

# BayesStore: Querying Probabilistic Information Extraction

*Daisy Zhe Wang*

University of California, Berkeley

5<sup>th</sup> January, 2010

# Outline

## PART1

- Probabilistic Data Analytics
- State-of-the-art Methodologies
- BayesStore Methodology and Techniques
- Example Applications

## PART2

- Probabilistic Information Extraction
  - CRF and Viterbi
- Implementing CRF and Viterbi in BayesStore
- Querying CRF-based Information Extraction
  - Querying Maximum-Likelihood Extraction
  - Querying Full Distribution
- Conclusion and Future Work

# Probabilistic Data Analytics

## Information Extraction Systems



**Extracted Entities** (e.g. names,

Which VLDB attendees work for Microsoft **with  $p > 0.8$** ?

## Sensor Networks



**Sensor readings** (e.g. light,

What's the **Gaussian distribution** of average pressure of the area?

## Data Integration Systems



**Integration Results** (e.g. schema

What is **top-10 probable** records of an employee called "Bond"?

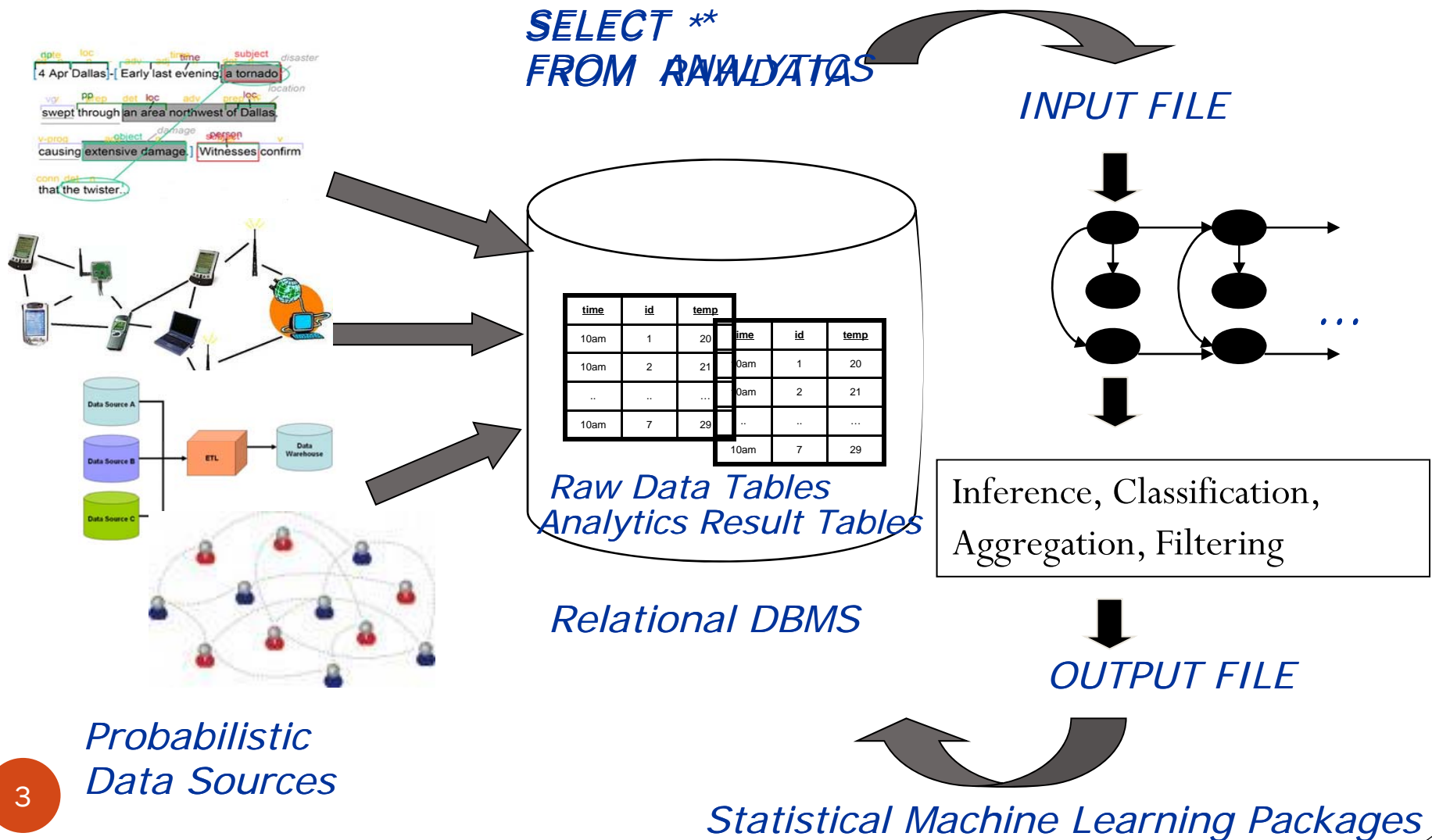
## Social Networks



**Predictive Analysis Results**

**How many** of soccer fans would be interested in the Ad for traveling to Argentina?

# State-of-the-art (I) --- RDBMS + Statistical Machine Learning Packages



# Probabilistic DBMS

	$S^p$		
	<b>A</b>	<b>B</b>	
$s_1$	‘m’	1	0.8
$s_2$	‘n’	1	0.5

	$T^p$		
	<b>C</b>	<b>D</b>	
$t_1$	1	‘p’	0.6

Figure 2: A probabilistic database  $D^p$

$pwd(D^p) =$	
world	prob.
$D_1 = \{s_1, s_2, t_1\}$	0.24
$D_2 = \{s_1, t_1\}$	0.24
$D_3 = \{s_2, t_1\}$	0.06
$D_4 = \{t_1\}$	0.06
$D_5 = \{s_1, s_2\}$	0.16
$D_6 = \{s_1\}$	0.16
$D_7 = \{s_2\}$	0.04
$D_8 = \emptyset$	0.04

(a)

