Entity Resolution: Probabilistic Approaches

Data Cleaning & Integration

CompSci 590.01 Spring 2017

DUKE
COMPUTER SCIENCE

Some contents are ased on:
Getoor & Machanavajjhala's *VLDB* 2012 tutorial slides

Outline

- Data preparation and matching features
- Pairwise-ER
- Leveraging constraints in ER
  - Record linkage: exclusivity
  - Deduplication: transitivity
  - "Collective" ER: general

Two approaches highlighted

- Bhattacharya & Getoor, *SDM* 2007
  - Capture dependencies using a generative probabilistic model (LDA in this case)

- Singla & Domingos, *ICDM* 2006
  - Capture a set of hard and soft constraints using an undirected model, using a logic-based syntax (MLN in this case)

LDA for entity resolution

- Define a generative model where (potentially ambiguous) references are generated from entities (with true identities)
  - E.g., references ("Aho", "A. V. Aho") vs. authors (Alfred V. Aho)
- Capture any useful, additional dependencies as part of the model
  - E.g., people in the same collaborative group tend to publish together; those in different groups tend not to
- References are observed; true identities and groups are hidden: infer the hidden from the observed
  - Unsupervised

Example

P1: "JOSTLE: Partitioning of Unstructured Meshes for Massively Parallel Machines" C. Walshaw, M. Cross, M. G. Everett, S. Johnson
P2: "Partitioning Mapping of Unstructured Meshes to Parallel Machine Topologies", C. Walshaw, M. Cross, M. G. Everett, S. Johnson, K. McManus
P3: "Dynamic Mesh Partitioning: A Unified Optimisation and Load-Balancing Algorithm", C. Walshaw, M. Cross, M. G. Everett
P4: "Code Generation for Machines with Multiregister Operations", Alfred V. Aho, Stephen C. Johnson, Jefferey D. Ullman
P5: "Deterministic Parsing of Ambiguous Grammars", A. V. Aho, S. C. Johnson, J. D. Ullman
P6: "Compilers: Principles, Techniques, and Tools", A. Aho, R. Sethi, J. Ullman

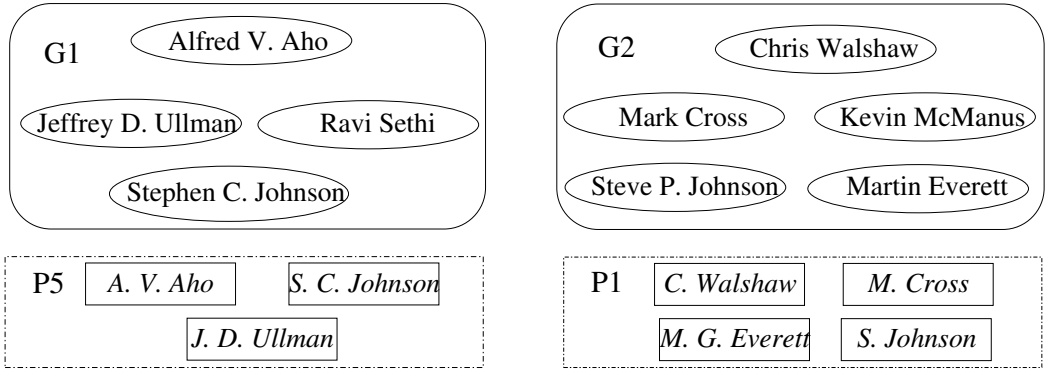
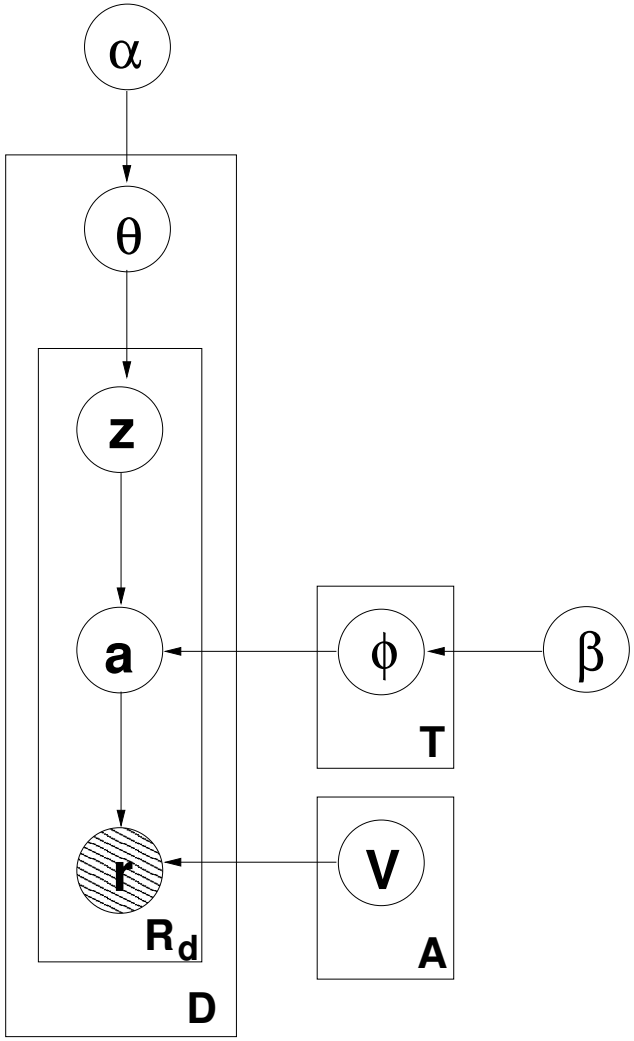


