PORTIA *Privacy, Obligations, and Rights in Technologies of Information Assessment*

Privacy-Preserving Bayesian Network Learning and Other Recent Results in Privacy-Preserving Data Mining

Rebecca Wright
Computer Science Department
Stevens Institute of Technology
www.cs.stevens.edu/~rwright

IPAM, UCLA

25 October, 2006

STEVENS Institute of Technology

(*Includes joint work with Zhiqiang Yang, Geetha Jagannathan, and Sheng Zhong*).

Overview

- Intro: privacy, privacy-preserving data mining
- Bayesian networks
- Privacy-preserving Bayesian network structure computation
- Using privacy-preserving data mining

Erosion of Privacy

"You have zero privacy. Get over it."

- Scott McNealy, 1999

- Changes in technology are making privacy harder.
  - increased use of computers and networks
  - reduced cost for data storage
  - increased ability to process large amounts of data
- Becoming more critical as public awareness, potential misuse, and conflicting goals increase.

The Data Revolution

- We are in the midst of a data revolution fueled by the perceived, actual, and potential usefulness of the data.
- Most electronic and physical activities leave some kind of data trail. These trails can provide useful information to various parties.
- However, there are also concerns about appropriate handling and use of sensitive information.
- Privacy-preserving methods of data handling seek to provide sufficient privacy as well as sufficient utility.

Abuses of Sensitive Data

- Identity theft
- Loss of employment, health coverage, personal relationships
- Unfair business advantage
- Potential aid to terrorist plots

Surveillance and Data Mining

- Analyze large amounts of data from diverse sources.
- Law enforcement and homeland security:
  - detect and thwart possible incidents before they occur
  - recognize that an incident is underway
  - identify and prosecute criminals/terrorists after incidents occur
- Other applications as well:
  - Biomedical research
  - Marketing, personalized customer service

Privacy-Preserving Data Mining

Allow multiple data holders to collaborate to compute important information while protecting the privacy of other information.

- Security-related information
- Public health information
- Marketing information
- etc.

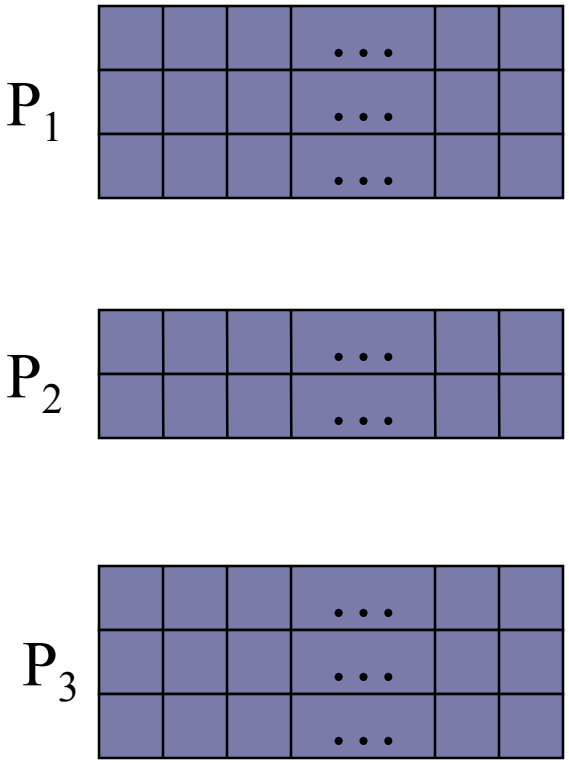
Technological tools include cryptography, data perturbation and sanitization, access control, inference control, trusted platforms.

Advantages of Privacy Protection

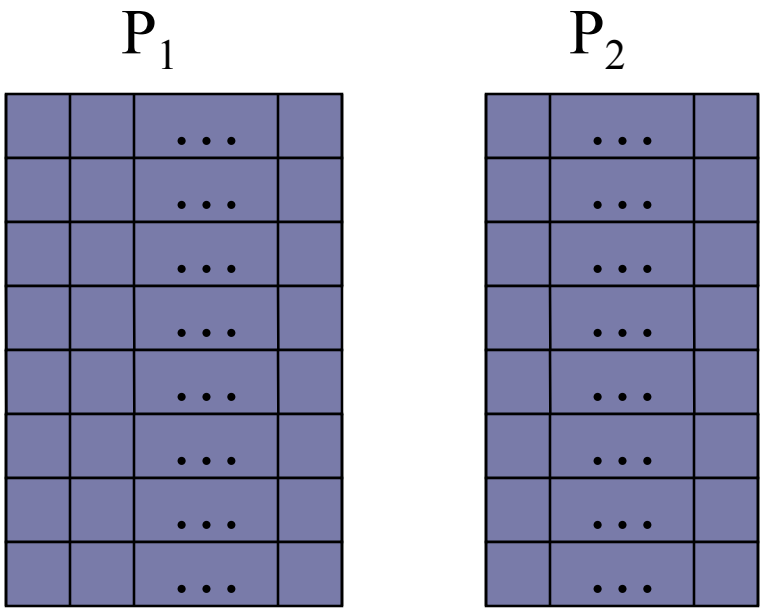
- protection of personal information
- protection of proprietary or sensitive information
- enables collaboration between different data owners (because they may be more willing or able to collaborate if they need not reveal their information)
- compliance with legislative policies (e.g., HIPAA, EU privacy directives)

