Topics in Probabilistic and Statistical Databases

Lecture 2: Representation of Probabilistic Databases

Dan Suciu
University of Washington

Review: Definition

The set of all possible database instances:

$$\boxed{\mathbf{Inst} = \{I_1, I_2, I_3, \ldots, I_N\}}$$

Sample space (Ω)

Definition A *probabilistic database* PDB = (**Inst**, Pr) is a discrete probability distribution:

$$\boxed{\Pr : \mathbf{Inst} \rightarrow [0,1]} \qquad \text{s.t. } \textstyle\sum_{i=1,N} \Pr(I_i) = 1$$

Definition A *possible world* is I s.t. Pr(I) > 0

A *possible tuple* is a tuple $t \in I$, for a possible world I

Representation System

Informally: it is a syntax + semantics that allows us to represent a probabilistic database concisely

Definition A representation system for ProbDB is (**S**,Rep), where **S** is a set of representations and Rep: **S** → (set of PDBs) assigns a probabilistic database to each representation

Review: Disjoint-Independent Databases

<u>Definition</u> A **PDB** is disjoint-independent if for any set T of possible tuples one of the following holds:

- T is an independent set, or
- T contains two disjoint tuples

A disjoint-independent database can be fully specified by:

- all marginal tuple probabilities
- an indication of which tuples are disjoint or independent

Representations Systems for D/I-Databases

- MystiQ's representation
- Trio's xor-, maybe- tuples
- Attribute-level probabilities

D/I Relations in MystiQ

At the schema level:

- Possible worlds key = set of attributes
- Probability = expression on attributes

Constraint at the instance level:

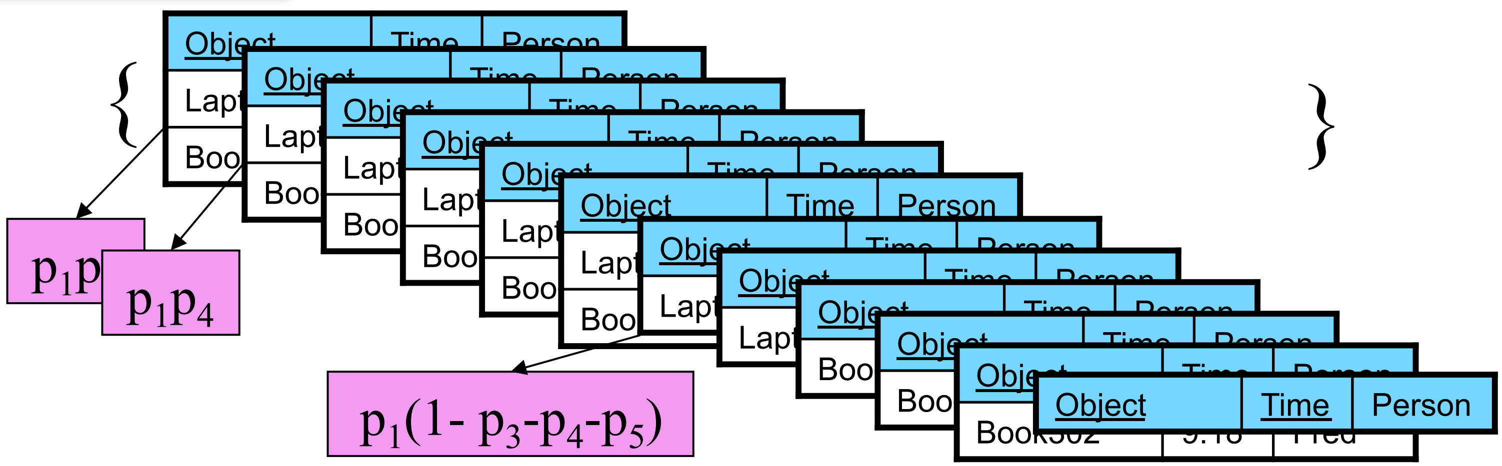
- For each key value, sum of probabilities ≤ 1

Possible worlds key = ?
Attribute expression = ?
Constraint = ?

