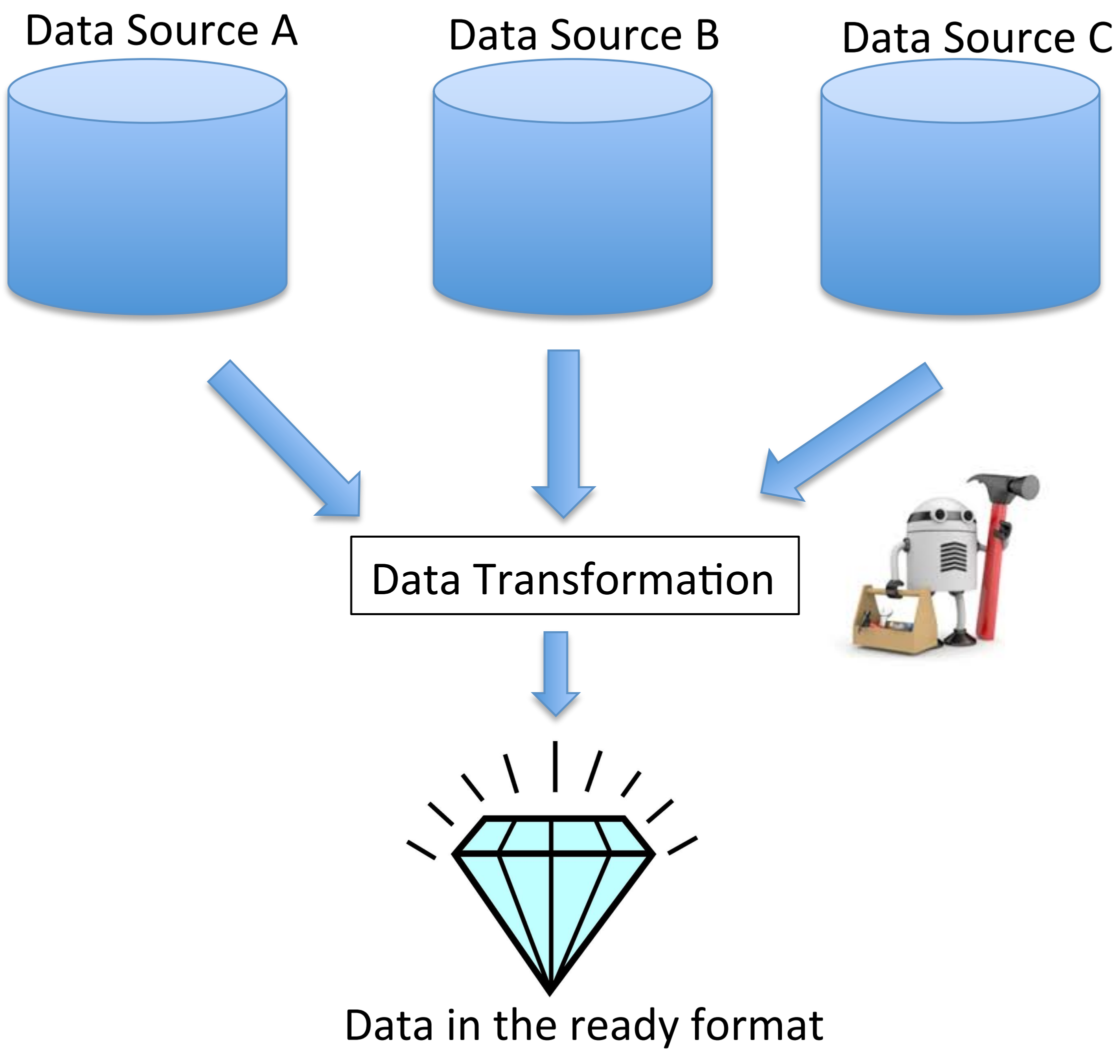
Iteratively Learning Conditional Statements in Transforming Data by Example

Bo Wu and Craig A. Knoblock
University of Southern California

Introduction

Motivation


Data Source A
Data Source B
Data Source C
Data Transformation
Data in the ready format