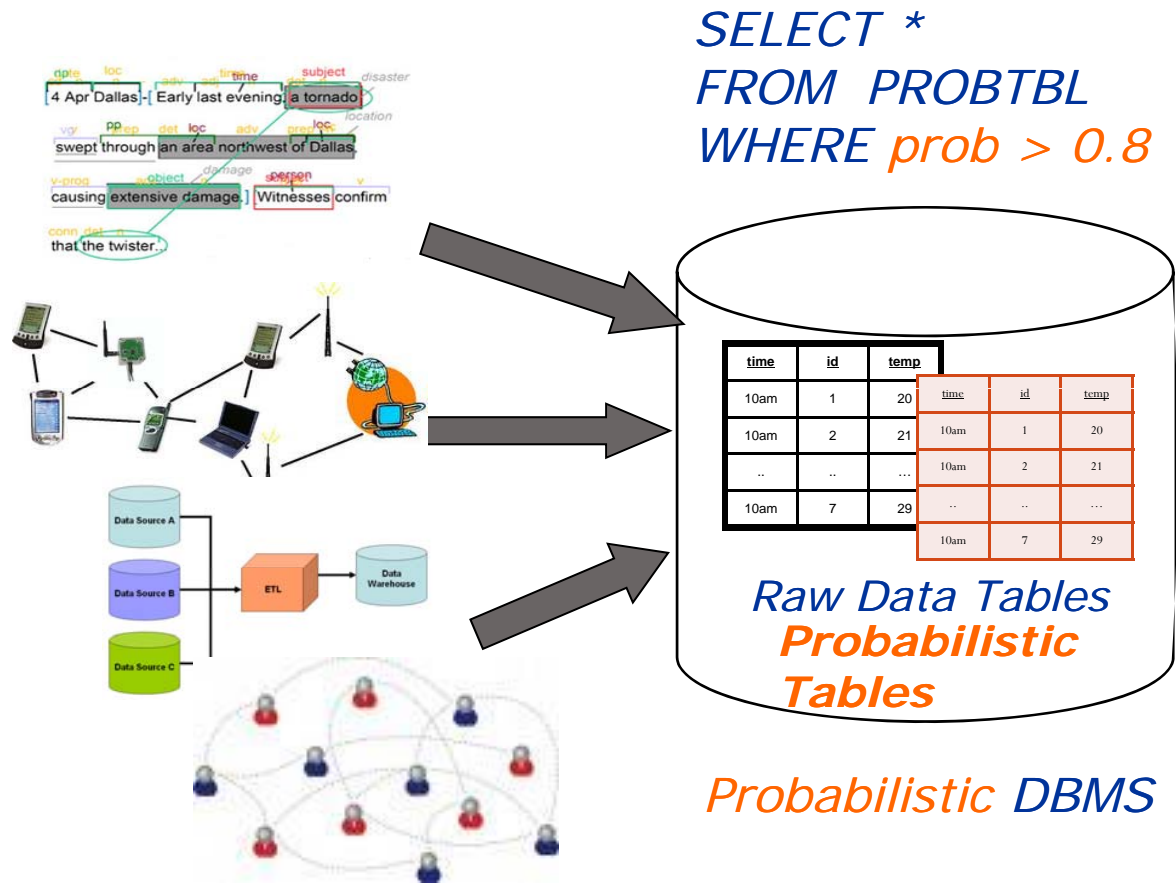
$q(u) : S^p(x, y), T^p(z, u), y = z$
(b)

$q^{pwd}(D^p) =$	
answer	prob.
{‘p’}	0.54
$\emptyset$	0.46

(c)

Figure 3: (a) The possible worlds for  $D^p$  in Figure 2, (b) a query  $q$ , and (c) its possible answers.

# State-of-the-art (II) ---- Probabilistic DBMS + Statistical Machine Learning Packages



*Probabilistic Data Sources*

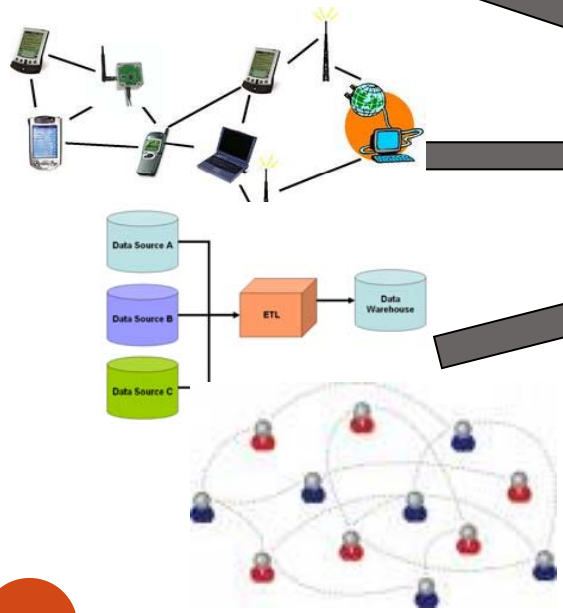
# Application Requirements for Probabilistic Data Analysis

- **Probabilistic:** Support the representation of probabilistic data with arbitrary probabilistic **correlations**
- **Integrated:** Support **ad-hoc queries** over probabilistic data, involving both relational and inference operators
- **Data Centric:** Push probabilistic data analytics tasks (e.g. model learning, inference) **to the data** in DBMS

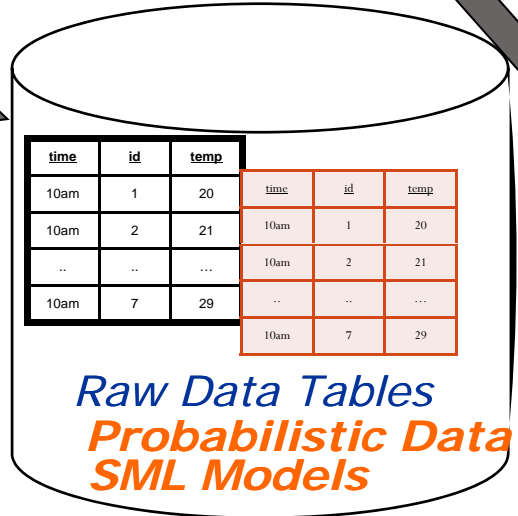
# BayesStore Methodology

## Probabilistic Data Sources

4 Apr Dallas - Early last evening, a tornado swept through an area northwest of Dallas, causing extensive damage. Witnesses confirm that the twister.



*SELECT \*  
Probabilistic Analytics  
FROM RAWDATA  
Queries (extension  
of SQL)*



*Raw Data Tables  
Probabilistic Data  
SML Models*

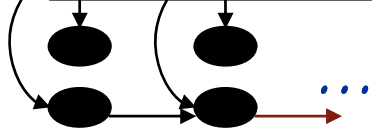
*Analytics DBMS*

Which friends of mine work for Microsoft with  $p > 0.8$ ?

What's the Gaussian distribution of average pressure of the area?

What is top-10 probable records of an employee called "Bond"?

How many of soccer fans would be interested in the Ad for traveling to Argentina?



*OUTPUT FILE*

*Statistical Machine Learning Packages*

# App1: Sensor Networks [Wang et al., VLDB08]

**Model:** First-order Bayesian Network

*Tp1: All Tp values*

*L2: All L values*



*Tp3: All Tp values  
with Sid=1*

*Tp4: All Tp values  
with Sid=2*



*Tp5: All Tp values  
with Sid != 2*



**Query**

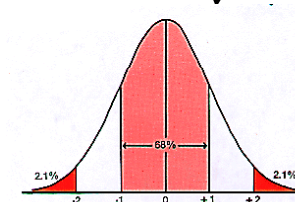
```

Select avg(TpP)
From Sensor
Where room=1
  
```

**Sensor<sup>P</sup> (Time, Room, Sid,  
Temperature(Tp)<sup>P</sup>, Light(L)<sup>P</sup>)**

Time	Room	Sid	Tp <sup>P</sup>	LP
1	1	1	Hot	
1	1	2	Cold	Dark
1	1	3		

**Result**



# App2: Information Extraction (IE)

[Wang et al., ICDE10]

## Text

For years, Microsoft Corporation CEO Bill Gates was against open source. But today he appears to have changed his mind. "We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access." .....