S =

Object	Time	Person	P
LaptopX77	9:07	John	p_1
LaptopX77	9:07	Jim	p_2
Book302	9:18	Mary	p_3
Book302	9:18	John	p_4
Book302	9:18	Fred	p_5

Rep(S) =



X-Relations in Trio

- Maybe tuple is t?, meaning it may be missing
- X-tuple is <t1, t2, …> meaning it may be any of t1, t2, …

- An X-relation is a collection of X-tuples and maybe-tuples

Maybe- and X-tuples in Trio

S =

ID	Saw(witness, car)		
11	(Amy,Acura):0.8		?
12	(Betty,Acura):0.4	(Betty,Mazda):0.6	

ID	Drives(person, car)	
51	(Hank,Acura):0.6	?

Rep(S) = ?

Attribute-level Uncertainty

- For some attribute A, give a probability distribution on possible values
  - Note: sum of probabilities must be 1
- More generally, for a set of attributes A1, A2, … give a probability distribution on possible values

[Barbara'92]

Attribute-Level Uncertainty

S =

TABLE 1
EXAMPLE PROBABILISTIC RELATION

Key	Independent	Interdependent	Independent
	Deterministic	Stochastic	Stochastic
EMPLOYEE	DEPARTMENT	QUALITY BONUS	SALES
Jon Smith	Toy	0.4 [Great Yes] 0.5 [Good Yes] 0.1 [Fair No]	0.3 [$30–34K] 0.7 [$35–39K]
Fre Jones	Houseware	1.0 [Good Yes]	0.5 [$20–24K] 0.5 [$25–29K]

Rep(S) = ?

Review Query Semantics

Semantics 1: Possible Sets of Answers

A probability distributions on *sets of tuples*

$$\forall A.\ \Pr(Q = A) = \sum_{I \in \mathbf{Inst}.\ Q(I) = A} \Pr(I)$$

Semantics 2: Possible Tuples

A probability function on *tuples*

$$\forall t.\ \Pr(t \in Q) = \sum_{I \in \mathbf{Inst}.\ t \in Q(I)} \Pr(I)$$

The Representation Problem

- How do we represent correlations between tuples in a probdb ?
- How do we represent query answers (views) ?

Main Techniques

- Incomplete databases
  - Concerned with representing *possible worlds*, i.e. no probabilities
- Probabilistic Networks
  - Concerned with representing *correlations*, i.e. no databases

Incomplete Databases

[Imilelinski&Lipsi'1984, Green&Tannen'2006]

Incomplete Databases

Let **Inst =** the set of all possible instances over domain D

Definition An incomplete database is a set of possible worlds

$$\boxed{\mathbf{I} = \{I_1, I_2, I_3, \ldots\} \subseteq \mathbf{Inst}}$$

Definition A representation system is (**S**, Rep) where **S** is a set of representations described by some syntax, and:

$$\boxed{\mathrm{Rep} : \mathbf{S} \rightarrow 2^{\mathbf{Inst}}}$$

Discussion

- We often denote an incomplete database $\{I_1, I_2, \ldots\}$ with $<I_1, I_2, \ldots>$

- This is called an OR-set: the true state can be any $I \in <I_1, I_2, \ldots>$

Rule of Thumb #1

A ProbDB = an Incomplete DB + probabilities

Discussion

- Question: what does < > mean (the *empty* set of worlds) ?

Discussion

- Question: what does < > mean (the *empty* set of worlds) ?

