Learning to Match the Schemas of Data Sources:

A Multistrategy Approach

ANHAI DOAN
PEDRO DOMINGOS
ALON HALEVY

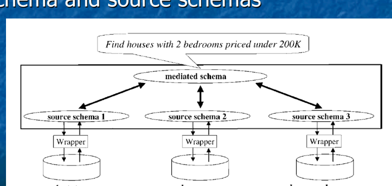
Presented by: Shimin Guo

Overview

- Introduction
- Multistrategy learning
- Base learners
- Meta-learner
- Empirical evaluation
- Summary

Introduction

- Goal: A uniform query interface to a multitude of data sources
- Schema matching: between the mediated schema and source schemas



Assumption

- All schemas represented with XML DTDs
- Sources present data in XML
- Find only one-to-one mappings

Observation

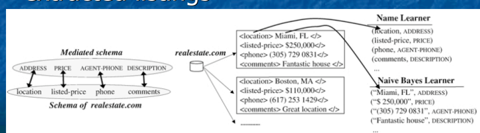
- Many types of information can be exploited
  - name, data format, word frequency…
- A specific type of information may be especially useful for some schema elements, while less so for others
- Solution: multistrategy Learning

Multistrategy learning

- Training:
  - Manually map a small set of source schemas to the mediated schema
  - Multiple "base learners" learn from these examples, each from a different perspective
  - A "meta-learner" give weights to base learners with regard to each schema element based on the learners' performance on that element (cross validation)
- Matching:
  - each base learner makes predictions independently
  - the final prediction is the weighted average of each individual predictions

Training – base learners

- Manually map a small set of source schemas to the mediated schema
- Extract data listings from these sources
- Train the base learners using the extracted listings



Matching – base learners

- Extract a set of data listings from a new source
- The unit of matching (at the base learner level) is one instance of an element
- Predictions made by each single learner is of the form:
  - $<(c_1, s_1),\ldots,(c_n, s_n)>$
  - instance matches element c_i with confidence score (probability) s_i

Types of base learners

- Name learner
  - stores training samples of the form (expanded tag-name, label)
  - makes prediction based on similarity of expanded tag-name
  - similarity measure: TF/IDF distance
  - measure is large if two documents share many important terms, and small otherwise
  - works well on descriptive names
  - not good at names that don't share synonyms

Types of base learners

- Content learner
  - stores training examples in the form of (data-value, label)
  - otherwise the same as name learner
  - works well on long textual elements, e.g. house description, or elements with descriptive values, e.g. color
  - not good at short, numeric elements

10

Types of base learners

- Naive Bayes learner
  - tokenizes data instances by parsing and stemming the words
  - for instance $d=\{w_1,\ldots w_k\}$, predictions are
    - $<(c_1, P(c_1|d)),\ldots, (c_n, P(c_n|d))>$
    - $P(c_j|d) \propto P(c_j)\, P(d|c_j)$ (Bayes' rule)
    - $P(c_j)$: fraction of training instances with label c_j
    - $P(d|c_j) = P(w_1|c_j)\, P(w_2|c_j)\ldots P(w_k|c_j)$
    - $P(w_1|c_j)$: frequency of w_1 in all training instances with label c_j
  - assumes tokens appear independently of each other given the label (which is generally not true)
  - works well when word frequency matters, not good at numerical fields

11

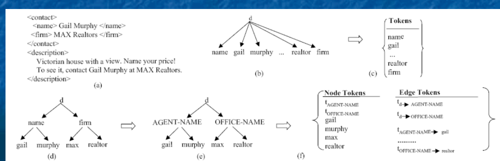
Types of base learners

- County-name recognizer
  - searches a database to verify if a data instance is a county name
  - an example of how recognizers with a narrow and specific area of expertise can be incorporated

12

Types of base learners

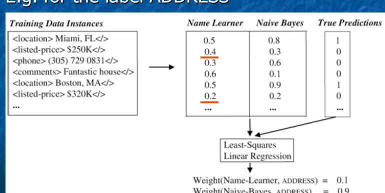
- XML learner
  - exploits structure information
  - similar to naive Bayes learner except that it also considers *structure* tokens in addition to *text* tokens.



