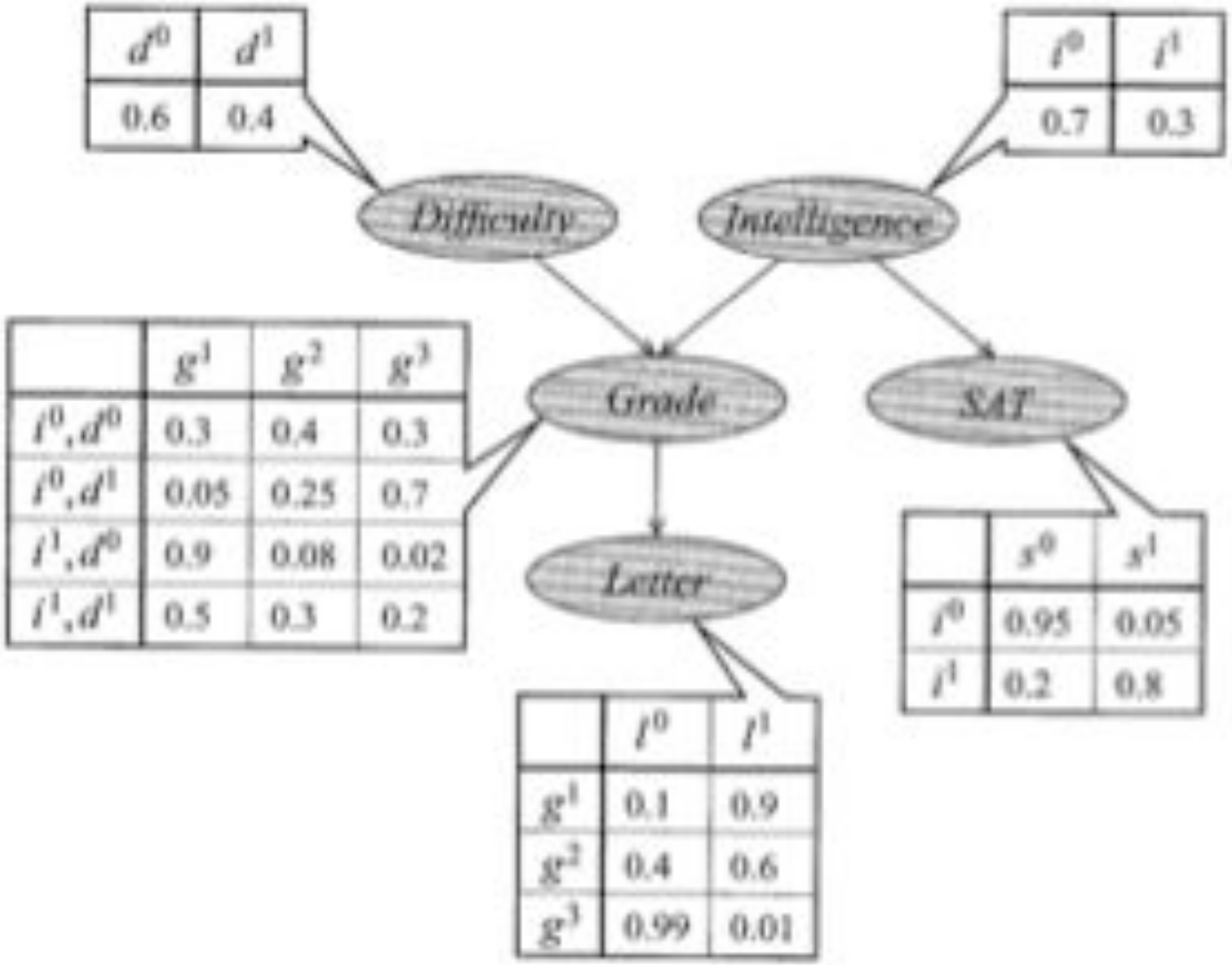
We will reuse P(I) and P(S | I) from p.7.

P(D) represents the distribution of difficult and easy classes.

The distribution over the student‘s grade is a conditional distribution P(G | I,D).

Shown on the next slide is again the structure of the BN together with a choice of the CPDs.

Student example revised as BN



Student example revised as BN

What is the probability of e.g. i^1, d^0, g^2, s^1, l^0?

(the student is intelligent, the course is easy, the probability that a smart student gets a B in an easy class, the probability that a smart students gets a high score on his SAT, and the probability that a student who got a B in the class gets a weak letter.)

The total probability for this is

$$P(i^1, d^0, g^2, s^1, l^0) = P(i^1)\, P(d^0)\, P(g^2 \mid i^1, d^0)\, P(s^1 \mid i^1)\, P(l^0, g^2) =$$
$$= 0.3 \cdot 0.6 \cdot 0.08 \cdot 0.8 \cdot 0.4 = 0.004608$$

We can use the same process for any state in the joint probability space.
This is an example of the **chain rule** for BN:

$$P(I,D,G,S,L) = P(I)\, P(D)\, P(G \mid I,D)\, P(S \mid I)\, P(L \mid G).$$

Basic independencies in BN

In the student example, we used the intuition
that edges represent direct dependence.

E.g. we stated that the professor‘s recommendation letter depends
only on the student‘s grade. There were no direct edges to L except from G.

We can formally express this by a conditional independence statement

$$(L \perp I,D,S \mid G)$$

This means once we know the student‘s grade, our beliefs about the quality
of the letter are not influenced by information about other variables.

In the same way $(S \perp D,G,L \mid I)$

Basic independencies in BN

Now let us consider G.
Is G also indepdendent from all other variables excepts its parents I and D?

Let us consider the case i^1, d^1, a smart student in a difficult test.

Is G indepdent of L in this setting? No!

If we observe l^1 (strong letter), then our probability in g^1 should go up.

Thus we expect $P(g^1 \mid i^1, d^1, l^1) > P(g^1 \mid i^1, d^1)$

(see CPD: right side is 0.5; left side turns out to be 0.712)

→ we do not expect a node to be conditionally independent of all other nodes given its parents.
It can still depend on its descendants as well.

Basic independencies in BN

Can G depend on other nodes than L? No.

E.g. when knowing that the student has high intelligence, knowing his SAT score gives us no additional information that is relevant for predicting his grade.

$(G \perp S \mid I,D)$

In the same way, I is not independent of its descendants G, L or S.
The only nondescendant of I is D.
This makes sense. `Intelligence` and `Difficulty` of a test are indepdent.

$(I \perp D)$

For D, both I and S are nondescendants.

$(D \perp I, S)$

BN semantics

Definition: A Bayesian network structure G is a directed acyclic graph whose nodes represent random variables X_1 , … X_n.

Let $Pa_{X_i}{}^{G}$ denote the parents of X_i in G and $\text{NonDescendants}_{X_i}$ denote the variables in the graph that are not descendants of X_i .

Then G encodes the following set of conditional independence assumptions, called the **local independencies**, and denoted by $I_l(G)$:

For each variable X_i : $(X_i \perp \text{NonDescendants}_{X_i} \mid Pa_{X_i}{}^{G})$

In other words, the local independencies state that each node X_i is conditionally independent of its nondescendants given its parents.

In the student example, the local Markov independencies are precisely the once given before.