- Answer: inconsistency !
  - We do not allow < >

Examples

- Every traditional database instance I is also an incomplete database $\mathbf{I} = \langle I \rangle$

- The *no-information*, or *zero-information* incomplete database is:

  $$\mathbf{N} = \langle\, I \mid I \in \mathbf{Inst} \,\rangle$$

  (in other words: $\mathbf{N} = \mathbf{Inst}$)

Four Representation Systems

- Codd-tables
- V-tables
- C-tables
- OR-sets

Codd-Tables

T =

SUPPLIER	LOCATION	PRODUCT	QUANTITY
Smith	London	Nails	@
Brown	@	Bolts	@
Jones	@	Nuts	40,000

NULL or ⊥

Rep(T) = ?

V-Tables

T =

COURSE	TEACHER	WEEKDAY
Databases	x	Monday
Programming	y	Tuesday
Databases	x	Thursday
FORTRAN	Smith	z

Rep(T) = ?

"marked nulls":
$\perp_1, \perp_2, \ldots$

C-Tables

T =

SUPPLIER	LOCATION	PRODUCT	con
x	London	Nails	x = Smith
Brown	New York	Nails	$x \neq$ Smith

Any Boolean condition

Rep(T) = ?

Boolean C-Tables: Vars only in Cond

T =

A	B	Cond
a1	b1	$(x=1) \wedge (y=2) \vee (z=1)$
a1	b2	$x=2$
a2	b2	$z=1$

We often state explicitly the domain of each variable:

Dom(X) = {1, 2}
Dom(Y) = {1,2,3}
Dom(Z) = {0,1} . . .

Rep(T) = ?

In G&T's definition all vars are Boolean; minor distinction.

OR-Sets

Review of Nested Relational Algebra NRA:

Types: $\boxed{T ::= \text{baseType} \mid T \times T \mid \{ T \}}$

Operations (in class):…

OR-Sets

Types: $T ::= \text{baseType} \mid T \times T \mid \{ T \} \mid < T >$

Operations (in class):…

OR-Sets Examples

What are their types ? What do they mean ?

- {[Gizmo, <99,110>], [Camera, <10,12,19>]}

- {<3,5>,<3,7>,<5,8,12>}

- {<{<1,2>,<3>},{<4>}>,<{<1>},{<2,3>,<4>}>}

Discussion

Which concepts from incomplete databases where borrowed by the three simple representation systems for ProbDB ?

- MystiQ’s disjoint/independent tables
- Trio’s X-tuples
- Attribute level probabilities

Two Important Properties

- Completeness: can a representation system represent *all* incomplete databases ?

- Closure: can the result of a query also be represented in the same system ?

Closure and Completeness

Assume the domain D is finite

Why ?

Definition A representation system is complete if for any incomplete database **I** there exists a representation S s.t. Rep(S) = **I**.

Which Are Complete ? Why ?

- Codd-Tables ?
- V-Tables ?
- C-Tables ?
- OR-Sets ?

Which Are Complete ? Why ?

- Codd-Tables ? NO: constant cardinality
- V-Tables ? NO: constant cardinality
- C-Tables ? YES
- OR-Sets ? YES

Closure

Definition A representation system is closed w.r.t. a query language Q, if for every representation S and query q in Q, there exists S’ s.t. Rep(S’) = q(Rep(S))

$$
\begin{tikzcd}
\boxed{S} \arrow[r, "q'"] \arrow[d, "\mathrm{Rep}"'] & \boxed{S'} \arrow[d, "\mathrm{Rep}"'] \\
\boxed{\mathrm{Rep}(S)=\{I_1, I_2, \ldots\}} \arrow[r, "q"] & \boxed{\mathrm{Rep}(S')=\{q(I_1), q(I_2), \ldots\}}
\end{tikzcd}
$$

Which Are Closed w.r.t. RA ?

- Codd-Tables ?
- V-Tables ?
- C-Tables ?
- OR-Sets ?

Which Are Closed w.r.t. RA ?

- Codd-Tables ? NO – in class
- V-Tables ? NO – in class
- C-Tables ? YES – in class
- OR-Sets ? YES – in class

Completeness ➔ Closure

- Fact: every complete system is closed w.r.t. the Relational Algebra
  - Why ?