Models for Distributed Data Mining, I

- Horizontally Partitioned

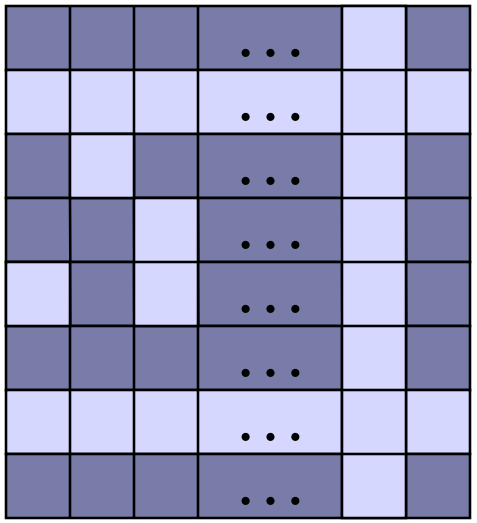


- Vertically Partitioned



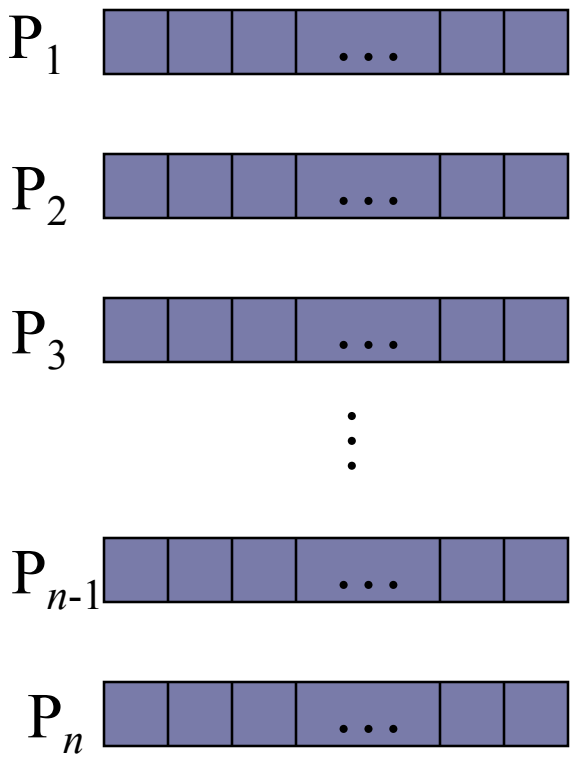
Models for Distributed Data Mining, II

- Arbitrarily partitioned

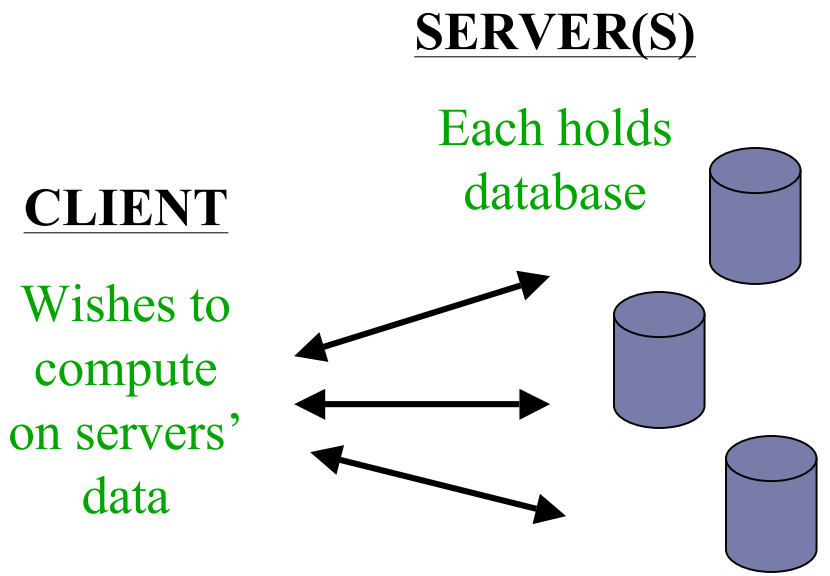


Models for Distributed Data Mining, III

- Fully Distributed



- Client/Server(s)

