Multi-view Exploratory Learning for AKBC Problems

Bhavana Dalvi
School of Computer Science
Carnegie Mellon University
bbd@cs.cmu.edu

William W. Cohen
School of Computer Science
Carnegie Mellon University
wcohen@cs.cmu.edu

Abstract

In this paper, we argue that many Automatic Knowledge Base Construction (AKBC) tasks which have previously been addressed separately can be viewed as instances of single abstract problem: multiview semi-supervised learning with an incomplete class hierarchy. We also present a general EM framework for solving this abstract task, and summarize past work on various special cases of multiview semi-supervised exploratory learning.

1 Introduction

Traditional semi-supervised learning (SSL) techniques consider the missing labels of unlabeled datapoints as latent/unobserved variables, and model these variables, and the parameters of the model, using techniques like Expectation Maximization (EM). We consider two extensions to traditional SSL methods which make it more suitable for AKBC tasks. First, we consider jointly assigning multiple labels to each instance, with a flexible scheme for encoding constraints between assigned labels: this makes it possible, for instance, to assign labels for multiple levels from a hierarchy. Second, we account for another type of latent variable, in the form of unobserved classes. In open-domain web-scale information extraction problems, it is an unrealistic assumption that the class ontology or topic hierarchy we are using is complete. Our proposed framework combines structural search for the best class hierarchy with SSL, reducing the semantic draft associated with erroneously grouping unanticipated classes with expected classes.

Together, these extensions allow a single framework to handle a large number of AKBC tasks, including macro-reading, micro-reading, multi-view macro- or micro-reading, alignment of KBs to wikipedia or on-line glossaries, and ontology extension. Below we will summarize how these tasks can be modeled in this framework, and then introduce a general framework for solving the multiview semi-supervised exploratory learning problem.

Following are various AKBC tasks that the Exploratory Learning framework can help with.

- **Macro-reading:** This task refers to semi-supervised classification of noun-phrases into a big taxonomy of classes, using distributional representation of noun-phrases. We proposed an exploratory learning method [11] for macro-reading that could reduce the semantic drift of seeded classes hence helping the traditional semi-supervised learning objectives.
- **Micro-reading:** Micro-reading differs from macro-reading in the sense that instead of using collective distributional features of a noun-phrase, we are trying to disambiguate an occurrence of noun-phrase w.r.t. the local context within a sentence or paragraph. We have applied our exploratory learning technique for clustering NIL entities in the KBP entity discovery and linking (EDL) task [7]. We are working on using such hierarchical semi-supervised learning techniques for the task of word sense disambiguation by considering occurrences of all monosemous words as training data and polysemous word occurrences as test data. In these experiments, we use the WordNet synset hierarchy to deduce ontological constraints.

- **Multi-View Macro(/Micro)-reading:** This task refers to what the "Macro-reading" does, plus collecting signals from multiple data views. E.g. NELL [4] proposed an information extraction system that classifies noun-phrases into a class hierarchy using clues from 2 different sources: text patterns occurring in sentences and semi-structured data in the form of lists and tables. We proposed a multi-view semi-supervised learning method [9] for this task. An example of "Multi-view micro reading" is a word sense disambiguation task where there are multiple data views of word synsets. E.g., resources like WordNet, Wiktionary contain for each word sense, a gloss and a set of example usages. One can use the glosses and example usages as multiple views to train multi-view exploratory learning models.
- **Alignment of online glossaries to an existing Knowledge Base:** This task can also be viewed as a gloss finding task for an existing gloss-free knowledge base. We proposed a hierarchical semi-supervised learning method, named GLOFIN [12] for this task. GLOFIN does entity sense disambiguation by classifying a mention of an entity from a potential gloss, into knowledge base categories while incorporating the ontological constraints. A KB with glosses has been shown to be helpful for the task of entity recognition and disambiguation from search queries [13, 5].
- **Ontology extension:** Most of the concept or topic hierarchies available are incomplete to represent entities present on the Web. This task refers to the problem of discovering new classes that can be added to existing concept hierarchies to make them representative of the real world data. Many techniques have been proposed [15, 18, 17] with similar goals, however they are applicable in limited settings. We proposed a unified hierarchical exploratory learning technique OptDAC-ExploreEM that can populate known seeded classes in an ontology along with extending it with newly discovered concepts while taking care of ontological constraints [10, 8].
- **Relation learning:** All the tasks mentioned above are about discovering new concepts or new instances of existing concepts, however an important challenge for existing knowledge bases is the sparsity of relations. We hypothesize that along with concept learning, our proposed exploratory learning techniques can also be applied to the problem of relation learning.