Completeness ← Closure ?

- Challenge: give an example of a system that is closed w.r.t. Relational Algebra but is not complete !

[Das Sarma (unpublished) 2008]

Completeness ← Closure ?

Consider a representation system **S** s.t.

- **S** can represent any deterministic instance <I>
- **S** can represent <{0}, {1}>
- **S** is closed under $\{\Pi, \times, \sigma\}$

Then **S** is complete

PROOF: in class

Lineage

- Lineage = a Boolean expression annotating a tuple that explains why the tuple is there
- Technically: lineage = condition in a Boolean C-table
- A.k.a provenance

[Benjelloun, VLDBJ'2008]

Example

Start from Trio's X-relations:

ID	Saw(Witness, Car)
X1	<[Amy, Mazda], [Amy, Toyota]> ?
X2	<[Betty, Honda]>

ID	Drives(Person, Car)
Y1	<[Jimmy, Mazda], [Jimmy, Toyota]>
Y2	<[Billy, Mazda], [Billy, Honda]>

[Benjelloun, VLDBJ'2008]

Example

Convert to Boolean C-Tables

Saw

Witness	**Car**	**Cond**
Amy	Mazda	X1=1
Amy	Toyota	X1=2
Betty	Honda	X2=1

Drives

Person	**Car**	**Cond**
Jimmy	Mazda	Y1=1
Jimmy	Toyota	Y1=2
Billy	Mazda	Y2=1
Billy	Honda	Y2=2

Q: How do we say
"Amy is a maybe tuple",
"Betty is a certain tuple" ?

Example

Answer:

Dom(X1) = {0,1,2}

Dom(X2) = {1}

Dom(Y1) = {1,2}

Dom(Y2) = {1,2}

[Benjelloun, VLDBJ'2008]

Example

Compute the query q(w,p) :- Saw(w,c),Drives(c,p)

Saw

Witness	Car	Cond
Amy	Mazda	X1=1
Amy	Toyota	X1=2
Betty	Honda	X2=1

Drives

Person	Car	Cond
Jimmy	Mazda	Y1=1
Jimmy	Toyota	Y1=2
Billy	Mazda	Y2=1
Billy	Honda	Y2=2

Witness	Person	Cond
Amy	Jimmy	$X1=1 \land Y1=1 \lor X1=2 \land Y1=2$
Betty	Billy	$X2=1 \land Y2=2$

Uniform Lineage

Call a lineage expression *uniform* if:

- It is in DNF
- All conjuncts have the same number k of literals
- Each literal is of the form X=v

Representation of uniform lineage: add 2k columns !

Example

Compute the query q(w,p) :- Saw(w,c),Drives(c,p)

Saw

Witness	Car	X	V
Amy	Mazda	X1	1
Amy	Toyota	X1	2
Betty	Honda	X2	1

Drives

Person	Car	Y	W
Jimmy	Mazda	Y1	1
Jimmy	Toyota	Y1	2
Billy	Mazda	Y2	1
Billy	Honda	Y2	2

Witness	Person	X	V	Y	W
Amy	Jimmy	X1	1	Y1	1
Amy	Jimmy	X1	2	Y1	2
Betty	Billy	X2	1	Y2	2

$$X1=1 \wedge Y1=1 \vee X1=2 \wedge Y1=2$$

Summary of Incomplete DBs

- Main goal: specify a set of possible worlds
  - Very relevant to ProbDBs !
- Key concepts: closure and completeness
  - Turn out to be equivalent, except trivial cases
  - Boolean C-tables closed&complete; others not
- Lineage: important tool in ProbDB, is derived from C-tables
- Some open questions next…

Want to tell & show?
➔ Email by Wed.