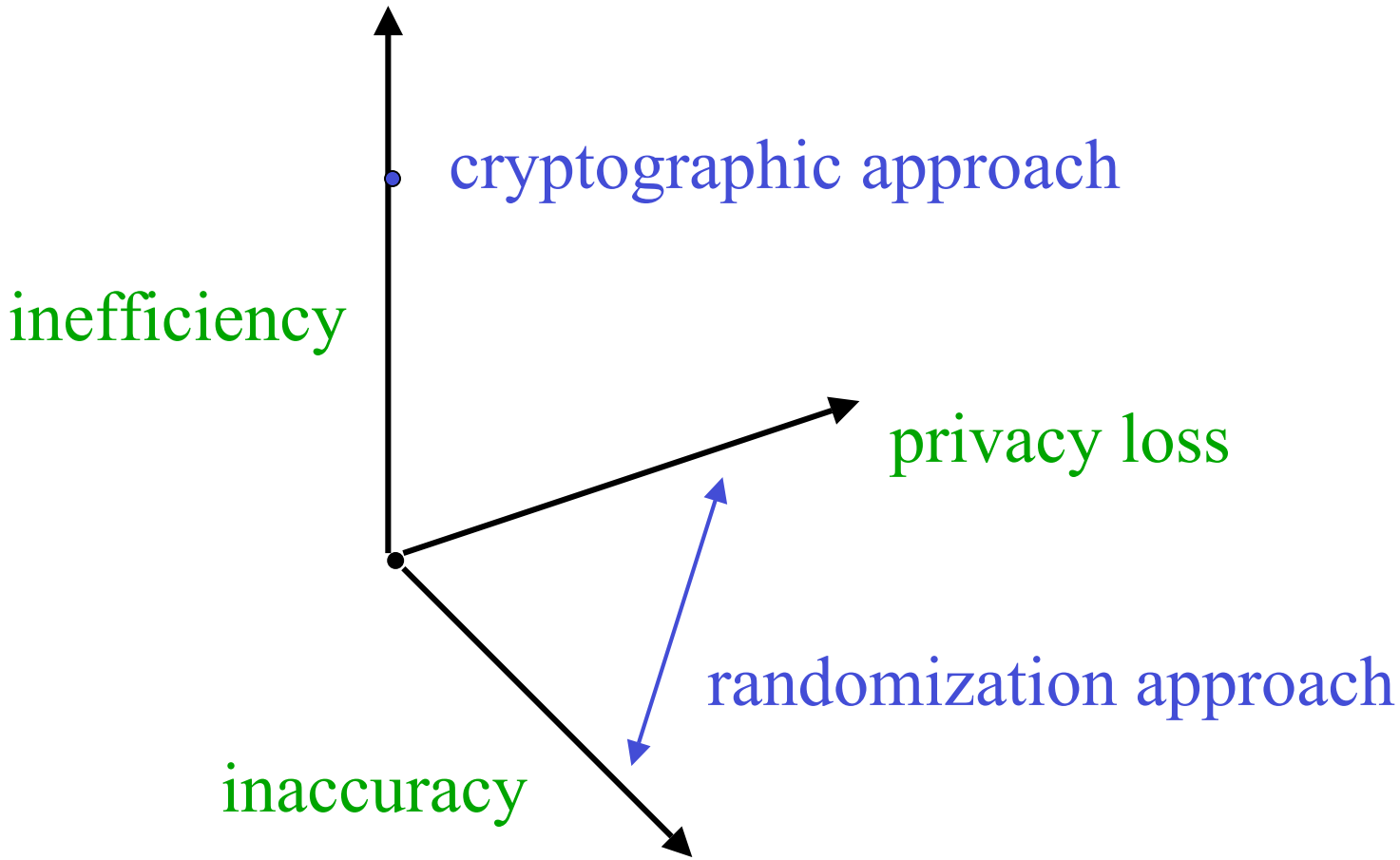


Approaches to PPDM

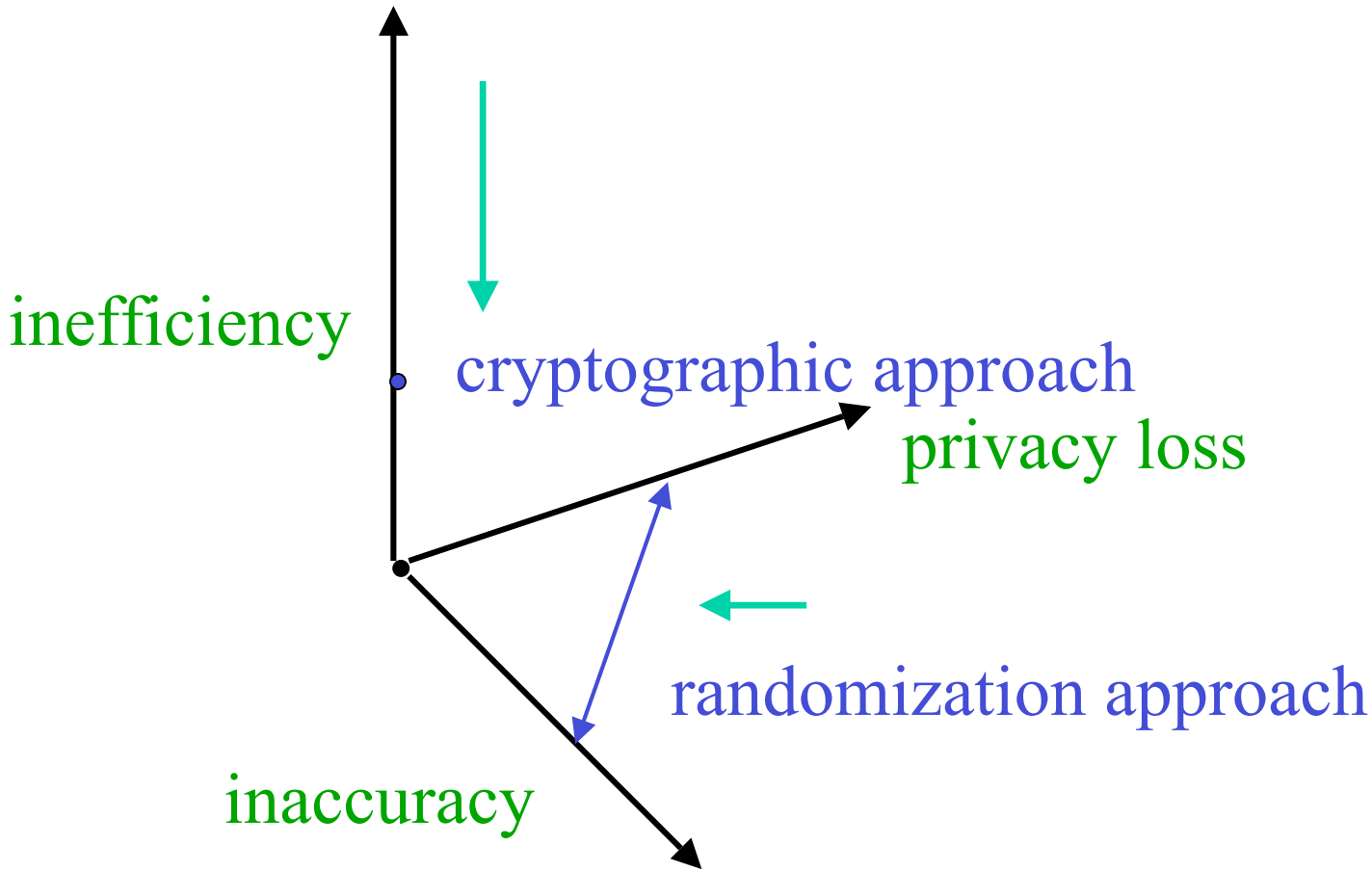
First solutions were introduced in 2000, taking different approaches.