2 Generic Exploratory Learning Framework

Expectation Maximization (EM) is a popular technique for estimating parameters for a semi-supervised learning task. In a typical EM setting, the M-step finds the best parameters θ to fit the data X, and the E-step probabilistically labels the unlabeled datapoints with a distribution over the known classes $C_1, C_2, \ldots C_k$. In some variants of EM, including the ones we consider here, a "hard" assignment is made to classes instead, an approach named *classification EM* [6]. Our exploratory version of EM differs in that it can introduce new classes $C_{k+1} \ldots C_m$ during the E-step. Algorithm 1 describes the proposed generic Exploratory Learning framework. It can handle possibly multiple data views, and possibly incomplete class hierarchies. Since the EM algorithm can be used for both classification and clustering tasks, we will use the terms "class" and "cluster" interchangeably.

Handling Multiple Data Views

This step is done in Line 11 of Algorithm 1. Here, we are considering a scenario where the information about datapoints is coming from multiple views. For instance consider the Web document classification task with 2 data views, the text within the document and the anchor texts of its inbound hyper-links. Similarly, in an information extraction task to populate a Knowledge Base (KB) like NELL [4], each noun-phrase to be classified has different feature sets or data views; e.g., surrounding text contexts, HTML table occurrences, morphological features etc.

We performed extensive comparison of different ways of combining information from multiple views [9] on nine different multi-view datasets. We found that simple techniques like taking summation of scores generated by each view, or concatenating feature sets from multiple views work reasonably well, and act as hard to beat baselines. Our proposed MAXAGREE method [9] improved performance over above mentioned baselines, by decoupling the label assignments in each view and adding a soft constraint for them being same across views. Further the hierarchical extension of MAXAGREE [9] gave state of the art performance on entity classification task for NELL KB with very few seeds.

Algorithm 1 Generic Exploratory Learning Algorithm

1: **function Exploratory EM** (X^l, Y^l, X^u, Z_k): θ_{k+m}, Z_{k+m}, Y^u
2: **Input**: X^l labeled data points (In case of v data views: $X^{l(1)} \dots X^{l(v)}$); Y^l labels of X^l; X^u unlabeled data points (In case of v data views: $X^{u(1)} \dots X^{u(v)}$); Z^0 manually input constraints on k seed classes (subclass-superclass or mutual-exclusion kind); P_{new} probability of creating a new class
3: **Output:** $\{\theta_1, \dots, \theta_{k+m}\}$ parameters for k seed and m newly added classes (Class j in case of v data views: $\theta_j^{(1)} \dots \theta_j^{(v)}$); Z_{k+m} Set of class constraints between $k+m$ classes; Y^u labels for X^u
4: h = height of ontology that is part of Z_k
5: Initialize classifiers θ_j for class C_j using seeds provided for C_j
6: **while** class assignments AND #classes not converged **do**
7: k_{old} = #classes before the E step
8: Log-likelihood *BaseLL* $= logP(X|\theta^{(t)}_{k_{old}}, Z^{(t)}_{k_{old}})$
{E step: (Iteration t) Classify each datapoint at each level}
9: **for** i=1 to $|X|$ **do**
10: Find $P(C_j|X_i)$ for all classes C_j
{If there are v data views then combine scores from all views }
11: $P(C_j|X_i)$ = **ComputeCombinedMultiViewScore**($X_i^{(1)} \dots X_i^{(v)}, \theta_j^{(1)} \dots \theta_j^{(v)}$)
12: **if** check if **NewClassCreationCriterion**($P(C_j|X_i)$, X_i) is satisfied **then**
13: $Z^{(t)}$ = **UpdateConstraints**($\{X^l \cup X^u\}$, $\{Y^l \cup Y^u\}$, $Z^{(t)}$)
14: **end if**
15: $Y_i^{(t)}$ = **ConsistentAssignment**($P(C_j|X_i)$, h, $Z^{(t)}$)
16: **end for**
17: k_{new} = #classes after the E step
18: Log-likelihood *ExploreLL* $= logP(X|\theta^{(t)}_{k_{new}}, Z^{(t)}_{k_{new}})$
{M step: Recompute model parameters based on $Y^{(t)}$ }
19: **if ModelSelectionCriterion**(k_{old}, *BaseLL*, k_{new}, *ExploreLL*) selects exploratory model **then**
{Adopt the new model with k_{new} classes}
20: $\theta^{(t+1)}_{k_{new}} = argmax_\theta L(X^l, Y^l, X^u, Y^{u(t)}|\theta^{(t)}_{k_{new}}, Z^{(t)}_{k_{new}})$; $Z^{(t+1)}$ = $Z^{(t)}_{k_{new}}$
21: **else**
{Keep the old model with k_{old} classes}
22: $\theta^{(t+1)}_{k_{old}} = argmax_\theta L(X^l, Y^l, X^u, Y^{u(t)}|\theta^{(t)}_{k_{old}}, Z^{(t)}_{k_{old}})$; $Z^{(t+1)}$ = $Z^{(t)}_{k_{old}}$
23: **end if**
24: **end while**
25: **end function**