A Data Table

Accession	Credit	Dimensions	Medium	Name
01.2	Gift of the artist	5.25 in HIGH x 9.375 in WIDE	Oil on canvas	John Mix Stanley
05.411	Gift of James L. Edison	20 in HIGH x 24 in WIDE	Oil on canvas	Mortimer L. Smith
06.1	Gift of the artist	Image: 20.5 in. HIGH x 17.5 in. WIDE	Oil on canvas	Theodore Scott Dabo
06.2	Gift of the artist	9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE	Oil on canvas	Leon Dabo
…				
09.8	Gift of the artist	12 in|14 in HIGH x 16 in|18 in WIDE	Oil on canvas	Gari Melchers

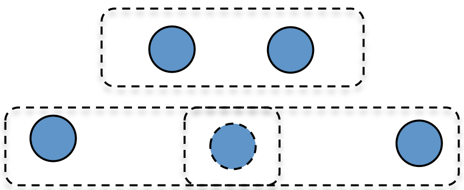
Programming by Example

	Raw Value	Target Value
R1	5.25 in HIGH x 9.375 in WIDE	9.375
R2	20 in HIGH x 24 in WIDE	24
R3	Image: 20.5 in. HIGH x 17.5 in. WIDE	17.5
R4	9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE	null 19.5 **19.5**
	. . .	
R5	12 in|14 in HIGH x 16 in|18 in WIDE	null 18

Problem:

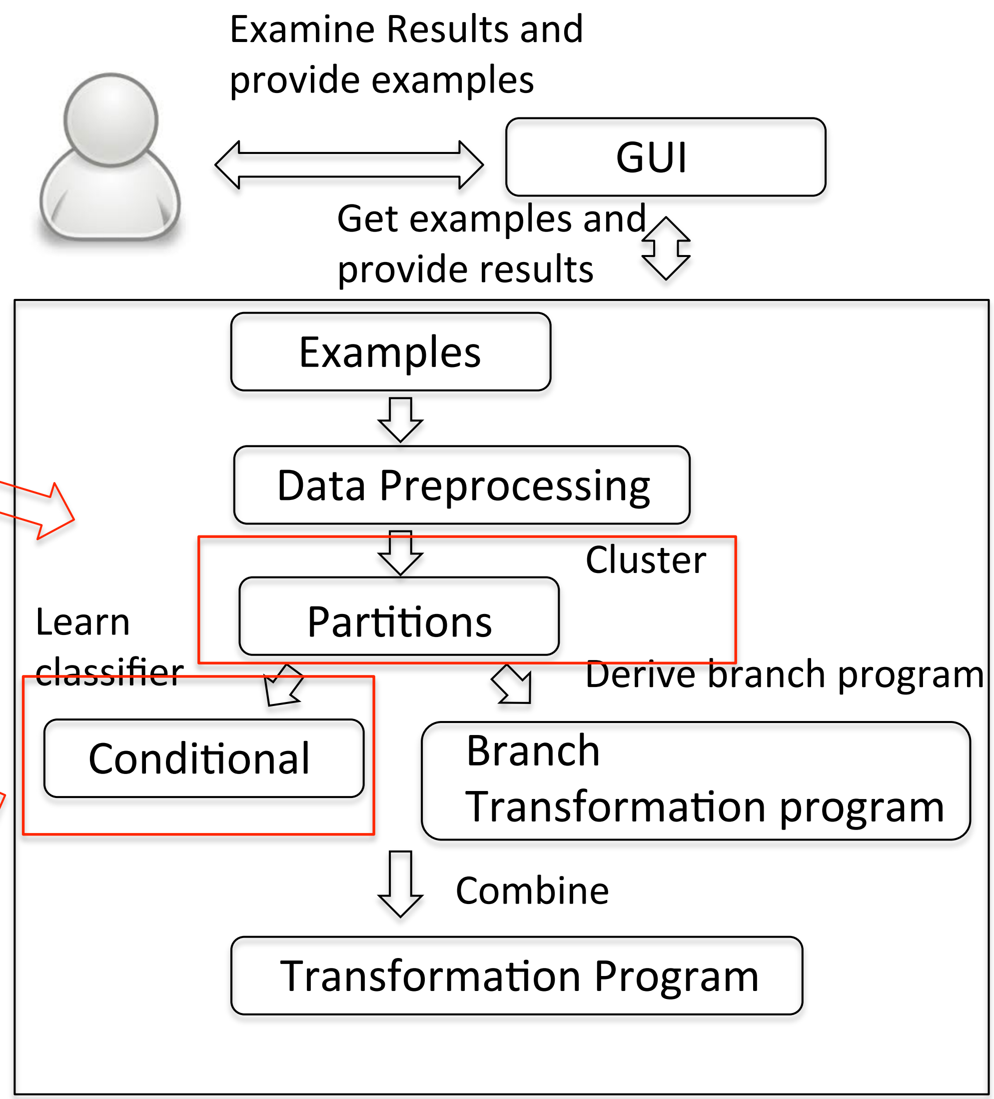
Learn accurate conditional statements efficiently for data with heterogeneous formats using few examples