- [LP00]: using cryptography, computes ID3 decision trees for data held by two parties, provably leaking nothing else.
- [AS00]: using random data perturbation, computes reasonably accurate ID3 decision trees for data held by two parties, while obscuring original data.

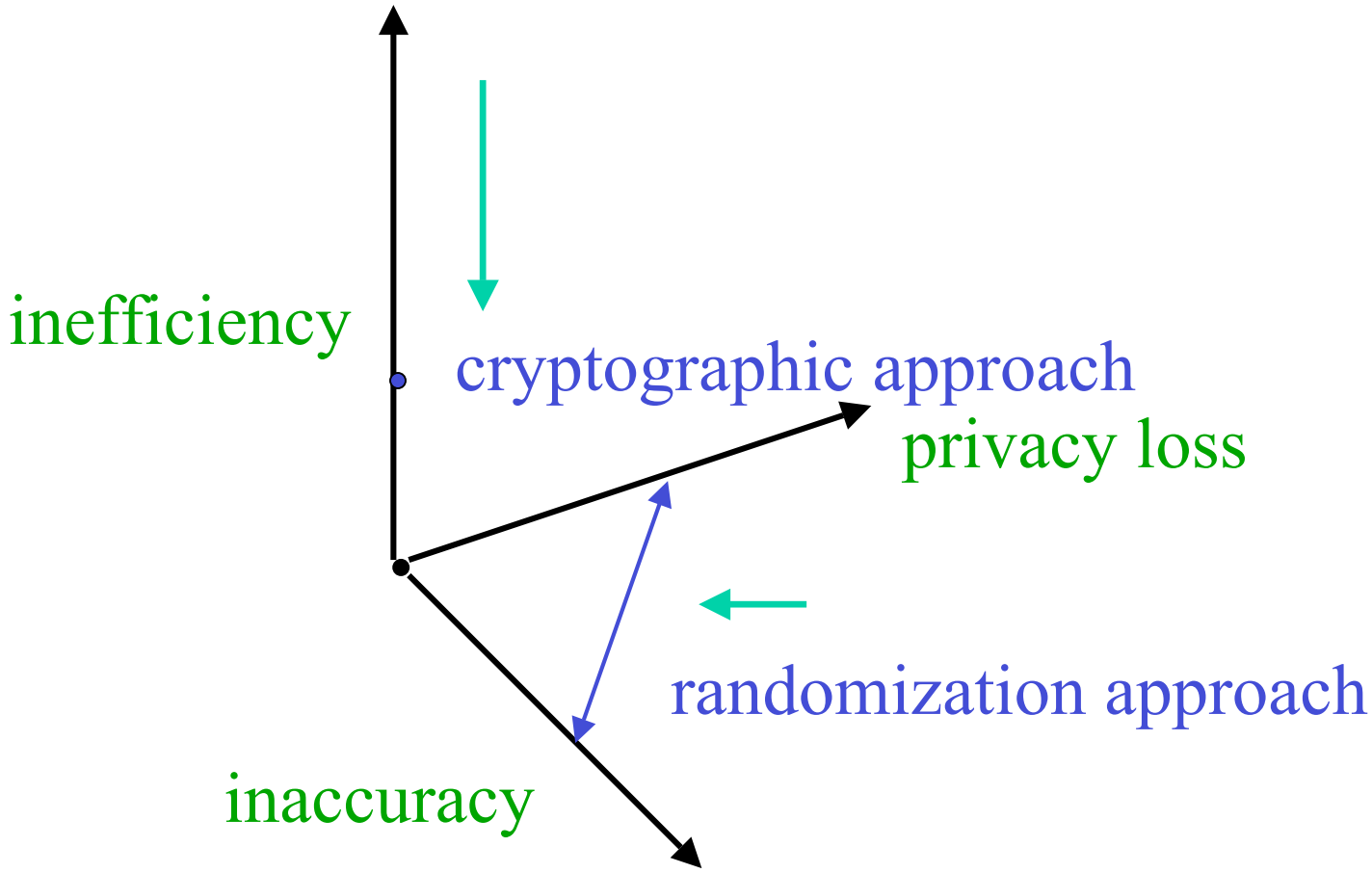
Cryptography vs. Randomization



Cryptography vs. Randomization



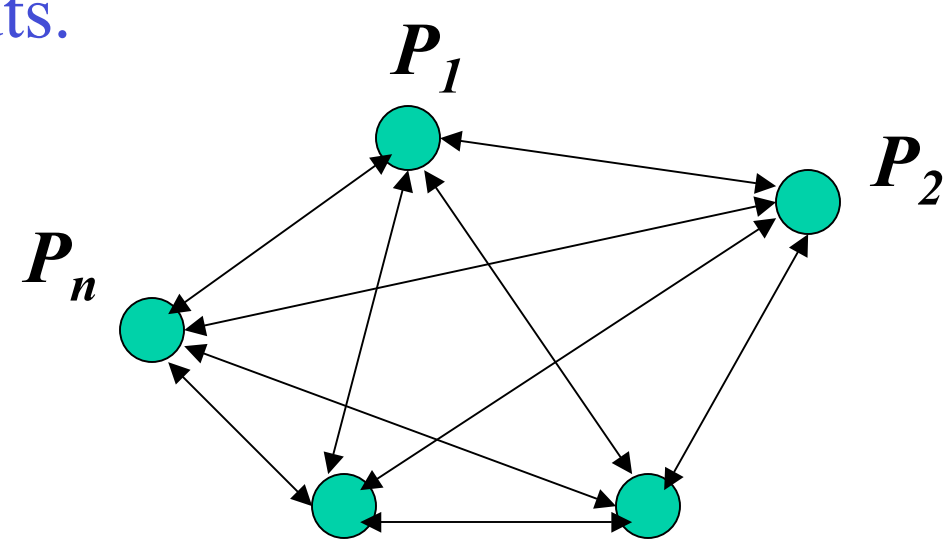
Cryptography vs. Randomization



Utility — related to accuracy but also generality — is also important.

Secure Multiparty Computation

- Allows *n* players to privately compute a function *f* of their inputs.



- Overhead is polynomial in size of inputs and complexity of *f* [Yao86, GMW87, BGW88, CCD88, ...]
- In theory, can solve any private distributed data mining problem. In practice, not efficient for large data.

Our PPDM Work

Our work takes the cryptographic approach.

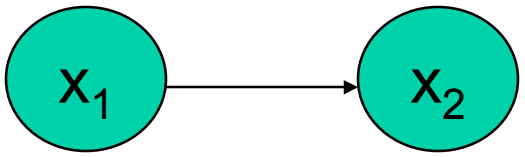
- [WY04, YW05, YW06]: privacy-preserving construction of Bayesian networks from vertically partitioned data.
- [YZW05]: privacy-preserving frequency mining in the fully distributed model (enables naïve Bayes classification, decision trees, and association rule mining).
- [JW05, JPW06]: privacy-preserving clustering: *k*-means clustering for arbitrarily partitioned data and a divide-and-merge clustering algorithm for horizontally partitioned data.

Bayesian Networks

- A Bayesian network (BN) is a graphical model that encodes probabilistic relations among variables.
- Knowledge structure for representing knowledge about uncertain variables
- Computational architecture for computing posterior probabilities given evidences about selected nodes