Dynamically Introducing New Classes

This step is done in Line 12 of Algorithm 1. Here, we make a locally optimal decision about whether to create a new class/cluster for a particular datapoint. We have access to labels of all the documents already labeled in the current E step, and the probability of this particular datapoint belonging to any of the existing classes/clusters.

Our intuition is that, for a datapoint x with posterior class probabilities for k existing classes being $P(C_j|x)$, a new class should be introduced to hold x if $P(C_j|x)$ is close to uniform. We have proposed two heuristic based criteria to achieve this [11]. In the JS criterion, we require that Jensen-Shannon divergence between the posterior class distribution $P(C_j|x)$ to the uniform distribution be less than $\frac{1}{k}$. The MinMax criterion is a somewhat simpler approximation to this intuition: a new cluster is introduced if the maximum probability is no more than twice the minimum probability. One can also train a classifier to predict when to create a new class based on the training and validation sets. The Jensen-Shannon divergence, class skew ratios etc. can act as features of this classifier.

Handling Ontological Constraints

This step is done in Lines 13 and 15 of Algorithm 1. Consider a toy example of ontological class constraints in Figure 1. Here, we can see two kinds of class constraints imposed by the ontology. Following are example constraints: (1) The "Subset" constraint between "Fruit" and "Food" categories suggests that if a datapoint is classified as "Fruit", then it should also be classified as "Food". (2) The

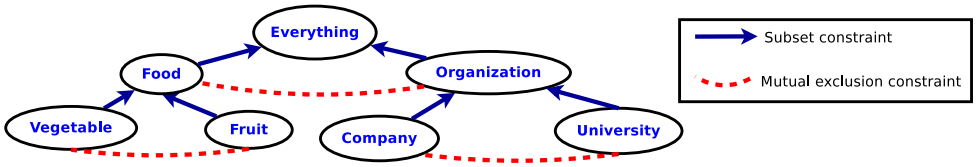

Figure 1: An example of ontological class constraints.

"Mutual Exclusion" constraint between "Food" and "Organization" says if a datapoint is classified as "Food", then it should not be classified as "Organization", and vice versa. Let $\{C_1, \ldots C_K\}$ be the Knowledge Base (KB) categories. Let *Subset* be the set of all subset or inclusion constraints, and *Mutex* be the set of all mutual exclusion constraints. In other words, *Subset* = $\{\langle i, k\rangle : C_i \subseteq C_k\}$ and *Mutex* = $\{(i, k) : C_i \cap C_k = \phi\}$. The class constraints referred to as Z_k in Algorithm 1 can be defined as $Z_k = \{$*Subset*, *Mutex*$\}$.

We define $y_{ji} \in \{0, 1\}$ as an indicator variable specifying whether x_i belongs to C_j. The category membership probabilities $\{P(C_j|x_i)\}$ estimated in the E step, are used to compute the indicator variables $\{y_{ji}, \forall\, 1 \leq i \leq N,\ 1 \leq j \leq K\}$. Our GLOFIN algorithm [12], proposed a Mixed-Integer Program (MIP) per datapoint x_i to estimate y_{ji}'s. This MIP takes the scores $\{P(C_j|x_i)\}$, and class constraints Z_k as input and produces a consistent bit vector of labels y_{ji}'s as output. Finding such consistent label assignments is referred as "ConsistentAssignment" in Algorithm 1 Line 15.

Further, when we add a new class C_{new} to the ontology, class constraints associated with C_{new} are updated in the "UpdateConstraints" step in Algorithm 1 Line 13. E.g. consider we add a new class node as a child of 'Food' in the toy ontology in Figure 1, then we will update class constraints Z to include a subset constraint: C_{new} is subset of 'Food', and mutual exclusion constraints between C_{new} and its siblings 'Vegetable' and 'Fruit'.