Figure 1: Author entities in two different collaboration groups and two generated papers. The ovals are the entities belonging to groups shown as encapsulating rectangles. Dotted rectangles represent papers with author references shown as smaller solid rectangles. Each paper is generated by the group above it.

Even though "S. C. Johnson" (in P5) and "S. Johnson" (in P1) seem to refer to the same author, we should infer otherwise based on the two collaboration groups

LDA-ER model



- Suppose there are T possible author groups; for each group z, choose a distribution ϕ^z (from a Dirichlet distribution with hyper-parameter β) over the A possible authors
- For each paper d, choose a distribution θ^d (from a Dirichlet distribution with hyper-parameter α) over the author groups
- To generate one of the R^d author references for d, first pick group z from θ^d, then pick author a from ϕ^z, and finally, apply the noise model V^a to obtain r

Dirichlet process

- A distribution over distributions
- Allows the model complexity to grow with increasing data—e.g., no need to specify the number of clusters in advance
  - Intuitively, when drawing the n-th component distribution (cluster), the probability of choosing an existing component is proportional to # times chosen in the previous $n-1$ draws, but a new component has a nonzero probability of being sampled
- LDA-ER augments the above with the additional author group structure, allowing A to be learned; T is fixed, however
  - See paper for details

Customized noise model

- First and middle names can be initialed, dropped, or retained
- All strings can then be corrupted with character insertion, deletion, or replacement

- Noise model parameters start from an initial estimate typically of datasets in the domain
- During inference, they evolve (slowly) based on the sampled authors and references

Inference

- Basic Gibbs sampling: iteratively sample the values of the hidden group and author labels for each reference conditioned on existing labels for all other references
- Use a block-based algorithm that basically mimics agglomerative clustering + allowing clusters to split
  - A cluster (references with the same author label) can stay unchanged, merge with another, or split and have a part assigned to an unassigned author label

Two approaches highlighted

- Bhattacharya & Getoor, *SDM* 2007
  - Capture dependencies using a generative probabilistic model (LDA in this case)

- Singla & Domingos, *ICDM* 2006
  - Capture a set of hard and soft constraints using an undirected model, using a logic-based syntax (MLN in this case)

Markov Logic: the idea

- A logical KB (knowledge base) is a set of hard constraints on the set of "possible worlds"
- Make constraints soft—when a world violates a formula, it becomes less probable but not impossible
- Give each formula a *weight*
  - Higher weight means stronger constraint

$$P(\text{world } w) \propto \exp\left(\sum \text{weights of formulas } w \text{ satisfies}\right)$$

Markov Logic Network

A Markov Logic Network (MLN) is a set of pairs (F_i, w_i) where

- F_i is a formula in first-order logic
- w_i is a real number

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right)$$

where Z is a normalizing constant and n_i is the number of true groundings of F_i in x

MLN example

- Two constants in KB: {Anna, Bob}
- Two predicates, Smokes & Cancer
- One formula with weight w:

$$\forall x: \text{Smokes}(x) \Rightarrow \text{Cancer}(x)$$

- Possible worlds:

Smokes(Anna)	Cancer(Anna)	Smokes(Bob)	Cancer(Bob)
0	0	0	0
0	0	0	1
0	0	1	0
1	0	1	0
…	…	…	…
1	1	1	1