Open Questions in Incomplete/ Probabilistic Databases

- Variables
- OWA v.s. CWA
- Certain tuples, possible tuples
- Partial information order
- Strong v.s. weak representation systems

Homework: pick one and apply to probdbs.
Tell us next time your thoughts

Variables as Attribute Values

T =

A	B
⊥	b1
a2	b1

Rep(T) =

A large, or infinite set

ProbDBs don't use variables as attribute values

What if we allow ProbDBs to have variables ?

OWA v.s. CWA

- CWA: if R(a,b,c) is not mentioned in the knowledge/data base, then ¬R(a,b,c) is assumed

- OWA: ¬R(a,b,c) is inferred only if it is explicitly stated in the knowledge/data base

Which one is standard semantics in DB ? And in KR ?

OWA v.s. CWA in Incomplete DB

T =

A	B
⊥	b1
a2	b1

CWA: Rep(T) =

A	B
a1	b1
a2	b1

A	B
a1	b1

A	B
a3	b1
a2	b1

A	B
a4	b1
a2	b1

. . .

OWA: Rep(T) =

A	B
a1	b1
a2	b1

A	B
a1	b1
a2	b1
. .	. .

. . .

A	B
a1	b1

A	B
a1	b1
. .	. .

. . .

What would OWA mean for ProbDBs ?

Certain v.s. Possible Tuples

Consider an incomplete database $\mathbf{I} = \{I_1, I_2, I_3, \ldots\}$

Definition The *certain tuples* $\square\mathbf{I} = I_1 \cap I_2 \cap I_3 \cap \ldots$

Definition The *possible tuples* $\Diamond\mathbf{I} = I_1 \cup I_2 \cup I_3 \cup \ldots$

What do certain/possible tuples correspond to in ProbDB ?

Certain v.s. Possible Tuples

- Two incomplete databases **I**, **J** are *equivalent* if □**I** = □**J**
- Two incomplete databases **I**, **J** are *equivalent w.r.t. a query language Q* if forall q in Q, □q(**I**) = □q(**J**)
- These notions are used to define "weak representation systems" (see [AHV])..

Are there similar notions of equivalences between ProbDBs ?

Partial Information Order

Let $(D, \leq)$ be an ordered set.

Consider two sets $A=\{a_1,\ldots,a_m\}$, $B=\{b_1,\ldots,b_n\}$.

When can we say $A \leq B$?

Definition [Smythe or upper] $A \leq^\sharp B$ if $\forall b_j\, \exists a_i: a_i \leq b_j$

Definition [Hoare or lower] $A \leq^\flat B$ if $\forall a_i\, \exists b_j: a_i \leq b_j$

Definition [Plotkin or convex] $A \leq^\natural B$ if $A \leq^\sharp B$ and $A \leq^\flat B$

Partial Information Order

Let's order OR-sets

- Values of base type: $a \leq a$ and $\perp \leq a$
- Records: $[x,y] \leq [u,v]$ when ?

- OR-sets $<a1,a2,a3> \;\leq\; <b1,b2>$ when ?

- Sets: $\{a1,a2,a3\} \leq \{b1, b2\}$ when ?

Partial Information Order

Let’s order OR-sets

- Values of base type: $a \leq a$ and $\perp \leq a$
- Records: $[x,y] \leq [u,v]$ when ?
  - When $x \leq u$ and $y \leq v$
- OR-sets $\langle a1,a2,a3 \rangle \leq \langle b1,b2 \rangle$ when ?
  - Smythe
- Sets: $\{a1,a2,a3\} \leq \{b1, b2\}$ when ?
  - Hoare (for OWA), or equality (for CWA)

Partial Information Order

What is the partial information order on ProbDBs ?