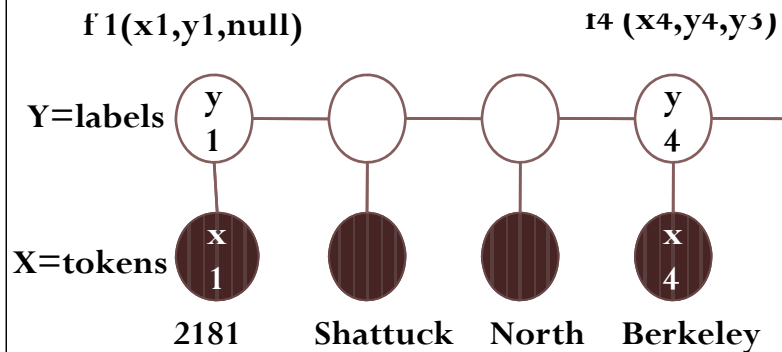
## Query

```
Select * From People
Where Organization = 'Microsoft'
and prob(*)>0.8?
```

## People<sup>P</sup>

<u>Name<sup>P</sup></u>	<u>Title<sup>P</sup></u>	<u>Organization<sup>P</sup></u>

## Model: Conditional Random Fields (CRF)



## Result

Bill Gates  
Bill Veghte  
...

(from Cohen's IE tutorial, 2003)

# App3: Social Networks [Hellerstein et al., VLDB09]

- **Data**: click streams, ad server logs, CRM records, profile database
- **Task**: “hypertargeting”: uses the information in member’s profile to serve up ads they might be interested
- **Models** for “hypertargeting” (e.g. SVN, BN, NLP models)
  - Model 1: who is soccer fan
  - Model 2: whether two people are “similar”
- **Probabilistic Data**: results of the models (e.g. classification, predictions)
- **Queries** over probabilistic data/model
  - Query 1: How many female soccer fans under the age of 30 visited the Toyota community over the last four days and saw a web ad.?
  - Query 2: How are these people similar to those that visited Nissan?
  - Models are applied and **combined** in **ad-hoc exploratory queries** over **changing subsets of data** (e.g. Toyota/Nissan community)

# BayesStore Core Techniques

- Represent **probabilistic data** and **probabilistic model** with arbitrary correlation efficiently in DBMS
  - Incomplete relations and probabilistic attributes
  - New data types, such as vectors and matrices
- Implement inference operations **natively** in DBMS
  - Operators as UDFs(User Defined Functions) over new data types
  - **Efficient** declarative implementation of inference algorithms
- Co-optimize relational and inference operators
  - Improve **query efficiency**
- Support principled probabilistic querying framework
  - Improve **answer quality**

# Outline

## PART1

- Probabilistic Data Analytics
- State-of-the-art Methodologies
- BayesStore Methodology and Techniques
- Example Applications

## PART2

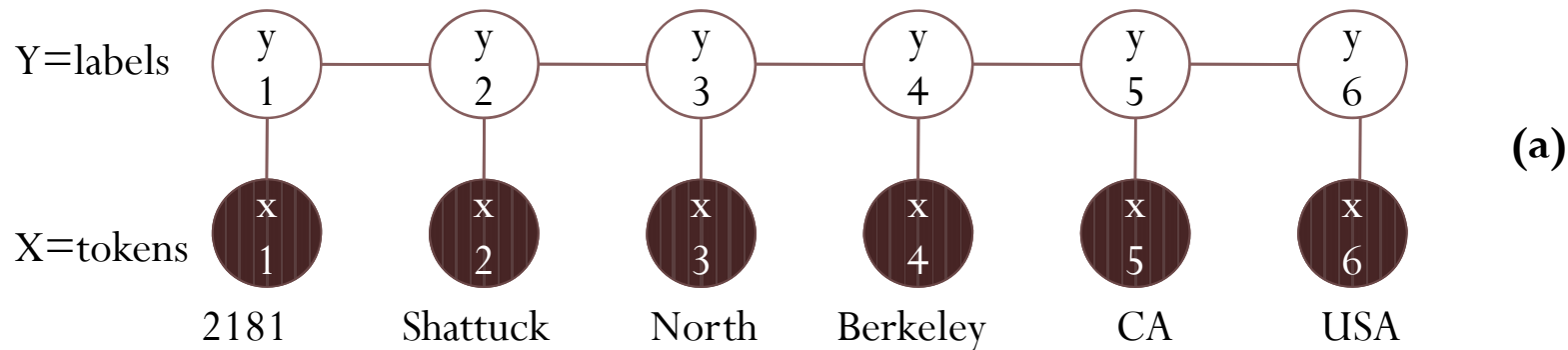
- Probabilistic Information Extraction
  - CRF and Viterbi
- Implementing CRF and Viterbi in BayesStore
- Querying CRF-based Information Extraction
  - Querying Maximum-Likelihood Extraction
  - Querying Full Distribution
- Conclusion and Future Work

# Probabilistic Information Extraction (IE)

- IE Tasks
  - Named Entity/Record Extraction
  - Document Classification (e.g. Email Signature Block Extraction)
  - Sentiment Analysis
- Rule-based IE (e.g. DBLife)
- **Probabilistic IE**: better accuracy, flexibility and adaptability
  - Naïve Bayes (NB)
  - Hidden Markov Model (HMM)
  - Conditional Random Fields (CRF)

# Conditional Random Fields (CRF)

CRF Model:



Features (i.e. correlations):

$$f_1(y_i, y_{i-1}, x_i) = [x_i \text{ appears in a city list}] \cdot [y_i = \text{city}]$$

$$f_2(y_i, y_{i-1}, x_i) = [x_i \text{ is an integer}] \cdot [y_i = \text{apt.num}]$$

$$\cdot [y_{i-1} = \text{streetname}]$$

(b)

Probabilistic Distribution:

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left\{ \sum_{i=1}^T \sum_{k=1}^K \lambda_k f_k(y_i, y_{i-1}, x_i) \right\},$$

(c)

Possible Extraction Worlds:

	x	2181	Shattuck	North	Berkeley	CA	USA	
y1	apt. num	street name	city	city	state	country		{0.6}
y2	apt. num	street name	street name	city	state	country		{0.1}

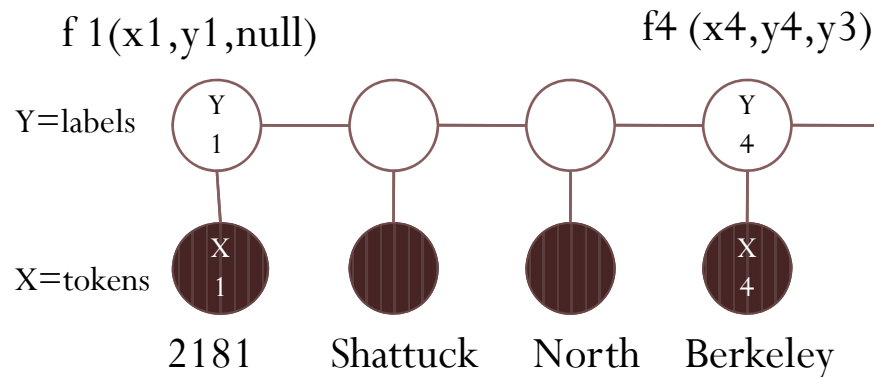
(d)