Previous Approach



Compatibility score ($O(n^3)$)

Few Training Data



Transformation Program

BNK: blankspace
NUM[0-9]+: 98
UWRD[A-Z]: I
LWRD[a-z]: mage
WORD[a-zA-Z]
START:
END:

$Transform(value)$

Conditional Statement

label = classify(value)

switch label:

case "format1":

Branch Transformation Program

$pos_1 = value.indexOf('BNK', 'NUM', -1)$

$pos_2 = value.indexOf('NUM', 'BNK', 2)$

$output = value.substr(pos_1, pos_2)$

case "format2":

Branch Transformation Program

$pos_3 = value.indexOf('|', 'NUM', 2)$

$pos_4 = value.indexOf('NUM', 'BNK', -1)$

$output = value.substr(pos_3, pos_4)$

return output

Example: 9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE ➔ 19.5

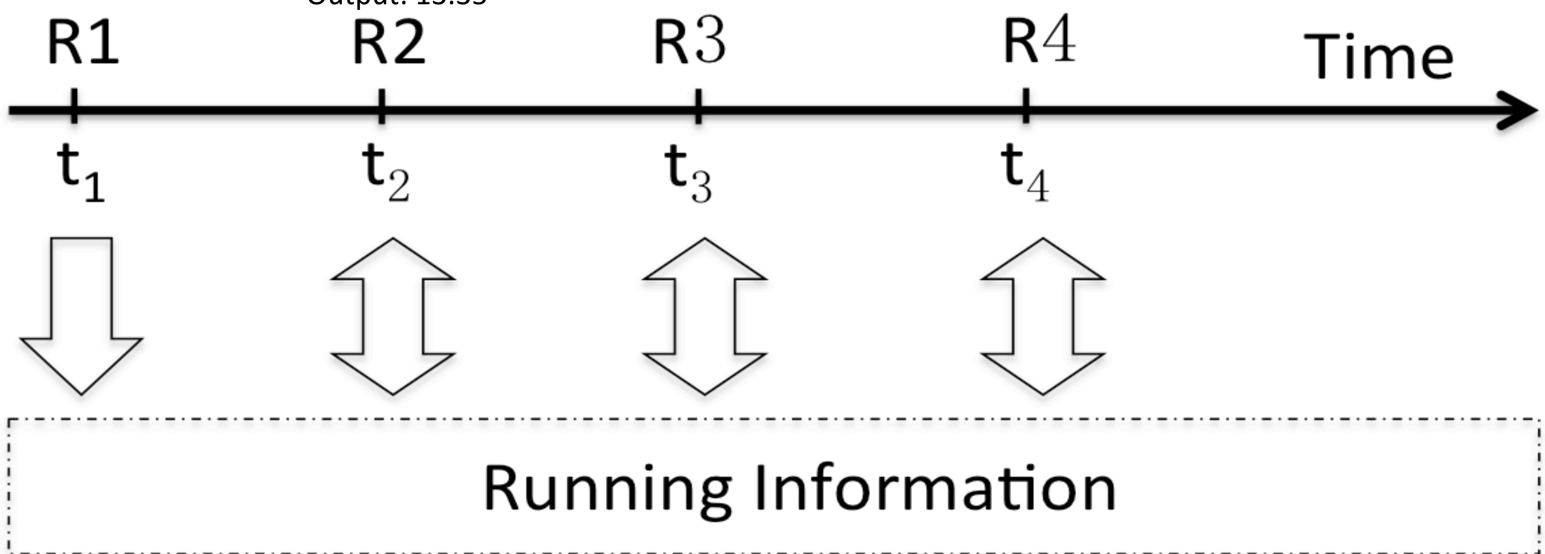
Our Approach

Main Idea

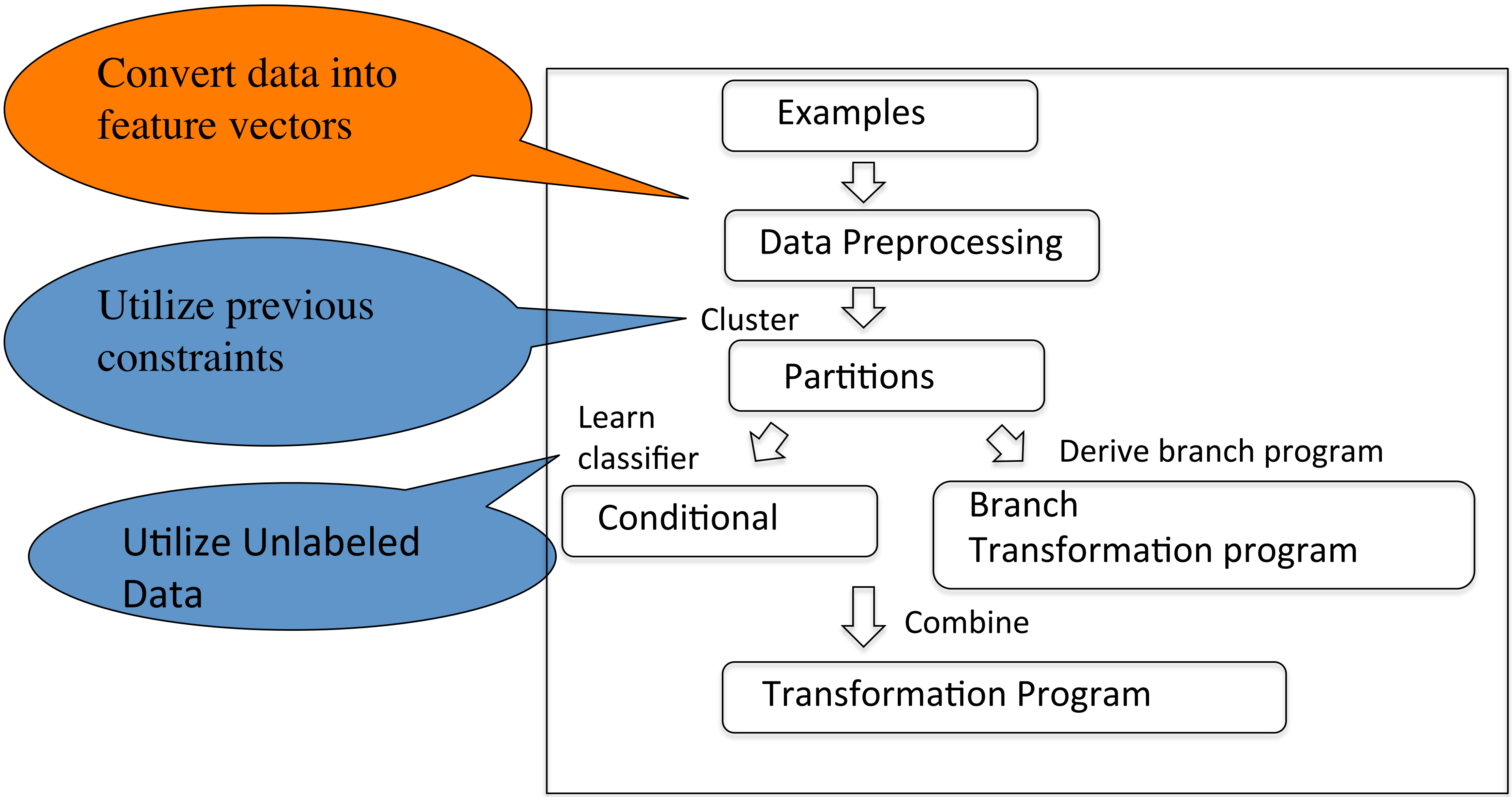
Learning the conditional statement iteratively