Probabilistic Networks

Probabilistic Models

Problem setting:

- Given m random variables $V_1, \ldots, V_m$
- Each with domain D (same for all V_i)
- A probability space over D^m: Pr: $D^m \rightarrow [0,1]$, st $\sum_{v1,\ldots,vm \text{ in } D} Pr(v_1,\ldots,v_m) = 1$
- Called the *joint probability distribution*

Problem: give a compact representation of Pr

Example

V1	V2	P
1	1	0.06
1	2	0.14
2	1	0.24
2	2	0.56

A "disjoint" probabilistic database

Here m=2, but in general m is large, need compact rep.

Background

- Marginal probability:

$$\boxed{Pr(V_i=a) = \sum_{v1,\ldots,vm \text{ in } D,\ vi = a} Pr(v_1,\ldots,v_m)}$$

- What does this mean ? $Pr(V_i)$, $Pr(V_i V_j)$
- Conditional prob: $\boxed{Pr(E \mid F) = Pr(EF)/Pr(F)}$
- Independence: $Pr(EF) = Pr(E) * Pr(F)$
  Equivalently: $Pr(E \mid F) = Pr(E)$
- Conditional indep: $Pr(E,F \mid G)= Pr(E|G)*Pr(F|G)$

Independence

V1	V2	P
1	1	0.06
1	2	0.14
2	1	0.24
2	2	0.56

V1	P
1	0.2
2	0.8

V2	P
1	0.3
2	0.7

Marginal probabilities

Are V1, V2 independent, i.e. Pr(V1,V2) = Pr(V1)*Pr(V2) ?
Note: need to check 4 equalities (why ?)

Factored Representation

More general: if P(V1=a,V2=b) = p(a) * q(b) forall a,b
then V1, V2 are independent

V1	V2	P
a_1	b_1	p_1q_1
a_1	b_2	p_1q_2
a_2	b_1	p_2q_1
a_2	b_2	p_2q_2
. . .	. . .	

=

V1	P
a_1	p_1
a_2	p_2
. . .	

$\times$

V2	P
b_1	q_1
b_2	q_2
. . .	

Factors

1^{st} Connection to ProbDBs: Variable → Attribute

- Every joint distribution over variables V_1, …, V_m corresponds to a trivial probabilistic table with attributes V_1, …, V_m
  – What's "trivial" about it ?

1st Connection to ProbDBs: Variable ← Attribute

- Conversely: each block in a disjoint/ independent relation defines a joint distribution on the values of its attribute

Object	Time	Person	P
LaptopX77	9:07	John	0.5
		Jim	0.5
Book302	9:18	Mary	0.2
		John	0.4
		Fred	0.4

Independence

TABLE 1
EXAMPLE PROBABILISTIC RELATION

Key	Independent	Interdependent	Independent
	Deterministic	Stochastic	Stochastic
EMPLOYEE	DEPARTMENT	QUALITY BONUS	SALES
Jon Smith	Toy	0.4 [Great Yes] 0.5 [Good Yes] 0.1 [Fair No]	0.3 [$30–34K] 0.7 [$35–39K]
Fre Jones	Houseware	1.0 [Good Yes]	0.5 [$20–24K] 0.5 [$25–29K]

Independence

Employee	**Department**	**Quality**	**Bonus**	**Sales**	**P**
John Smith	Toy	Great	Yes	30k-34k	0.4*0.3
John Smith	Toy	Good	Yes	30k-34k	0.5*0.3
John Smith	Toy	Fair	No	30k-34k	0.1*0.3
John Smith	Toy	Great	Yes	35k-39k	0.4*0.7
John Smith	Toy	Good	Yes	35k-39k	0.5*0.7
John Smith	Toy	Fair	No	35k-39k	0.1*0.7
Fre Jones	Houseware	Good	Yes	20k-24k	0.5
Fre Jones	Houseware	Good	Yes	24k-29k	0.5

Independence

Factors are disjoint/indep. Tables !