# Viterbi Top-k Inference on CRF

Viterbi Dynamic Programming Algorithm:

$$V(i, y) = \begin{cases} \max_{y'} (V(i-1, y') + \sum_{k=1}^K \lambda_k f_k(y, y', x_i)), & \text{if } i \geq 0 \\ 0, & \text{if } i = -1. \end{cases} \quad (3)$$

CRF Model:



Dynamic Programming V matrix:

pos	street num	street name	city	state	country
0	5	1	0	1	1
1	2	15	7	8	7
2	12	24	21	18	17
3	21	32	32	30	26
4	29	40	38	42	35
5	39	47	46	46	50

# Outline

## PART1

- Probabilistic Data Analytics
- State-of-the-art Methodologies
- BayesStore Methodology and Techniques
- Example Applications

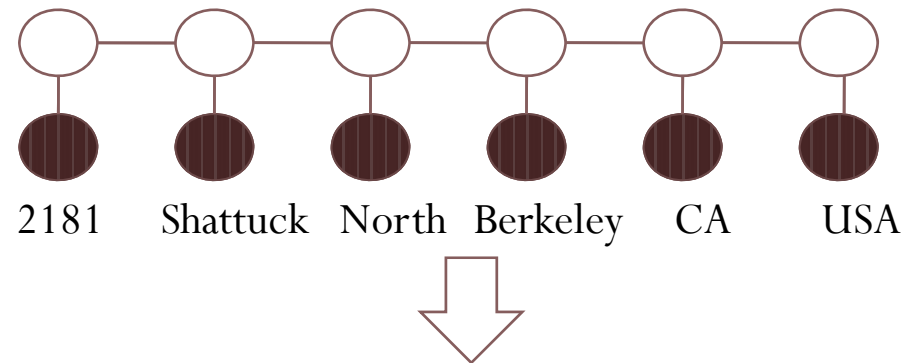
## PART2

- Probabilistic Information Extraction
  - CRF and Viterbi
- **Implementing CRF and Viterbi in BayesStore**
- Querying CRF-based Information Extraction
  - Querying Maximum-Likelihood Extraction
  - Querying Full Distribution
- Conclusion and Future Work

# Text Data and CRF Representations

- **Text Data – Token Table:** Inverted file, one token per row
- **CRF – Factor Table:** one  $\langle \text{token}, \text{prevLabel}, \text{label} \rangle$  triple per row

For years, **Microsoft Corporation** CEO **Bill Gates** was against open source. But today he appears to have changed his mind. "We can be open source. We love the concept of shared source," said **Bill Veghte**, a **Microsoft VP**. "That's a super-important shift for us in terms of code access." .....



docID	pos	token	Label
1	0	2181	
1	1	Shattuck	
1	2	North	
1	3	Berkeley	
1	4	CA	
1	5	USA	

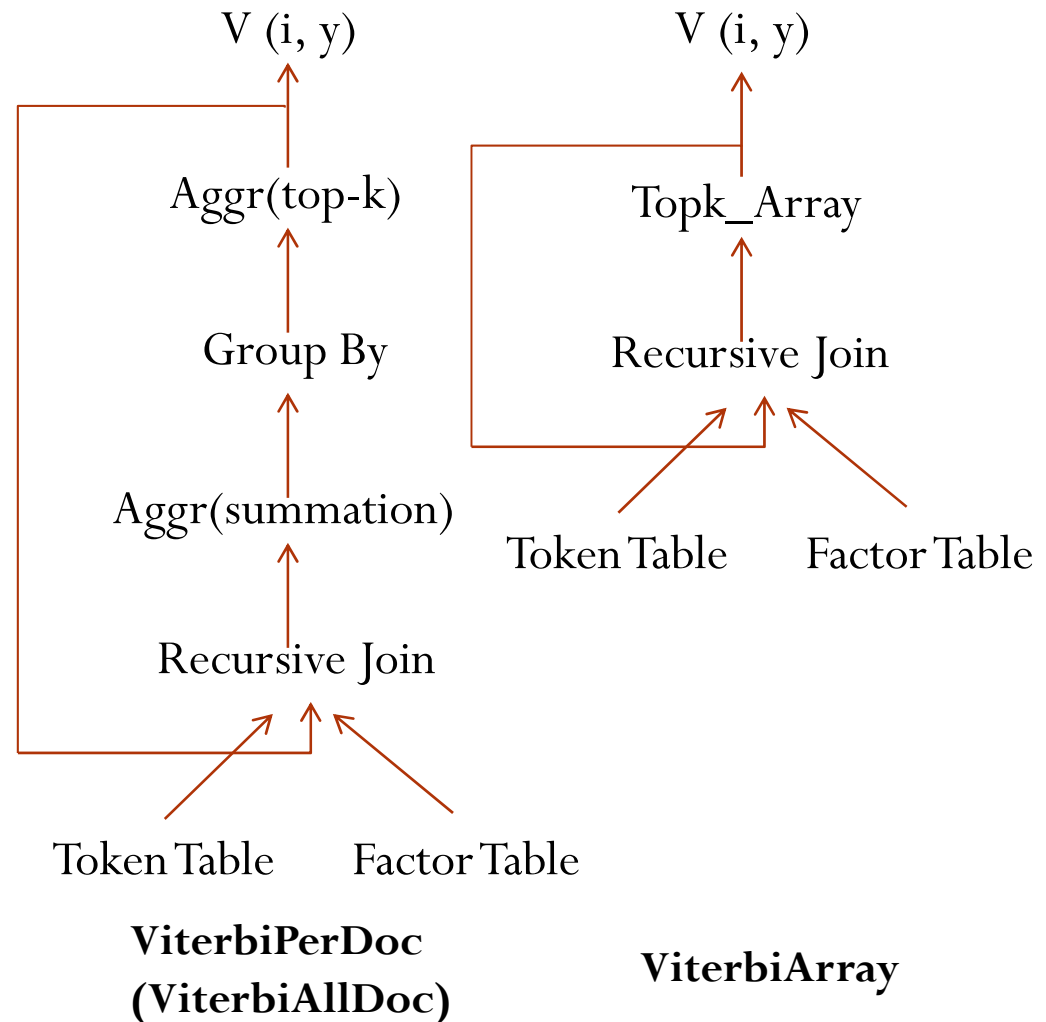
Token Table

token	prevLabel	label	score
2181(DIGIT)	null	street num	22
2181(DIGIT)	null	street name	5
...	..	..	
Berkeley	streetname	street name	10
Berkeley	streetname	city	25
..	..	..	