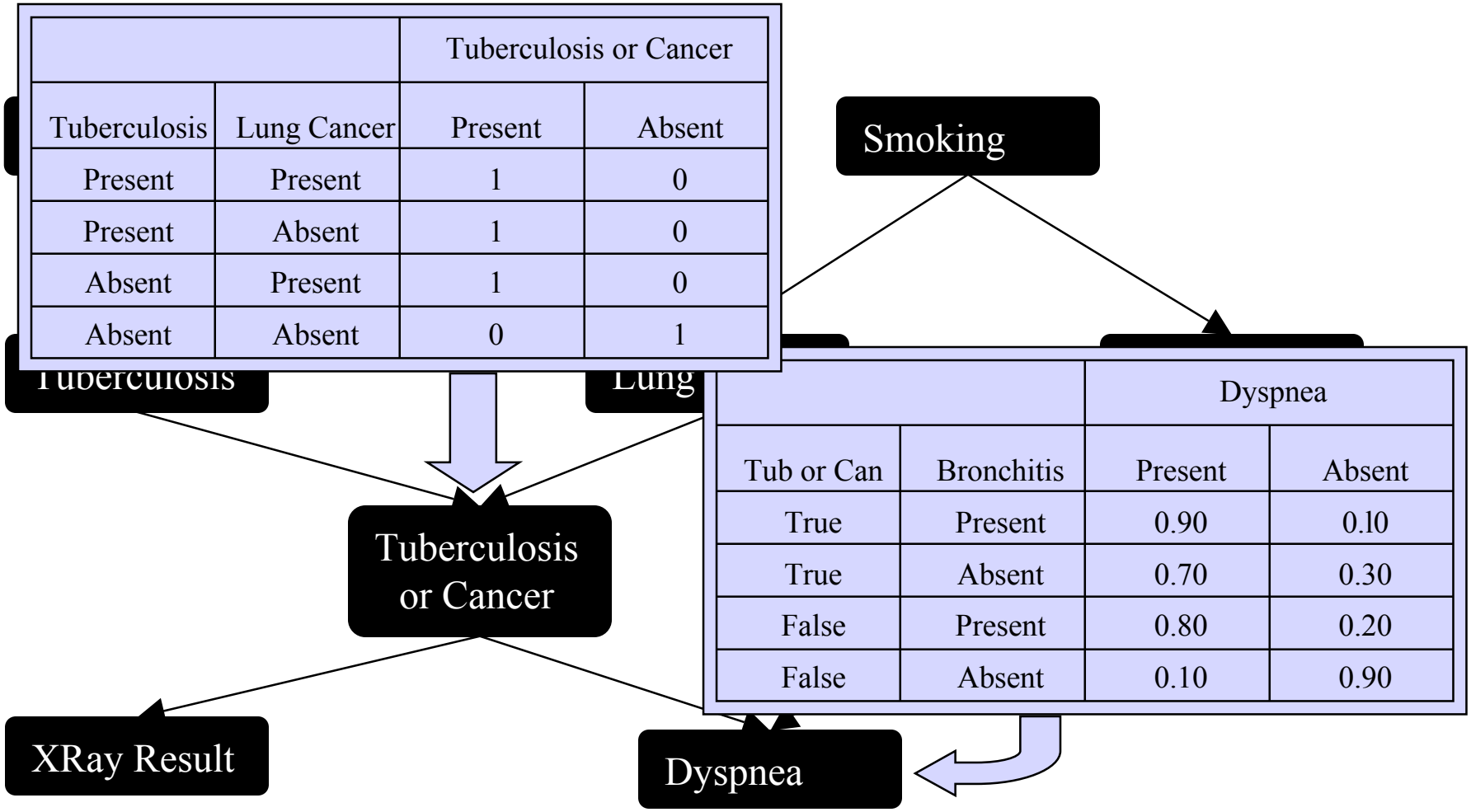
Bayesian Networks, cont'd

- A Bayes network is: (B_s, B_p)
- $B_s = (V, E)$ is a directed acyclic graph, each node in V represents a variab
  probabilistic relationshi
- B_p denotes the local pro
  among variables.



x_1	x_2	Prob(x_2 | x_1)
0	0	0.1
0	1	0.9
1	0	0.3
1	1	0.7

Example: Medical Diagnostics