Employee	**Department**	**Quality**	**Bonus**	**P**
John Smith	Toy	Great	Yes	0.4
John Smith	Toy	Good	Yes	0.5
John Smith	Toy	Fair	No	0.1
Fre Jones	Houseware	Good	Yes	1

$\bowtie$

Employee	**Department**	**Sales**	**P**
John Smith	Toy	30k-34k	0.3
John Smith	Toy	35k-39k	0.7
Fre Jones	Houseware	20k-24k	0.5
Fre Jones	Houseware	24k-29k	0.5

Rule of Thumb #2

Correlated table =
  Independent tables + join

Same principle as in traditional schema normalization

- Decompose big table into small tables with independent attributes
- Big table = a view (single join !) over the small tables

Database 101

- Assume that (Quality, Bonus) is independent from (Sales)
- Drop the **P** column → a traditional table (no more probabilities)

Question What ‘functional dependency’ holds in this table ?

Database 101

- Assume that (Quality, Bonus) is independent from (Sales)
- Drop the **P** column → a traditional table (no more probabilities)

Question What ‘functional dependency’ holds in this table ?

A Multivalued FD !:

Emp, Dept ↠ Quality, Bonus Sales

Rule of Thumb #3

Factor decomposition =
MVD decomposition + probability identities

Conditional Independence

V1	V2	W	P
1	1	a	0.03
1	2	a	0.07
2	1	a	0.12
2	2	a	0.28
1	1	b	0.125
1	2	b	0.125
2	1	b	0.125
2	2	b	0.125

Are V1, V2 independent given W, i.e.
Pr(V1,V2 | W) = Pr(V1 | W) * Pr(V2 | W) ?

Factors

Conditional Independence

V1	V2	W	P
1	1	a	0.03
1	2	a	0.07
2	1	a	0.12
2	2	a	0.28
1	1	b	0.125
1	2	b	0.125
2	1	b	0.125
2	2	b	0.125

P(W=a)=0.5; P(-- | W=a):

V1	V2	P
1	1	0.06
1	2	0.14
2	1	0.24
2	2	0.56

P(W=b)=0.5; P(-- | W=b):

V1	V2	P
1	1	0.25
1	2	0.25
2	1	0.25
2	2	0.25

They: they are conditional independent

Conditional Independence

V1	V2	W	P
1	1	a	0.03
1	2	a	0.07
2	1	a	0.12
2	2	a	0.28
1	1	b	0.125
1	2	b	0.125
2	1	b	0.125
2	2	b	0.125

$=$

W	P
a	0.5
b	0.5

$\bowtie$

V1	<u>W</u>	P
1	a	0.2
2	a	0.8
1	b	0.5
2	b	0.5

$\bowtie$

V2	<u>W</u>	P
1	a	0.3
2	a	0.7
1	b	0.5
2	b	0.5

Representing Conditional Independent Vars

More general: if P(V1=a,V2=b,W=c) = p(a,c) * q(b,c) forall a,b,c then V1, V2 are independent given c

V1	V2	W	P
a_1	b_1	c_1	$p_{11}q_{11}$
a_1	b_2	c_1	$p_{11}q_{21}$
a_2	b_1	c_1	$p_{21}q_{11}$
a_2	b_2	c_1	$p_{21}q_{21}$
. .	. .	c_2	$P_{12}q_{12}$
. .	. .		

=

W	P
c_1	r_1
c_2	r_2
. . .	

$\bowtie$

V1	W	P
a_1	c_1	p_{11}/P_1
a_2	c_1	p_{21}/P_1
. .	c_2	p_{12}/P_2
. .		

$\bowtie$

V2	W	P
b_1	c_1	q_{11}/Q_1
b_2	c_1	q_{21}/Q_1
. .	c_2	q_{12}/Q_2
. .		

Summary of Prob Networks

- (Conditional) indep. = MVDs + prob ident.
- Factored decomposition = base tables + joins
- Next time:
  - Discuss the ***networks*** in probabilistic networks; WS-decomposition, U-relations
  - Partial/approximate representations
  - Begin query evaluation
- Send email by Wed. if want to show&tell