Factor Table

# Viterbi Implemented in SQL

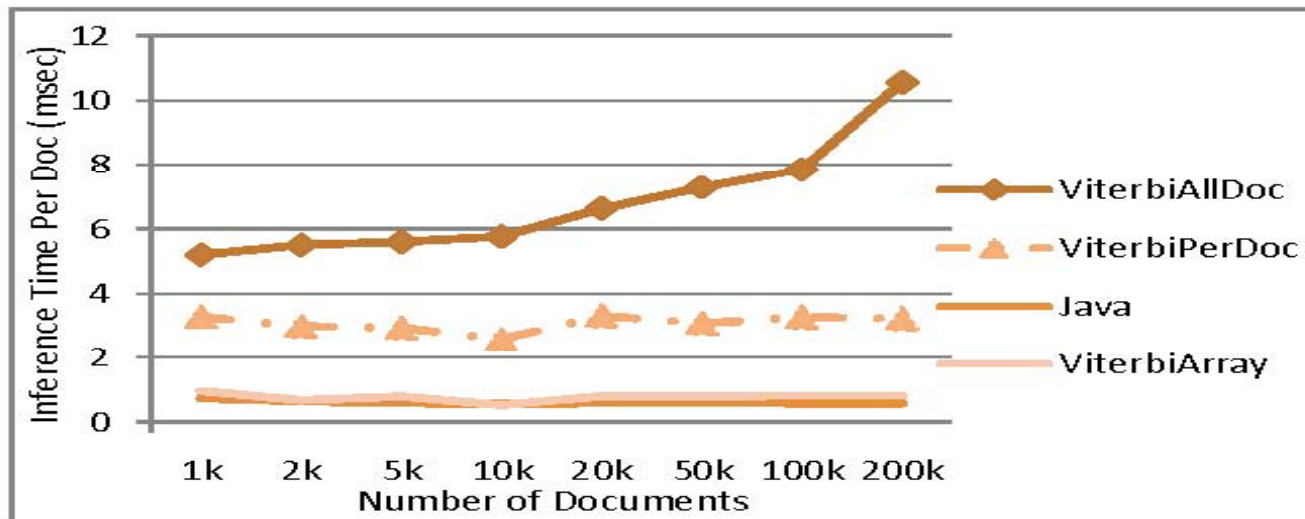
- **ViterbiPerDoc (ViterbiAllDoc)**
  - Implemented as a UDF
  - WITH RECURSIVE : compute  $V(i,y)$  from  $V(i-1,y)$
  - Problem: 5 times slower than hand-tuned CRF open source library
- **ViterbiArray**
  - Represent factors  $f(x,y,y')$  as an array
  - UDFs over array types for aggregation
  - Improve the main memory performance, Join efficiency, representation compactness
  - Results: Equal or better performance compare to open source CRF library



# Evaluation Setup

- **Implementation**
  - PostgreSQL8.4.1
  - 2.4 GHz Intel Pentium 4 Linux system with 1GB RAM
- **Dataset**
  - Address Strings Dataset from YellowPages
  - Bibliography Strings Dataset
  - Email Signature Blocks from Enron Dataset
- **Model**
  - CRF model learned using CRF Java open source package using feature set in the machine learning literature
- **Evaluation 1:**
  - Runtime Efficiency comparison between different SQL implementations of the Viterbi Algorithm, and the Java implementation in CRF open source package

# Evaluation1: [Runtime Efficiency] SQL Implementations of Viterbi



**Figure 6: Average inference time (msec) for a single document for different implementations of the Viterbi algorithm.**

dataset	ViterbiAllDoc	ViterbiPerDoc	ViterbiArray	Java
address	10.5 msec	3.2 msec	0.8 msec	0.5 msec
bib	1760.1 msec	175.1 msec	6.2 msec	16.2 msec

**Figure 7: Average inference time per document (msec) for different Viterbi implementations on address and bib dataset.**

# Outline

## PART1

- Probabilistic Data Analytics
- State-of-the-art Methodologies
- BayesStore Methodology and Techniques
- Example Applications

## PART2

- Probabilistic Information Extraction
  - CRF and Viterbi
- Implementing CRF and Viterbi in BayesStore
- Querying CRF-based Information Extraction
  - Querying Maximum-Likelihood Extraction
  - Querying Full Distribution
- Conclusion and Future Work

# Entity Tables and Two Query Families

**Address  
(EntityTbl1)**

strID	apt. num <sup>P</sup>	street num <sup>P</sup>	street name <sup>P</sup>	city <sup>P</sup>	state <sup>P</sup>	country <sup>P</sup>
1	null	2181	Shattuck	North Berkeley	CA	USA
2	12B	331	Fillmore St.	Seattle	WA	USA
3	224B	null	Ford South St.	Louis	MO	USA

(a)

**Company  
(EntityTbl2)**

strID	company name <sup>P</sup>	city <sup>P</sup>	state <sup>P</sup>
1	Google	Mountain View	CA
2	Yahoo!	Santa Clara	CA
3	Microsoft	Seattle	WA

(b)

**Query Family 1: (SPJ-over-ML) SPJ over  
Maximum-likelihood (ML) Queries**

```
CREATE VIEW entityTbl1-ML as
SELECT *, rank() OVER (ORDER BY prob(*) DESC) r
FROM entityTbl1
WHERE r = 1;
```

**Query Family 2: (Probabilistic SPJ) Top-k  
over Probabilistic SPJ Queries**

```
SELECT *, rank() OVER (ORDER BY prob(*) DESC) r
FROM SQLQuery
WHERE r <= k [ AND prob(*) > threshold ]
```

# Optimized Select-over-ML Query

## Example Query:

```
SELECT *  
FROM Address-ML  
WHERE city like '%Sacramento%'
```

- (1) Test if the text-string  $d$  contains “Sacramento”
- (2) Test if “Sacramento” is assigned label “city” in the string

## Optimizations:

- Condition (1) can be pushed down to Token Table with inverted index
- Condition (2) can be pushed into the Viterbi algorithm
  - Technique: early-stop dynamic programming
  - Stopping condition: if none of the top-1 partial segmentation satisfy condition (2)

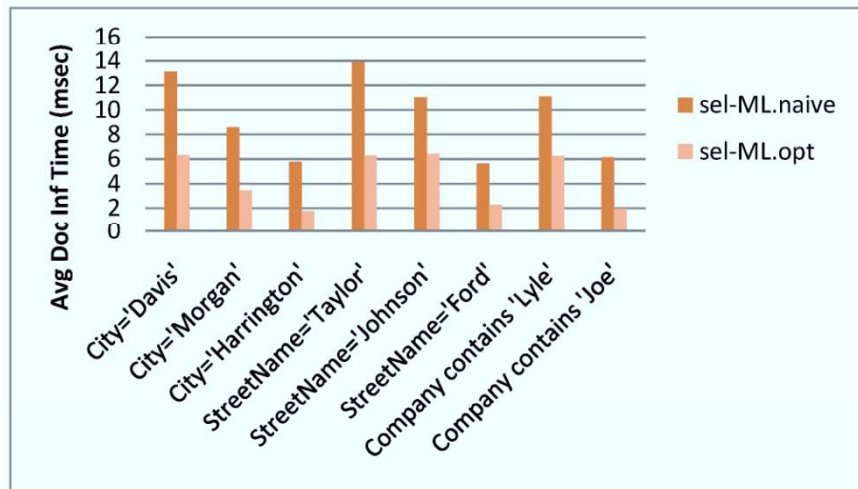
Sacramento  
Avenue  
San  
Francisco  
CA