Input: 5.25 in HIGH x 9.375 in WIDE
Output: 9.375

Input: 9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE
Output: 13.35



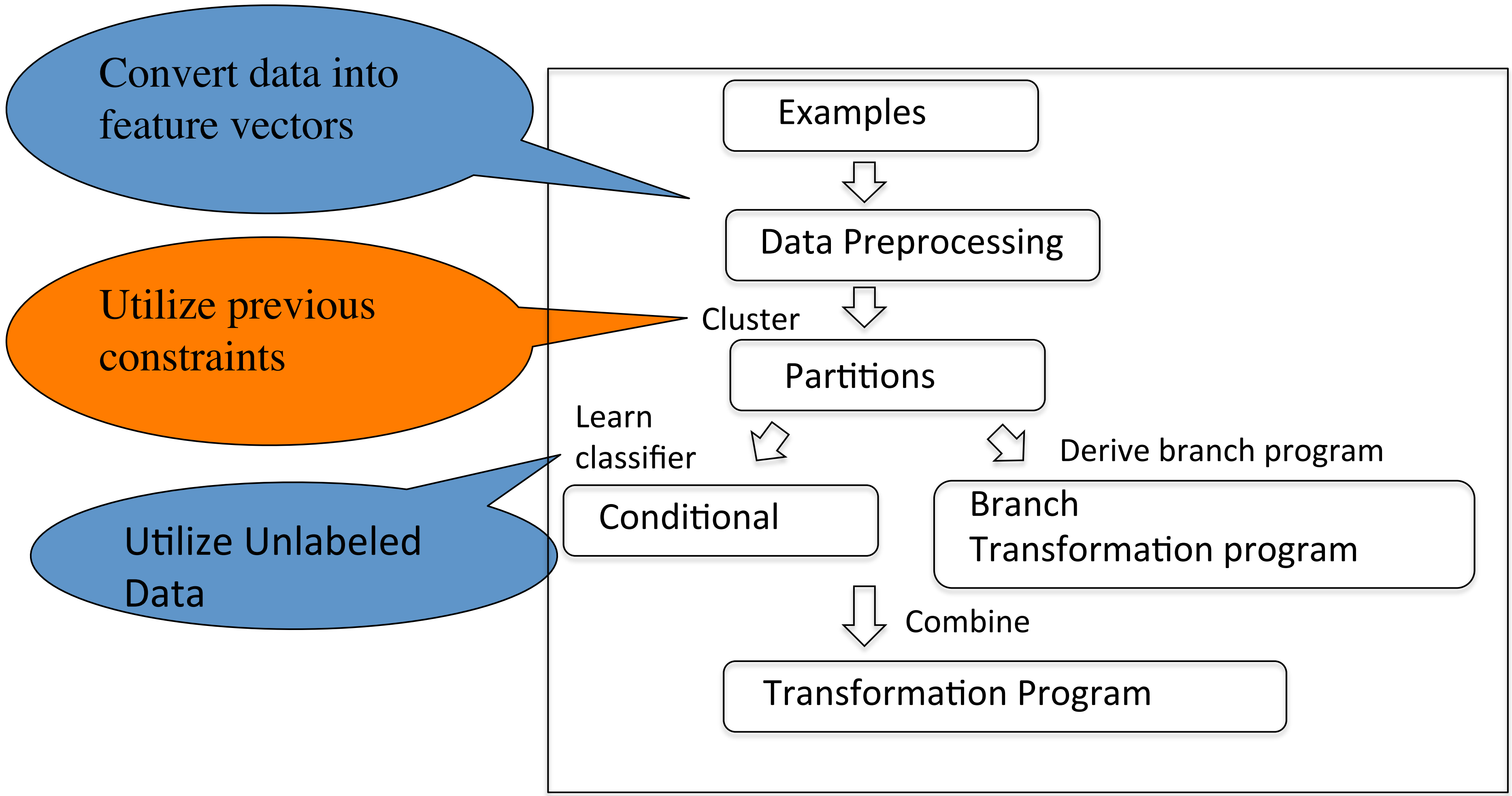
Our Approach



Data Preprocessing

<table>
<tr><td>String</td><td>9.75 in|16 in HIGH</td></tr>
<tr><td>Tokens</td><td>START NUM(9) Period(.) NUM(75) BNK LWRD(in) VBAR(|) NUM (16) BNK LWRD(in) BNK UWRD(H) UWRD(I) UWRD(G) UWRD(H) …</td></tr>
<tr><td>Token counts</td><td><table>
<tr><td>NUM</td><td>UWRD</td><td>LWRD</td><td>BNK</td><td>.</td><td>|</td><td>H</td><td>I</td><td>G</td><td>in</td><td>…</td></tr>
<tr><td>3</td><td>4</td><td>2</td><td>3</td><td>1</td><td>1</td><td>2</td><td>1</td><td>1</td><td>2</td><td>0</td></tr>
</table></td></tr>
<tr><td>Feature Vector</td><td><table>
<tr><td>LWRD</td><td>NUM</td><td>.</td><td>:</td><td>|</td><td>=</td></tr>
<tr><td>0.21</td><td>0.29</td><td>0.14</td><td>0.21</td><td>0.07</td><td>0.07</td></tr>
</table></td></tr>
</table>

Our Approach



Constraints

- Two Types of Constraints:
  - Cannot-merge Constraints:
    - Ex:

5.25 in HIGH x 9.375 in WIDE	9.375
9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE	13.75
20 in HIGH x 24 in WIDE	24

  - Must-merge Constraints:
    - Ex:

P1

5.25 in HIGH x 9.375 in WIDE	9.375
20 in HIGH x 24 in WIDE	24

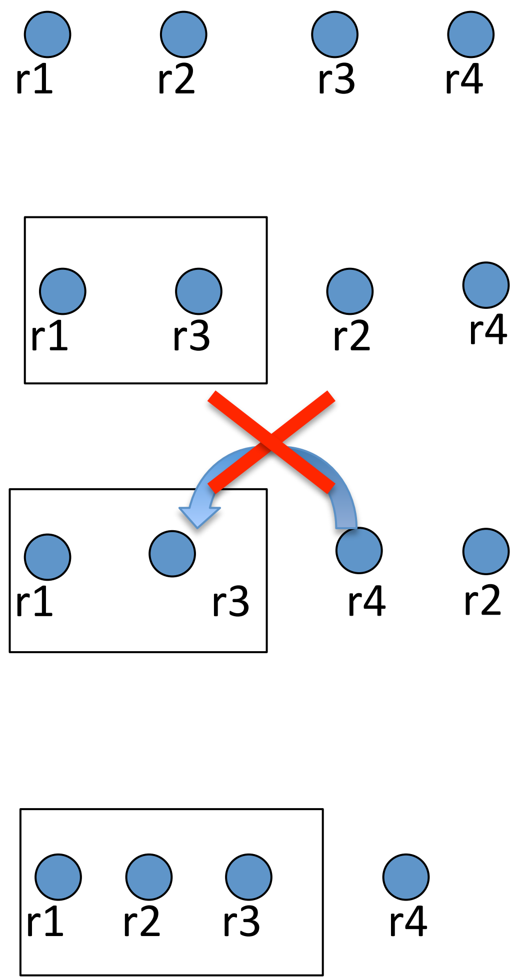
P2

9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE	13.75

P3

Image: 20.5 in. HIGH x 17.5 in. WIDE	17.5

Constrained Agglomerative Clustering



Update constraints
Learn distance metric