13

Training meta-learner

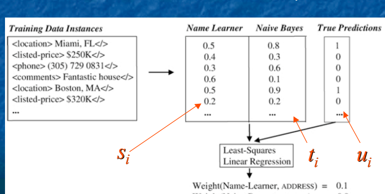
- For each label of the mediated schema, ask all base learners to give a confidence score associated with that label for every training data instances, and compare those to the correct answers
- E.g. for the label ADDRESS



14

Training meta-learner

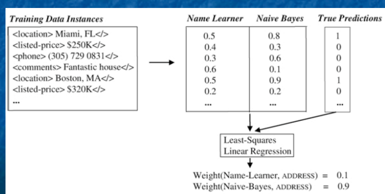
- Performs a least-square linear regression
- minimize $\sum [u_i - (s_i \times w_{NameLearner}^{ADDRESS} + t_i \times w_{NaiveBayes}^{ADDRESS})]^2$
- subject to $w_{NameLearner}^{ADDRESS} + w_{NaiveBayes}^{ADDRESS} = 1$



15

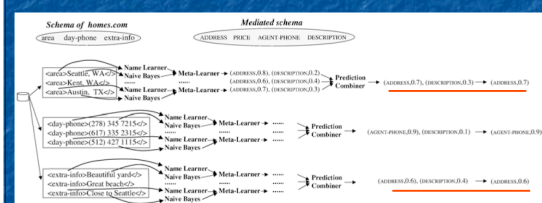
Training meta-learner

- Base learners should not make predictions for data instances they've been trained with
- solution: 5-fold Cross validation



16

Matching



17

Constraint handler

- Searches through the space of possible mapping conbinations to find the one with the lowest cost
- Cost is defined based on the likelihood of the mapping combination and how it conforms to the domain constraints

Constraint Types		Examples	Can Be Verified With
Hard	*Frequency*	At most one source element matches HOUSE. Exactly one source element matches PRICE.	Schema of target source
	Nesting	If **a** matches AGENT-INFO & **b** matches AGENT-NAME, then **b** is nested in **a.** If **a** matches AGENT-INFO & **b** matches PRICE, then **b** cannot be nested in **a**.	"
	Contiguity	If **a** matches BATHS & **b** matches BEDS, then **a** & **b** are siblings in the schema-tree, and the elements between them (if any) can only match OTHER.	"
	Exclusivity	There are no **a** and **b** such that **a** matches COURSE-CREDIT & **b** matches SECTION-CREDIT.	"
	Column	If **a** matches HOUSE-ID, then **a** is a key. If **a**, **b**, and **c** match CITY, FIRM-NAME, and FIRM-ADDRESS, resp., then **a** & **b** functionally determine **c**.	Schema + data from target source
Soft	*Binary*	Number of elements that match DESCRIPTION is not more than 3.	"
	Numeric	If **a** matches AGENT-NAME & **b** matches AGENT-PHONE, then we prefer **a** & **b** to be as close to each other as possible, all other things being equal.	Schema of target source

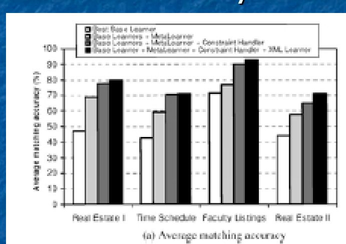
18

Empirical evaluation

- Four domains
- 5 sources per each domain
- Source schemas converted to XML DTD manually
- In each run, chose 3 sources for training, the rest for matching
- Metric: percentage of matchable source-schema tags matched correctly

19

Accuracy

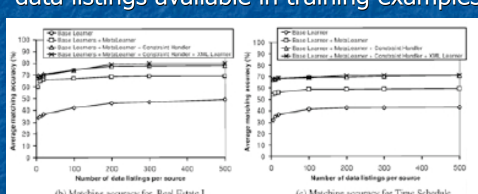


- From left to right: best single base learner, meta-learner using base learners, plus domain constraint handler, plus XML learner

20

Performance sensitivity

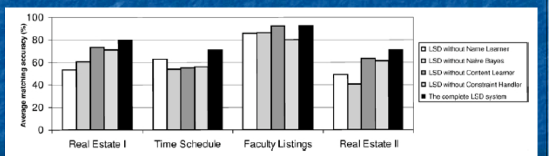
- To test the sensitivity to the number of data listings available in training examples



21

Lesion studies

- To test the contribution of each base learner and the constraint handler



22

Advantages

- Highly flexible and extensible
  - new learner modules can be incorporated easily
- Accounts for different levels of usefulness of a specific type of information with regard to different labels
- No need for parameter tuning

23

Disadvantage

- Source DTDs are usually not available
- Need to manually do some mapping to get started
- Sample size used in the experiments too small to be significant
- Did not show how performance will change as the size of training sample increases

24

Summary

- The multistrategy learning approach utilizes both schema and data
- A set of learners, each looking at the problem at a different perspective, each given different weights for different schema elements
- Predictions combined by a meta-learner

25

Issues

- The experiments show that performance is insensitive to the number of data listings available. Is it good or bad?
- How can we extend it to handle matching any two schemas?

26