# true groundings	PZ
2	$\exp(2w)$
2	$\exp(2w)$
1	$\exp(w)$
0	$\exp(0)$
…	…
2	$\exp(2w)$

Formulating ER with MLN

- Given a database of records, each represented as a set of typed predicates
  - E.g., HasTitle(paper, title), HasAuthor(paper, author), HasVenue(paper, venue)
- Infer, for each pair of constants of the same type, whether they refer to the same entity

- Approach: introduce an "equality" query predicate and define formulas to capture the constraints useful to ER

Capturing equality

- Introduce predicate Equals(x, y), or $x \sim y$ for short
- Introduce the axioms of equality
  - Reflexivity: $\forall x: x \sim x$
  - Symmetry: $\forall x, y: x \sim y \Rightarrow y \sim x$
  - Transitivity: $\forall x, y, z: (x \sim y) \wedge (y \sim z) \Rightarrow x \sim z$
  - Predicate equivalence: for each predicate R:
    $\forall x_1, x_2, y_1, y_2:$
    $(x_1 \sim x_2) \wedge (y_1 \sim y_2) \Rightarrow \big(R(x_1, y_1) \Leftrightarrow R(x_2, y_2)\big)$
- Use ∞ as weight for rules above (i.e., these are hard constraints)

Soft evidence

- Reverse predicate equivalence: for each predicate R: $\forall x_1, x_2, y_1, y_2$:
  $R(x_1, y_1) \wedge R(x_2, y_2) \Rightarrow \big((x_1 \sim x_2) \Leftrightarrow (y_1 \sim y_2)\big)$
- Intuitively, if two objects are in the same relation to the same object, there is evidence that they may be the same object
- Example:
  $\text{HasAuthor}(c_1, \text{J. Cox}) \wedge \text{HasAuthor}(c_2, \text{Cox J.})$
  $\Rightarrow \big((c_1 \sim c_2) \Leftrightarrow (\text{J. Cox} \sim \text{Cox J.})\big)$
- Certainly not true logically, so give it a finite weight but not ∞
- Remarkably, this is good enough to get ER started!

Field comparison

- Whole-string comparison on fields is too crude
- Introduce HasWord(*field*, *word*)
  - Reverse predicate equivalence would yield $\forall x_1, x_2, y_1, y_2$: HasWord$(x_1, y_1) \wedge$ HasWord$(x_2, y_2) \wedge (y_1 \sim y_2) \Rightarrow x_1 \sim x_2$, implying a simple token-based similarity measure on fields
- Example: HasWord(J. Cox, Cox) ∧ HasWord(Cox J., Cox) ∧ (Cox ∼ Cox) ⇒ J. Cox ∼ Cox J.
- Can specify a different weight for each word
- Can use n-grams to handle misspelling
- Can add negative evidence, e.g.:
  - $\forall x_1, x_2, y_1, y_2$: HasWord$(x_1, y_1) \wedge \neg$HasWord$(x_2, y_2) \wedge (y_1 \sim y_2) \Rightarrow \neg(x_1 \sim x_2)$

Going more "collective"

- Introduce Coauthor (with ∞ weight) and then apply reverse predicate equivalence (with some weight):
  - $\forall x, y_1, y_2$: HasAuthor(x, y_1) ∧ HasAuthor(x, y_2) ⇒ Coauthor(y_1, y_2)
- Introduce a formula (with a high weight) to capture that two authors (y_3 and y_4) are likely the same if both their papers (x_1 and x_2) and coauthors (y_1 and y_2) are the same:
  - $\forall x_1, x_2, y_1, y_2, y_3, y_4$:
    HasAuthor(x_1, y_1) ∧ HasAuthor(x_2, y_2)
    ∧ HasAuthor(x_1, y_3) ∧ HasAuthor(x_2, y_4)
    ∧ $(x_1 \sim x_2)$ ∧ $(y_1 \sim y_2)$
    ⇒ $y_3 \sim y_4$

Inference

- Too many ~ pairs to infer
  - Use a cheap heuristic to identify non-matches, and just adding them as false atoms
  - But allow them to be revisited later, to correct mistakes made by the heuristic
- Learning: supervised and transfer
  - Can define formulas and learn weights in one domain and transfer them to another
- Inference algorithm (MaxWalkSat) based on stochastic local search
  - Beginning with a random truth assignment, repeatedly flip the truth value of an atom to maximize an objective function, or a random atom in an unsatisfied clause

Summary

- Both supervised & unsupervised approaches exist to exploit constraints and relationships for "collective" ER
- Real world is complex and uncertain: use logic to help specify lots of complex dependencies; use statistics to help cope with uncertainty

- Next (following the project progress update):
  - Two more papers on collective ER
  - Then we move on to efficiency/scalability issues for ER