Distance Metric Learning

- Distance Metric (Weighted Euclidean) Learning

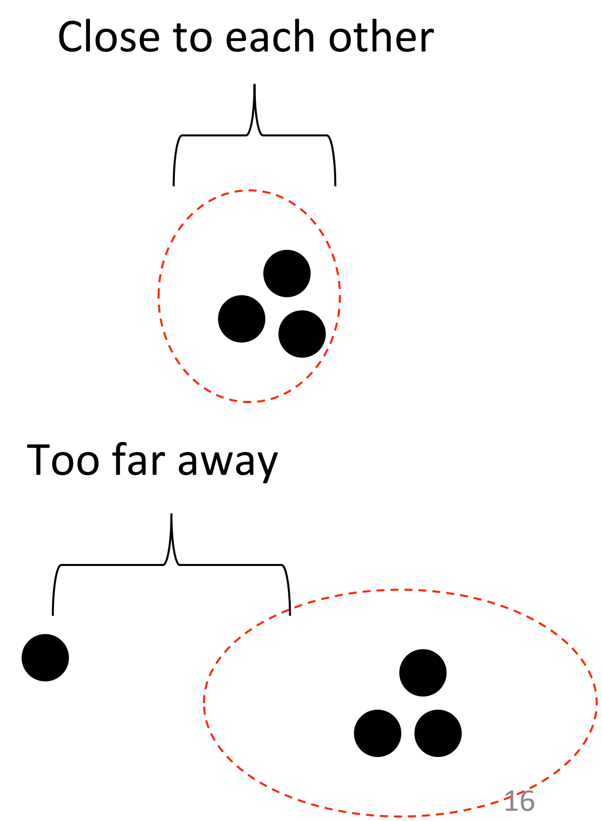
$$d(x,y) = \|x-y\|_w = \sqrt{\sum_i w_i (x_i - y_i)^2}$$

- Objective Function

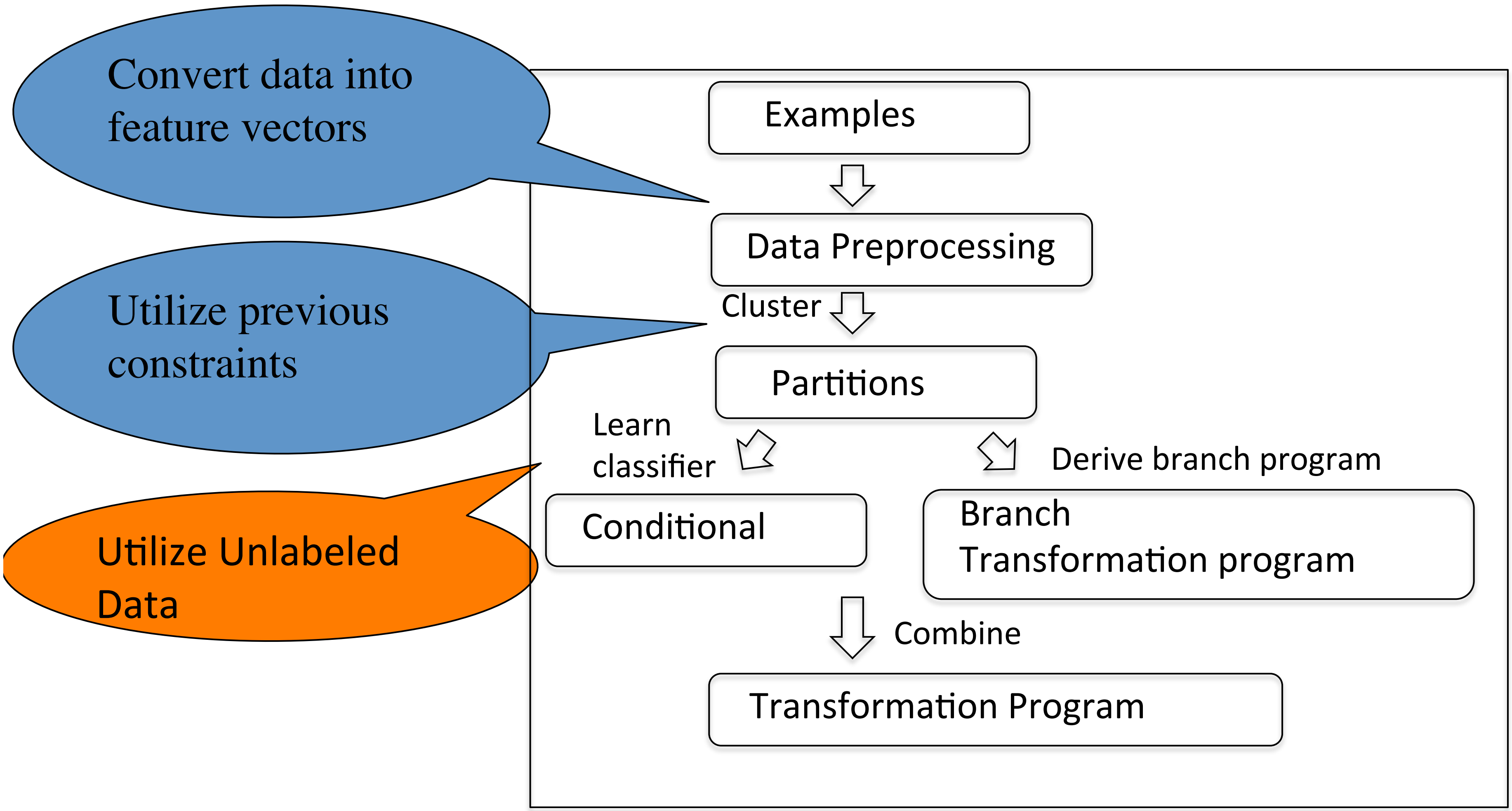
$$argmin_{w>0} \sum_i \|x_i - e_{x_i}\|_w + a * g(w) - b * h(w)$$