pos	street num	street name	city	state	country
0	5	1	0	1	1
1	2	15	7	8	7

**STOP!**

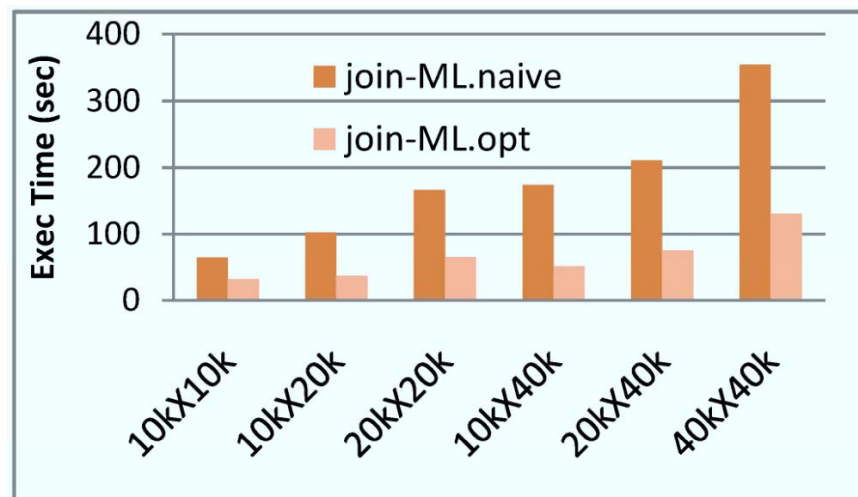
# Evaluation 2: [Runtime Efficiency]

## Optimized SPJ-over-ML Queries



### Select-over-ML: naïve vs. opt

**Figure 1:** Performance comparison between **sel-ML.naive** and **sel-ML.opt** with **difference selection conditions**.



### Join-over-ML: naïve vs. opt

**Figure 2:** Performance comparison between **join-ML.naive** and **join-ML.opt** with **different input sizes**.

# SPJ-over-ML vs. Probabilistic SPJ

## Example Email Signature Block:

Michelle L. Simpkins

Winstead Sechrest & Minick P.C.

100 Congress Avenue, Suite 800...

## Query 1. Find contacts with companyname containing "Winstead"

```
SELECT *  
FROM Contacts  
WHERE companyname LIKE '%Winstead%'
```

## Query 2. Find all contact pairs with the same companyname

```
SELECT *  
FROM Contacts C1, Contacts C2  
WHERE C1.companyname = C2.companyname
```

- **Problem with SPJ-over-ML:** even high-quality ML extractions contain errors
  - Example: ML extraction assigns NULL to the companyname attribute – Error!
  - Query 1 (select-over-ML) generates an empty result – **missing results!**
  - Query2 (selfjoin-over-ML) again generates an empty result –**she does not even match herself!**
- **Probabilistic SPJ:** SPJ queries are computed over the set of possible “worlds” (PWs) induced from the CRF
  - Improved Answer Quality
  - but Increased Computation Cost
- **A Design Space:** performance vs. answer quality

# Incremental Viterbi Algorithm

- Probabilistic SPJ queries needs efficient sorted access to the possible extractions of a document by descending probabilities
- Conventional Top-k Viterbi algorithm: each cell in  $V$  matrix maintain a list of top-k items, where  $k$  is known a priori
- **Novel Incremental Top-k Viterbi algorithm:** computes the NEXT highest-probability extraction incrementally and efficiently

## Intuition:

- (1) new items needs to be computed in  $V$  matrix to compute the NEXT
- (2) those new items needed only lies on the last extraction path
- (3) after new items are computed, use the same Viterbi formula to compute the NEXT

pos	street num	street name	city	state	country
0	5	1	0	1	1
1	2	15	7	8	7
2	12	24	21	18	17
3	21	32	32	30	26
4	29	40	38	42	35
5	39	47	46	46	50

# Probabilistic Join

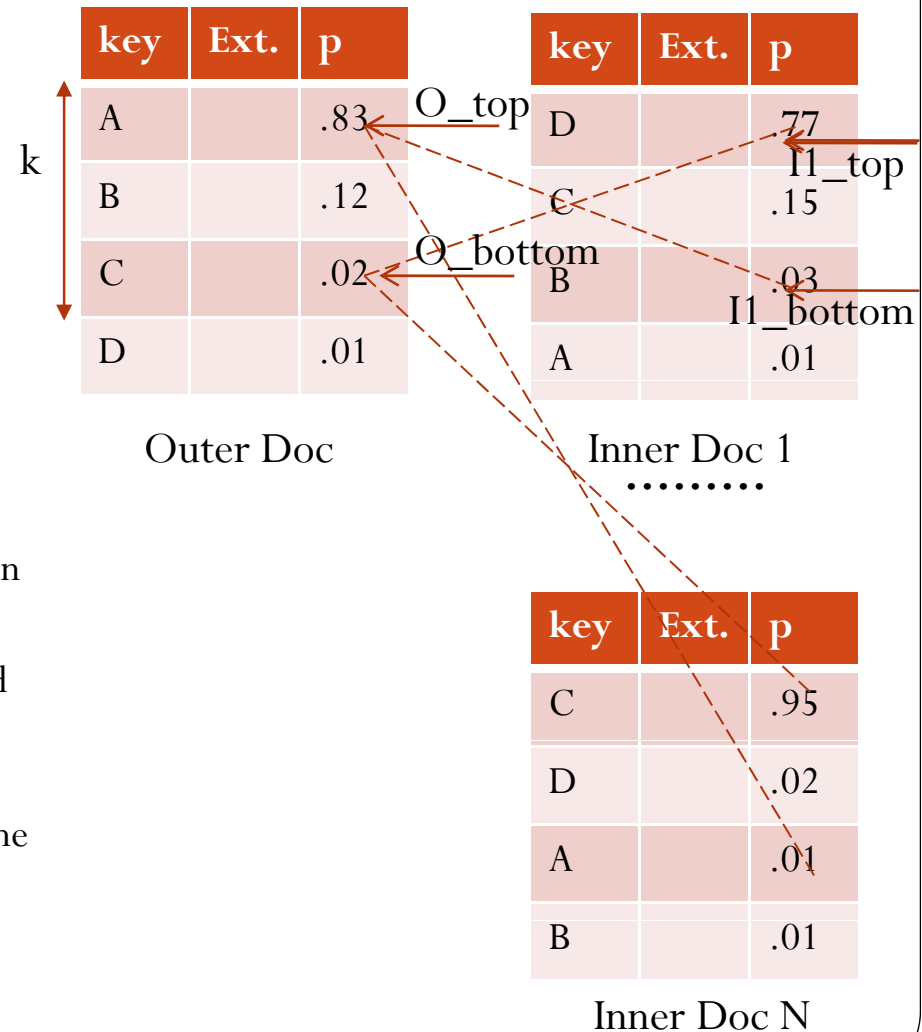
## Example Query:

```
SELECT *, rank() OVER (ORDER BY prob(*) DESC) r
FROM ( SELECT *
      FROM Address A, Company C
      WHERE A.city = C.city) as AddrComp
WHERE r = 1
```

