PANDA
A System for Provenance and Data

Robert Ikeda
Jennifer Widom
Stanford University
Example

Pipeline for sales predictions
Example

CustList_1
CustList_2
...
CustList_{n-1}
CustList_n

Dedup

Europe

Union

Predict

ItemAgg

ItemVolumes

ClothCo Items

Buying Patterns

Robert Ikeda
Example

CustList_1
CustList_2
...
CustList_{n-1}
CustList_n

Dedup

Europe

USA

Union

Predict

ItemAgg

ItemVolumes

ClothCo Items

Buying Patterns

<table>
<thead>
<tr>
<th>Item</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cowboy Hat</td>
<td>3</td>
</tr>
</tbody>
</table>

Robert Ikeda
Example

Dedup → Union → Predict → ItemAgg → ItemVolumes

CustList_1 → CustList_2 → ... → CustList_n

Europe → USA

ClothCo Items → Buying Patterns

<table>
<thead>
<tr>
<th>Item</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cowboy Hat</td>
<td>3</td>
</tr>
<tr>
<td>Cowboy Hat</td>
<td></td>
</tr>
<tr>
<td>Cowboy Hat</td>
<td></td>
</tr>
</tbody>
</table>

Robert Ikeda
Example

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amelie</td>
<td>...Paris, TX</td>
</tr>
<tr>
<td>Jacques</td>
<td>...Paris, TX</td>
</tr>
<tr>
<td>Isabelle</td>
<td>...Paris, TX</td>
</tr>
</tbody>
</table>
Example

- CustList_1
- CustList_2
- CustList_{n-1}
- CustList_n

Dedup

Europe

Union

Predict

ItemAgg

ItemVolumes

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amelie</td>
<td>65, quai d'Orsay, Paris, France</td>
</tr>
<tr>
<td>Jacques</td>
<td>39, rue de Bretagne, Paris, France</td>
</tr>
<tr>
<td>Isabelle</td>
<td>20 Rue D'orsel, Paris, France</td>
</tr>
</tbody>
</table>
Example

- CustList_1
- CustList_2
- CustList_{n-1}
- CustList_n

- Europe
- USA

- Dedup
- Union
- Predict

- ItemVolumes
- ClothCo Items
- Buying Patterns

<table>
<thead>
<tr>
<th>Item</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beret</td>
<td>3</td>
</tr>
</tbody>
</table>

Robert Ikeda
Past work tends to be...

**Panda**

1. Either data-based or process-based
   - Capture both — “data-oriented workflows”
2. Focused on modeling and capturing provenance
   - Also provenance operators and queries
3. Specific application domains
   - General-purpose
Remainder of Talk

- Processing nodes and provenance capture
- Provenance operations
- Provenance queries
- System and other issues
- Current research
Processing Nodes

- Relational nodes: structured, well-understood operations
- Opaque nodes
Provenance Capture

• **Model**
  – Likely to be similar to Open Provenance Model
  – Support provenance at a variety of granularities

• **Interface**
  – Allow processing nodes to create and manipulate provenance
  – For relational operations, can plug in existing provenance work
Provenance Operations

• Basic operations
  – Backward tracing
    ▪ Where did the cowboy-hat record come from?
  – Forward tracing
    ▪ Which predictions did this customer contribute to?
Provenance Operations

• Examples of additional functionality
  – Forward propagation
    ▪ Update all affected predictions after customers have moved from France to Texas
• **Examples of additional functionality**
  
  – Refresh ≈ Backward tracing + forward propagation
    
    ▪ Get latest predicted volume for cowboy hat sales (only) using latest customer lists and buying patterns
**Provenance Queries**

- **Examples**
  - How many people from each country contributed to the cowboy hat prediction?
  - Which customer list contributed the most to the top 100 predicted items?
Provenance Queries

• Examples
  – How many people from each country contributed to the cowboy hat prediction?
  – Which customer list contributed the most to the top 100 predicted items?