		Tuberculosis or Cancer	
Tuberculosis	Lung Cancer	Present	Absent
Present	Present	1	0
Present	Absent	1	0
Absent	Present	1	0
Absent	Absent	0	1

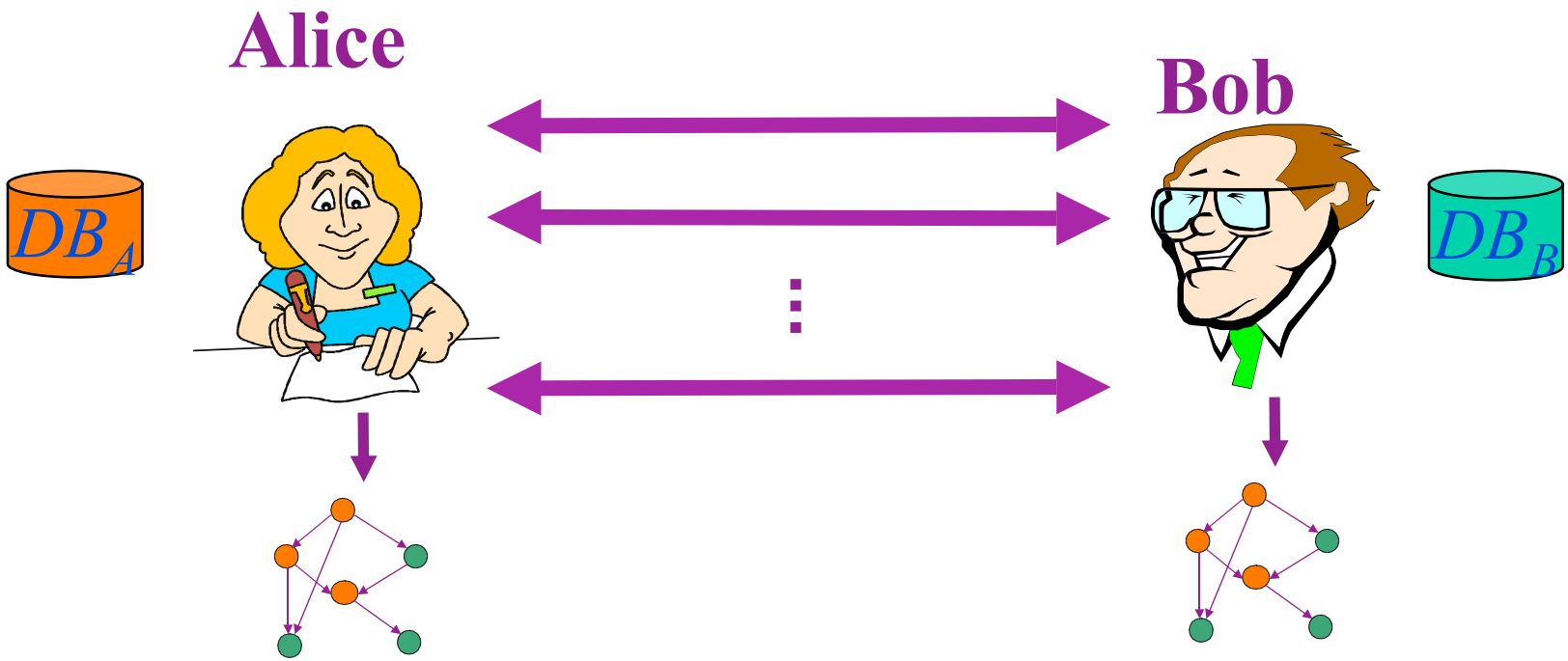
		Dyspnea	
Tub or Can	Bronchitis	Present	Absent
True	Present	0.90	0.10
True	Absent	0.70	0.30
False	Present	0.80	0.20
False	Absent	0.10	0.90