- Naïve Probabilistic Join algorithm: first compute top-k extraction views for both input document corpus, then compute join
- Problem: k is varies for different documents : majority  $k=1$ , others  $k \gg 1$
- Solution: Incremental Join algorithm based on Rank-join

### Intuition:

- (1) Each document is a ranked list of extractions computed by **incremental Viterbi** with descending probability
- (2) Rank-join is applied to **each pair** of joining document
- (3) A set of rank-joins are computed **simultaneously** for one outer document and a set of inner documents



# Probabilistic Selection

## Example Query:

```
SELECT *, rank() OVER (ORDER BY prob(*) DESC) r
FROM ( SELECT * FROM Address
      WHERE streetname like '%Davis%') as Address
WHERE r = 1 AND prob(*) > threshold
```

Compute the top-1 extraction that satisfy the condition (e.g. “Davis” is labeled streetname) with probability higher than the threshold

## Technique:

Generalized Constrained Viterbi algorithm

pos	street num	street name	city	state	country
0	5	1	0	1	1
1	7	15	7	8	7
2	XXX	24	XXX	XXX	XXX
3	XXX	32	XXX	XXX	XXX
4	29	40	38	41	33
5	36	47	46	46	49

# Probabilistic Projection

## Example Query:

```
SELECT *, rank() OVER (ORDER BY prob(*) DESC) r
FROM ( SELECT city
      FROM Address ) as Address
WHERE r = 1
```

## Intuition:

- (1) In extractions, each token is either given labels projected on (e.g. city) or “don’t care” (aggregation of projected out labels)
- (2) An additional U matrix is needed to compute the top-k aggregated paths
- (3) The aggregation is a multi-way rank-join

pos	street num	street name	city	state	country
0	5	1	0	1	1
1	10	18	10	11	10
2	18	30	27	24	23
3	30	41	43	39	35
4	43	54	52	56	49
5	54	62	61	61	57

pos	street num	street name	city	state	country
1	10	18	10	11	10
2	18	30	27	24	23
3	30	41	43	39	35
4	41	52	50	54	47
5	54	62	61	61	57
6	U (6,-1) = 73				

Equation(3)

Equation(4)

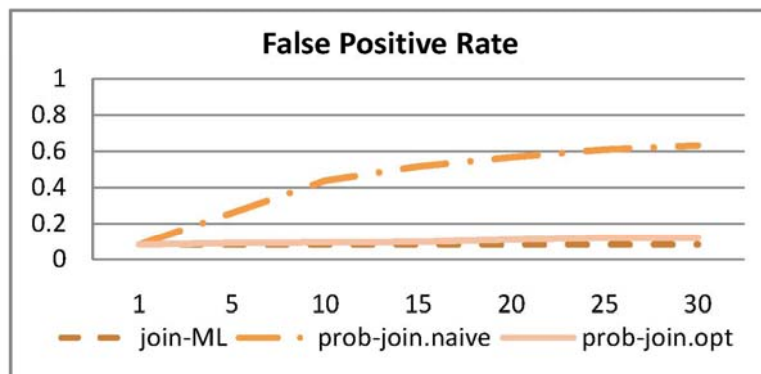
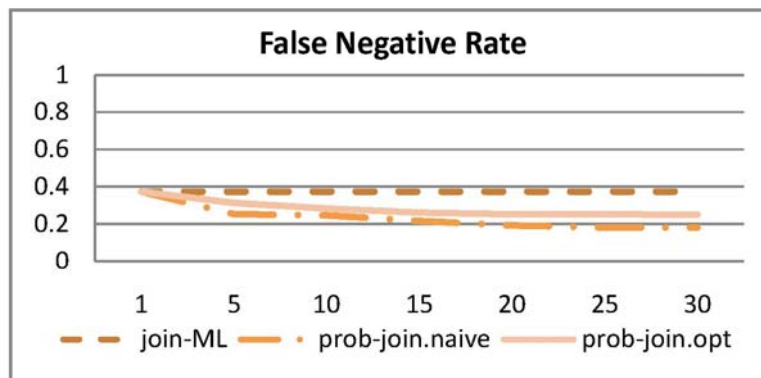
# Evaluation 3: [Answer Quality]

## Probabilistic SPJ Queries

(False -)	company	firstname	lastname	jobtitle	department
sel-ML	0.074	0.012	0.036	0.102	0.286
prob-sel	0.014	0	0.006	0.037	0.079
(False +)	company	firstname	lastname	jobtitle	department
sel-ML	0.010	0	0	0.010	0
prob-sel	0.009	0.006	0.006	0.010	0

### Probabilistic Select vs. Select-over-ML

Figure 1: False negative, false positive rates comparison between **sel-ML** and **prob-sel** queries with different selection conditions.

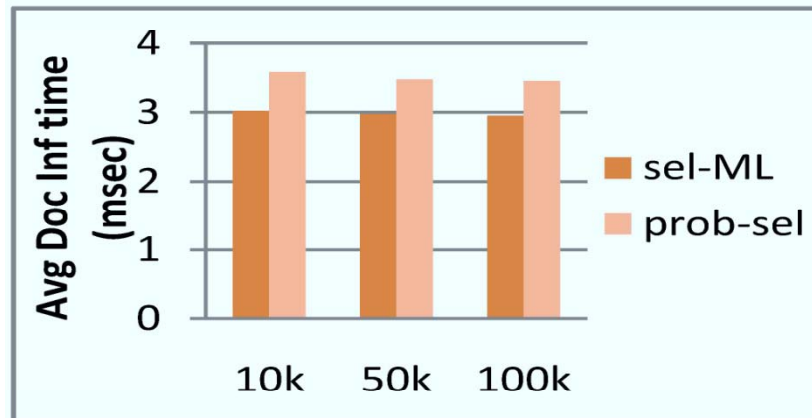


### Probabilistic Join vs. Join-over-ML

Figure 2,3: False negative, false positive rates comparison between **join-ML** and **prob-join** queries with different k values.

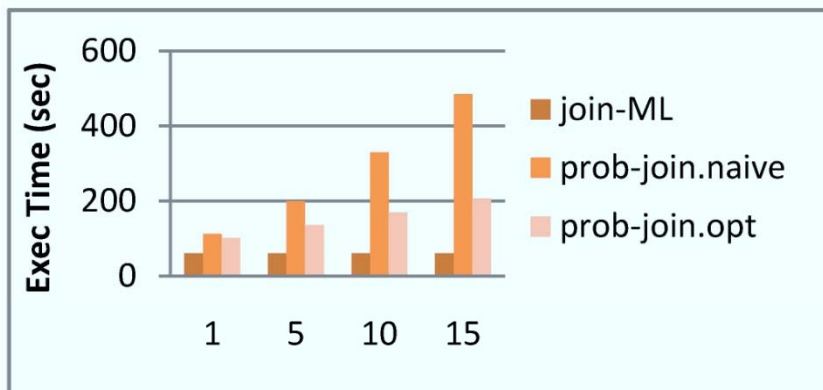
# Evaluation 4: [Runtime Efficiency]