• Seamlessly combine provenance and data
• Compact and intuitive language
• Amenable to optimization
System and Other Issues

• Query-driven provenance capture
• Eager vs. lazy computation and storage
• Fine-grained vs. coarse-grained
• Approximate provenance
Current Research

• Building up basic system infrastructure

• Refresh
  – Efficiently compute the up-to-date value of selected output elements

• Theoretical challenges
  – Optimizing provenance storage vs. recomputation
System Infrastructure

- Handles structured relational operations as well as arbitrary Python processing nodes
- Arbitrary acyclic transformation graphs
- Backward tracing and forward propagation
Refresh

• **Problem**
  – Efficiently compute the up-to-date value of selected output elements

• **Challenges**
  – Formally defining the refresh problem
  – Understanding when refresh can be done efficiently
  – Supporting a wide class of transformations and workflows
Future Work

- Most everything in this talk 😊
Thank You

Parag Agrawal, Abhijeet Mohapatra, Raghotham Murthy, Aditya Parameswaran, Hyunjung Park, Alkis Polyzotis, Semih Salihoglu
Running Example

CustList_1 → Dedup → Europe
CustList_2 → Dedup → USA
CustList_{n-1} → Dedup → Europe
CustList_n → Dedup → USA

Union → Predict → ItemAgg

ClothCo Items → Buying Patterns

O
PANDA
A System for Provenance and Data

Robert Ikeda
Jennifer Widom
Stanford University
Panda’s Niche

1. Data-based or process-based
2. Modeling and capturing provenance
3. Specific application domains

1. Merge data-based and process-based
2. Provenance operators and queries
3. General-purpose
Overview of Past Work

1. Data-based or process-based
2. Modeling and capturing provenance
3. Specific application domains
Running Example
Running Example

Pipeline for Sales Prediction
Provenance Capture

• **Processing Nodes**
  – Relational operations
  – Opaque processing

• **Requirements**
  – Interface
  – Model
Running Example

CustList_1
CustList_2
... 
CustList_{n-1}
CustList_n

Dedup
Europe
USA

Union
Predict
ClothCo Items
Buying Patterns

ItemAgg
ItemVolumes

Paris, Texas?
Processing Nodes

• Relational Operations
  – Relational operations
  – Opaque processing

• Opaque Processing
  – Interface
  – Model
Provenance Queries

- Operate over provenance and data
- Compact and intuitive
- Amenable to efficient planning

Considering only customers from a specific list, which items are in the highest demand?
Provenance Queries

• Seamlessly combine provenance and data
• Compact and intuitive language
• Amenable to optimization
Provenance Query Examples

- How many people from each country contributed to the cowboy hat prediction?
- Which customer list contributed the most to the top 100 predicted items?
Running Example