Bayes Network Applications

- Industrial
  - Processor Fault Diagnosis - by Intel
  - Auxiliary Turbine Diagnosis - GEMS by GE
  - Diagnosis of space shuttle propulsion systems - VISTA by NASA/Rockwell
- Military
  - Automatic Target Recognition - MITRE
  - Autonomous control of unmanned underwater vehicle - Lockheed Martin
  - Assessment of Intent
- Medical Diagnosis
  - Internal Medicine
  - Pathology diagnosis - Intellipath by Chapman & Hall
  - Breast Cancer Manager with Intellipath
- Commercial
  - Financial Market Analysis
  - Information Retrieval
  - Software troubleshooting and advice - Windows 95 & Office 97

Privacy-Preserving Bayes Networks

Goal: Cooperatively learn Bayesian network structure on the combination of DB_A and DB_B, ideally without either party learning anything except the Bayesian network structure itself.



K2 Algorithm for BN Learning

- Determining the best BN structure for a given data set is NP-hard, so heuristics are used in practice.
- The K2 algorithm [CH92] is a widely used BN structure-learning algorithm, which we use as the starting point for our solution.
- Considers nodes in sequence. Adds new parent that most increases a score function *f*, up to at most *u* parents per node.
- Number of nodes/variables: *m*
- Number of records: *n*

K2 Algorithm

Pred*(i)*: set of possible parents of node *i*.

π_i : set of current parents of node *i*.

For *i* = *1* to *m*

{

$\pi_i = \varnothing$

KeepAdding = true

While KeepAdding and $|\pi_i| < u$

{

Determine which element of $f(i, \pi_i)$, $f(i, \pi_i \cup \{z\})$ for $z \in$ Pred(*i*) – π_i yields the maximum score

If $f(i, \pi_i)$ is the maximum score, KeepAdding = false

If $(f, \pi_i \cup \{z\})$ is the maximum score, $\pi_i = \pi_i \cup \{z\}$

}

}

Our Modified K2 Algorithm

For $i = 1$ to m

{

$\pi_i = \varnothing$

KeepAdding = true

While KeepAdding and $|\pi_i| < u$

{

Using private sub-protocols, Alice and Bob jointly determine which element of $g(i, \pi_i)$, $g(i, \pi_i \cup \{z\})$ for $z \in \text{Pred}(i) - \pi_i$ yields the maximum score

If $g(i, \pi_i)$ is the maximum score, KeepAdding = false

If $(g, \pi_i \cup \{z\})$ is the maximum score, Alice and Bob learn random shares of $g(\pi_i \cup \{z\})$, and $\pi_i = \pi_i \cup \{z\}$

}

}

K2 Score Function

α-parameters:

α_{ijk}: given a set of parents π_i of node i, the number of records that are compatible with variable i taking on its kth possible value and with the jth unique instantiation of the variables in π_i

α_{ij}: $\sum_k \alpha_{ijk}$

d_i: number of possible values of variable i

score function to determine which edge to add:

$$f(i,\pi(i)) = \prod_j \frac{(d_i - 1)!}{(\alpha_{ij} + d_i - 1)!} \prod_k \alpha_{ijk}!$$

Our Solution: Approximate Score

Modified score function: approximates the same relative ordering, and lends itself well to private computation

- Apply natural log to *f* and use Stirling's approximation
- Drop constant factor and bounded term. Result is (essentially):

$$g(i,\pi(i)) = \sum_j \left(\sum_k \left(\tfrac{1}{2}(\ln \alpha_{ijk} + \alpha_{ijk} \ln \alpha_{ijk} \right) - \left(\tfrac{1}{2}(\ln l + l \ln l \right) \right) + p$$

where $l = \alpha_{ij} + d_i + 1$ and p is publicly computable by Alice and Bob

Our Solution: Components

Sub-protocols used:

- Private computation of α-parameters
- Private score computation
- Private score comparison

All intermediate values (scores and parameters) are shared using secret sharing. Privacy is with respect to an honest-but-curious adversary.