## Probabilistic SPJ Queries



### Probabilistic Select vs. Select-over-ML

**Figure 1:** Performance comparison between **sel-ML** and **prob-sel** queries with **different input sizes**.



### Probabilistic Join vs. Join-over-ML

**Figure 2:** Performance comparison between **join-ML** and **prob-join** queries with **different k values**.

# Outline

## PART1

- Probabilistic Data Analytics
- State-of-the-art Methodologies
- BayesStore Methodology and Techniques
- Example Applications

## PART2

- Probabilistic Information Extraction
  - CRF and Viterbi
- Implementing CRF and Viterbi in BayesStore
- Querying CRF-based Information Extraction
  - Querying Maximum-Likelihood Extraction
  - Querying Full Distribution
- Conclusion and Future Work

# Conclusion & Future Work

- Extensibility and Abstractions
  - Matrix and Vector data type and operations
  - Statistical Machine Learning models
  - Extended SQL language for probabilistic data analytics
- Parallelization
- Applications
  - Text Analytics
  - Log Analysis
  - Predictive Analysis
  - Collaborative Filtering
  - Clinical Database and Bio-informatics

Thank you! ... Questions? 😊

---

## 2. Inference Algorithms (I): Sum/Max-Product Algorithm in SQL

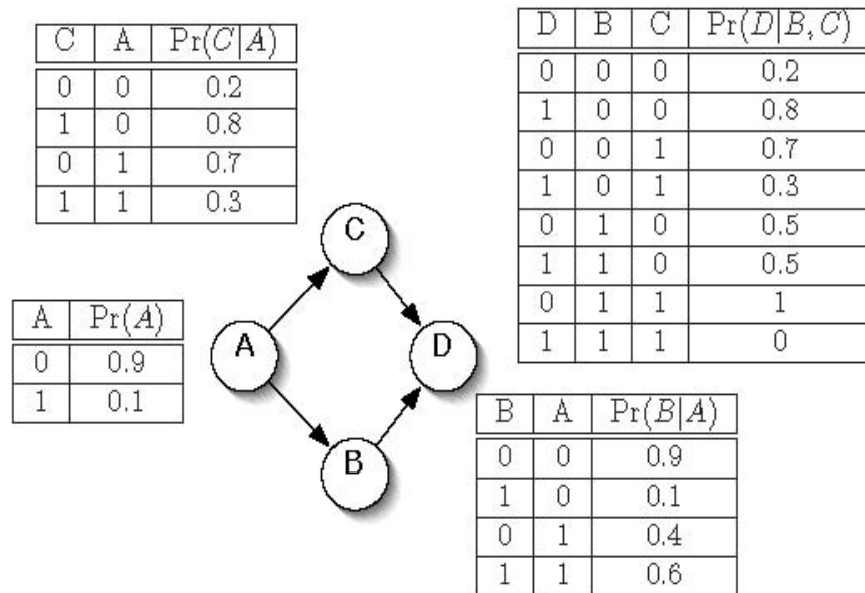


Figure 2: A simple Bayesian Network

Product is join between Factor Tables

Joint Distribution:

$$Pr(A,B,C,D) = Pr(A)Pr(B | A)Pr(C | A)Pr(D | B,C)$$

SQL Query for Joint Distribution:

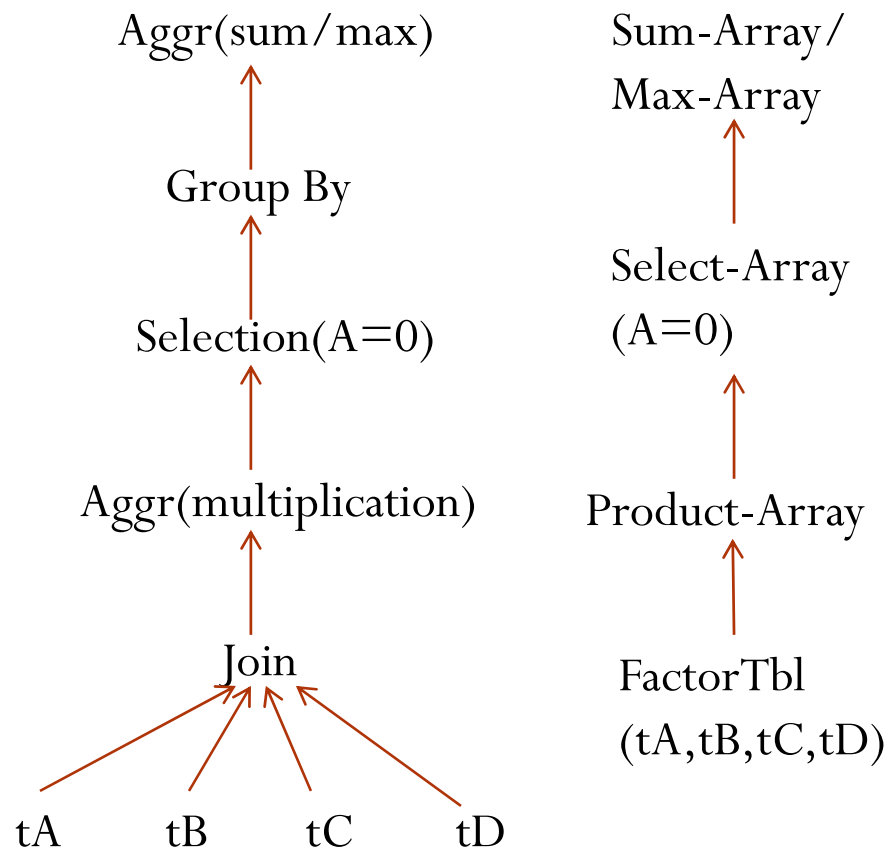
Create view joint as (

select A,B,C,D,(tA.p\*tB.p\*tC.p\*tD.p) as p

from tA, tB, tC, tD

where tA.A=tB.A and tA.A=tC.A ... )

## 2. Inference Algorithms (II): Sum/Max-Product Algorithm in SQL



**Sum/Max is group-by and aggregation**

Conditional Distribution:

$\Pr(C \mid A=0) =$

$$\sum_{\{B,D\}} \Pr(A)\Pr(B \mid A)\Pr(C \mid A)\Pr(D \mid B,C)$$

SQL Query for Conditional Distribution:

```

select C, sum(p)
from joint
where A=0
group by C
  
```

# Previous Attempts Storing IE Models in Probabilistic DB

- Gupta and Sarawagi (VLDB06) :
  - Statistical (CRF) Model to perform extraction
  - Extraction results store in ProbDB approximately
  - Optimize for accuracy and space trade-off

Figure 3: Four segmentations of the address string '52-A Goregaon West Mumbai 400 076' along with their probabilities.

Id	House_no	Area	City	Pincode	Prob
1	52	Goregaon West	Mumbai	400 062	0.1
1	52-A	Goregaon	West Mumbai	400 062	0.2
1	52-A	Goregaon West	Mumbai	400 062	0.5
1	52	Goregaon	West Mumbai	400 062	0.2

Figure 4: Segmentation-per-row model for the example in Figure 3.

Id	House_no	Area	City	Pincode
1	52 (0.3) 52-A (0.7)	Goregaon West (0.6) Goregaon (0.4)	Mumbai (0.6) West Mumbai (0.4)	400 062 (1.0)

Figure 5: One-row model for the example in Figure 3.