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
<th>Address</th>
<th>Item</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amelie</td>
<td></td>
<td>65, quai d'Orsay, Paris</td>
<td>Hat</td>
<td>3</td>
</tr>
<tr>
<td>Jacques</td>
<td></td>
<td>39, rue de Bretagne, Paris</td>
<td>Hat</td>
<td></td>
</tr>
<tr>
<td>Isabelle</td>
<td></td>
<td>20 Rue D'orsel, Paris</td>
<td>Hat</td>
<td></td>
</tr>
</tbody>
</table>
```

Dedup

Europe

USA

Union

Predict

ItemAgg

ItemVolumes
Running Example

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amelie</td>
<td>65, quai d'Orsay, Paris</td>
</tr>
<tr>
<td>Jacques</td>
<td>39, rue de Bretagne, Paris</td>
</tr>
<tr>
<td>Isabelle</td>
<td>20 Rue D'orsel, Paris</td>
</tr>
</tbody>
</table>
Running Example

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amelie</td>
<td>65, quai d'Orsay, Paris</td>
</tr>
<tr>
<td>Jacques</td>
<td>39, rue de Bretagne, Paris</td>
</tr>
<tr>
<td>Isabelle</td>
<td>20 Rue D'orsel, Paris</td>
</tr>
</tbody>
</table>
Running Example

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amelie</td>
<td>65, quai d'Orsay, Paris</td>
</tr>
<tr>
<td>Jacques</td>
<td>39, rue de Bretagne, Paris</td>
</tr>
<tr>
<td>Isabelle</td>
<td>20 Rue D'orsel, Paris</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beret</td>
<td>3</td>
</tr>
</tbody>
</table>

Europe

USA

Dedup

Union

Predict

ItemAgg

ItemVolumes

CustList_1

CustList_2

...}

CustList_{n-1}

CustList_n

ClothCo Items

Buying Patterns

Name

Address

Item

Demand
Processing Nodes

Relational Nodes: Structured, well-understood operations
Processing Nodes

CustList\textsubscript{1} \quad \text{Dedup} \quad \text{Europe} \quad \text{Union} \quad \text{Predict} \quad \text{ItemAgg} \quad \text{O}

CustList\textsubscript{2} \quad \text{Dedup} \quad \text{USA} \quad \text{Union} \quad \text{Predict} \quad \text{ItemAgg} \quad \text{O}

... \quad \text{Dedup} \quad \text{Europe} \quad \text{Union} \quad \text{Predict} \quad \text{ItemAgg} \quad \text{O}

CustList\textsubscript{n-1} \quad \text{Dedup} \quad \text{Europe} \quad \text{Union} \quad \text{Predict} \quad \text{ItemAgg} \quad \text{O}

CustList\textsubscript{n} \quad \text{Dedup} \quad \text{Europe} \quad \text{Union} \quad \text{Predict} \quad \text{ItemAgg} \quad \text{O}

Opaque Nodes
Predicted Uses

• **Explanation**
  – How was data derived?

• **Verification**
  – Is data erroneous or outdated?

• **Recomputation**
  – Can data be recomputed efficiently?
Processing Nodes

Relational nodes: structured, well-understood operations
Processing Nodes

Opaque nodes
Provenance Operations

- **Basic operations**
  - Backward tracing
    - Where did the cowboy-hat record come from?
  - Forward tracing
    - Which predictions did this customer contribute to?

- **Examples of additional functionality**
  - Forward propagation
    - Update all affected predictions after customers move from France to Texas
  - Refresh \(\approx\) Backward tracing + forward propagation
    - Update only the cowboy hat record given updated customer lists
Provenance Operations

- Examples of additional functionality
  - Forward propagation
    - Update all affected predictions after customers move from France to Texas
  - Refresh \(\approx\) Backward tracing + forward propagation
    - Update only the cowboy hat record given updated customer lists

CustList\(_1\) → Dedup → Europe
CustList\(_2\) → Dedup → USA
CustList\(_{n-1}\) → Dedup
CustList\(_n\) → Dedup

Dedup → Union → Predict → ItemAgg → ItemVolumes

- Examples of additional functionality
  - Forward propagation
    - Update all affected predictions after customers move from France to Texas
  - Refresh \(\approx\) Backward tracing + forward propagation
    - Update only the cowboy hat record given updated customer lists
Provenance Operations

- **Basic operations**
  - **Backward tracing**
    - Where did the cowboy-hat record come from?
  - **Forward tracing**
    - Which predictions did this customer contribute to?

- **Examples of additional functionality**
  - **Forward propagation**
    - Update all affected predictions after customers move from France to Texas
  - **Refresh ≈ Backward tracing + forward propagation**
    - Update only the cowboy hat record given updated customer lists
Provenance Queries

- Seamlessly combine provenance and data
- Compact and intuitive language
- Amenable to optimization
- Examples:
  - How many people from each country contributed to the cowboy hat prediction?
  - Which customer list contributed the most to the top 100 predicted items?
Provenance Queries

• **Examples:**
  – How many people from each country contributed to the cowboy hat prediction?
  – Which customer list contributed the most to the top 100 predicted items?

• Seamlessly combine provenance and data

• Compact and intuitive language

• Amenable to optimization
Processing Nodes

Relational nodes: structured, well-understood operations
Processing Nodes

Opaque nodes
Example

```
CustList_1  
<table>
<thead>
<tr>
<th></th>
<th>CustList_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>CustList_{n-1}</td>
<td></td>
</tr>
</tbody>
</table>

Dedup

Europe

Union

Predict

ItemAgg

ItemVolumes

ClothCo Items

Buying Patterns

Name

Address

Amelie
65, quai d'Orsay, Paris

Jacques
39, rue de Bretagne, Paris

Isabelle
20 Rue D'Orsel, Paris

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amelie</td>
<td>65, quai d'Orsay, Paris</td>
</tr>
<tr>
<td>Jacques</td>
<td>39, rue de Bretagne, Paris</td>
</tr>
<tr>
<td>Isabelle</td>
<td>20 Rue D'Orsel, Paris</td>
</tr>
</tbody>
</table>

Item

Demand

Beret

3

<table>
<thead>
<tr>
<th>Item</th>
<th>Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beret</td>
<td>3</td>
</tr>
</tbody>
</table>
```