Cryptographic Tools

- **Secret sharing:** A secret x is shared between Alice and Bob if together they can recompute x, but separately they cannot.
  - Example: $x = a + b \bmod n$, where a is random and n is known to both Alice and Bob.
- **Additive homomorphic encryption:** Given encryptions $E(m_1)$ and $E(m_2)$, it is possible to compute $E(m_1 + m_2)$ without knowledge of the secret key.
  - Paillier's cryptosystem is an example.
- **Private scalar product:** Given two bit vectors z held by Alice and z' held by Bob, compute secret shares of the scalar product $z \cdot z'$ [GLLM04].
- **Secure log computations:** Given x shared between Alice and Bob, compute shares of $\ln x$ and $x \ln x$ [LP00].

Private Computation of α-Parameters

Private inputs: D_A and D_B

Common input: values $1 \leq i \leq m$, $1 \leq j \leq q_i$, $1 \leq k \leq d_i$, plus the current value of π_i and an instantiation of the variables in π_i

Output: random shares of α_{ijk}

Private Computation of α-Parameters

- Alice creates a vector representing which of her (partial) records are compatible with i,j,k:
  - For $t = 1$ to n, Alice sets $I_A[t] = 0$ if the tth record is compatible with i,j,k, and $I_A[t] = 1$ otherwise.
- Bob does the same with his data to create I_B
- Alice and Bob use the private scalar product protocol to obtain shares of the number α_{ijk} of compatible records.

Private Score Computation

Input: Alice and Bob hold random shares of all α_{ijk}

Output: Alice and Bob get random shares of $g(i, \pi(i))$

$$g(i,\pi(i)) = \sum_j \left(\sum_k \left(\tfrac{1}{2}(\ln \alpha_{ijk} + \alpha_{ijk} \ln \alpha_{ijk} \right) - \left(\tfrac{1}{2}(\ln l + l \ln l) \right) \right) + p$$

where $l = \alpha_{ij} + d_i + 1$ and p is publicly computable by Alice and Bob

Private Score Computation

Five types of quantities to compute shares of:

- $\ln \alpha_{ijk}$: use [LP00]
- $\alpha_{ijk} \ln \alpha_{ijk}$: use [LP00]
- $l = \alpha_{ij} + d_i - 1$: Alice and Bob can compute new shares locally.
- $\ln l$: use [LP00]
- $l \ln l$: use [LP00]

p, multiplication by 1/2, and additions can be computed locally.

Private Score Comparison

Goal: Determine which of at most m shared score values is maximum.

- In this case, the number of inputs is bounded by m, which is generally much smaller than n.
- Hence, Yao's two-party general secure computation can be efficiently used.

Recap: Our Modified K2 Algorithm

For $i = 1$ to m

{

$\pi_i = \varnothing$

KeepAdding = true

While KeepAdding and $|\pi_i| < u$

{

Using private sub-protocols, Alice and Bob jointly determine which element of $g(i, \pi_i)$, $g(i, \pi_i \cup \{z\})$ for $z \in \text{Pred}(i) - \pi_i$ yields the maximum score

If $g(i, \pi_i)$ is the maximum score, KeepAdding = false

If $(g, \pi_i \cup \{z\})$ is the maximum score, Alice and Bob learn random shares of $g(\pi_i \cup \{z\})$, and $\pi_i = \pi_i \cup \{z\}$

}

}

Efficiency and Privacy

- Inner loop of K2 algorithm runs O(mu) times.
- Each time, O(u) (= O(m)) scores to compute, each requiring O(md^u) α-parameters to be computed (where d is a bound on the number of possible values for any variable).
- Each α-parameter computation, including the scalar product protocol, requires O(n) computation and communication. [This is the only place that n comes into the complexity.]
- Everything else can be done within poly(m, d^u) communication and computation.
- Note that we do leak the order in which edges were added. Nothing else is leaked.

Overview

- Intro: privacy, privacy-preserving data mining
- Bayesian networks
- Privacy-preserving Bayesian network structure computation
- Using privacy-preserving data mining

Using PPDM

To actually use privacy-preserving data mining, this kind of PPDM is not sufficient. Also needed:

- Policies and enforcement for what queries should and shouldn't be allowed. (And methods/tools for helping to choose such policies and understanding the implications).
- Methods for data-preprocessing, including data cleaning, error handling, data imputation [JW06], and adherence to standards for how to represent the data.
- Integration of many PPDM solutions into a common framework to provide sufficient usability and utility to users.
- For many applications, ability to prove that policies were met, ability to selectively obtain more information in some cases, audit logs (with their own sets of policies and enforcement issues), etc.

Conclusions

- Increasing use of computers and networks has led to a proliferation of sensitive data.
- Without proper precautions, this data could be misused.
- Many technologies exist for supporting proper data handling, but much work remains, and some barriers must be overcome in order for them to be deployed.
- Cryptography is a useful component, but not the whole solution.
- Technology, policy, and education must work together.