$$g(w) = ln(\sum_{X_m} \sum_{x_i, x_j \in X_m, i \neq j} \|x_i - x_j\|_w)$$

$$h(w) = ln \sum_{X_r} max_{x_i, x_j \in X_r} \|x_i - x_j\|_w$$



Our Approach



Utilize Unlabeled data in Learning Classifier

Partition 1

Examples	5.25 in HIGH x 9.375 in WIDE	9.375
	20 in HIGH x 24 in WIDE	24
	Image: 20.5 in. HIGH x 17.5 in. WIDE	17.5

Unlabeled	26 in. HIGH x 23 in. WIDE
	19.75 in HIGH x 22.75 in WIDE x 0.25 in DEEP
	33.5 in HIGH x 39 in WIDE
	...

Partition 2

Examples	9.75 in|16 in HIGH x 13.75 in|19.5 in WIDE	13.75

Unlabeled	12 in|14 in HIGH x 16 in|18 in WIDE
	20.25 in|19.75 in HIGH x 15.75 in|15.875 in WIDE
	55 in HIGH x 46 in|290 in WIDE
	...

Filter unlabeled data

1. Filter unlabeled data on the boundary

2. Only choose top K unlabeled data

Learn a SVM classifier

Results

Evaluation

- Dataset: 30 editing scenarios
  - Museum
  - Google Refine and Excel user forums
- Comparing Methods:
  - **SP**
    - The state-of-the-art approach that uses compatibility score to select partitions to merge
  - SPIC
    - Utilize previous constraints besides using compatibility score
  - DP
    - Learn distance metric
  - DPIC
    - Utilize previous constraints besides learning distance metric
  - **DPICED**
    - Our approach in this paper

Results

Success Rates:

	DPICED	DPIC	DP	SPIC	SP
SccRate	1	1	0.97	0.77	0.77

Time and Examples:

	Total Time (seconds)	Examples	Constraint Number
DPICED	3.9	5.4	6.1
DPIC	6.4	6.8	6.6
DP	8.3	6.8	17.6
SPIC	21.3	6.8	260.1
SP	26.5	6.9	305.8

Related Work

- Wrapper induction approaches
  - WIEN [Kushmerick, 1997], SoftMealy [Hsu et al., 1998], STALKER [Muslea et al., 1999]
- Programming-by-example approaches
  - FlashFill[Gulwani, 2011][Perelman et al., 2014], Data Wrangler [Kandel et al., 2011], SmartEditor [Lau et al. 2003]
- Clustering with constraints
  - Clustering with constraints [Xing et al., 2002][Bilenko et al., 2004][Bade et al., 2006][Zhao et al., 2010] [Zheng et al., 2011]

Discussion

- Iteratively learn conditional statements in PBE setting
  - Improve the efficiency
  - Learn more accurate conditional statements
  - generate a small number of branches.
- Incorporate ML tools as external functions in inductive programming

Future Work

- Integrate the partitioning and classification steps
  - Reduce accumulated errors
- Improve GUI to help user verifying the data
  - Identify unseen formats
  - Identify incorrectly classified records

- Thanks