Trade-off between Fitting the Available Data and Creating Too Many Classes

Note that adding a new class always improves log-likelihood of data, however creating too many new classes would result in over-fitting (one class per datapoint always result in the best likelihood). Hence we optimize the penalized log-likelihood of data by penalizing the model complexity, by applying model selection techniques (Lines 19-23 of Algorithm 1). At the end of every E step, we evaluate two models, one with and one without adding extra classes. These two models are scored using a model selection criterion, and the model with best penalized data likelihood score is selected in each iteration. For model penalties, we tried multiple well known criteria like BIC, AIC and AICc. The extended Akaike information criterion (AICc) [3] suited best for our experiments since our entity classification datasets have large number of features and small number of data points. Further we discard singleton clusters at the end of each E step, avoiding a new cluster per outlier datapoint. [11] discusses the similarity of our Exploratory EM algorithm with the Structural EM algorithm proposed by Friedman [14], and hence its convergence to the local maxima for penalized log likelihood.

Incorporating Different Document Representations

Computations done in Lines 10, 20, and 22 of Algorithm 1 depend on a particular choice of the document representation. A variety of techniques may be used for computing probabilities $P(C_j|x_i; \theta_j)$, where θ_j is the current estimate of model parameters for category C_j. We briefly describe one such choice here: the semi-supervised multinomial Naive Bayes [16]. In this model $P(C_j|x_i) \propto P(x_i|C_j) * P(C_j)$, for each unlabeled datapoint x_i. The probability $P(x_i|C_j)$ is estimated by treating each feature in x_i as an independent draw from a class-specific multinomial. In the noun-phrase classification task, the features could be surrounding text patterns, the number of outcomes of the multinomial being the vocabulary size.

Another possible variant is the seeded spherical K-Means algorithm proposed by Basu and Mooney [2], also a kind of semi-supervised EM algorithm. Third possible variant is seeded von Mises-Fisher proposed by Banerjee et al. [1], a generative mixture model approach to clustering datapoints based

on von Mises-Fisher distribution, defined for data distributed on the unit hypersphere (L_2 norm equals 1). We have experimented with all three document representations and showed improvements of exploratory learning over the usual semi-supervised baseline [11]. Note that our algorithm can be used with any other document representation that fits the generative EM framework.

Acknowledgments:

This work is supported by the Google PhD fellowship in Information Extraction and Google research grant. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Google.

3 Conclusions and Future Work

To summarize, we have found that exploratory learning helps reduce semantic drift on known seeded classes as well as provides a single method that solves a continuum of tasks from supervised learning to clustering. In both flat and hierarchical classification/clustering settings, we observed that running an algorithm in exploratory setting gave comparable or better seed class F1 score when compared to the traditional semi-supervised counterpart [11, 10, 8]. Further, exploratory learning is more powerful in conjunction with multiple data views and class hierarchy which can be imposed as soft constraints on the label vectors assigned to a datapoint. We also found that considering ontological constraints while inferring labels improves overall performance compared to flat classification for the the entity classification and gloss finding tasks [10, 12, 8]. Code for the flat exploratory K-Means algorithm and some datasets we created during this research can be downloaded from `http://www.cs.cmu.edu/~bbd/exploratory_learning/`.

Apart from reducing the semantic drift of seeded classes, another important advantage of the exploratory learning method is to discover multiple clusters of datapoints that did not fit any of the classes given to the algorithm. These are potentially new concepts that can be added to an existing knowledge base. E.g. , in our experiments with WebSets table corpus and NELL KB, we found that with some manual inspection, exploratory learning can add good quality concepts like "music notes", "dental procedures" etc., that were missing in the NELL KB.

Our exploratory learning framework is not tied to any particular document representation, choice of features or task. Hence one obvious future research direction lies in applying this idea to tasks other than concept extraction. E.g. Relation extraction is a very challenging task, and many ontologies like NELL that are rich in terms of coverage of concepts, have very sparse coverage in terms of relations between entities. Another relevant task is the context based mention disambiguation using WordNet as an ontology of synsets. We hope that the Exploratory learning technique can be extended for relation extraction and word sense disambiguation tasks as well.

References

[1] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von mises-fisher distributions. In *JMLR*, 2005.

[2] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *ICML*, 2002.

[3] K. P. Burnham and D. R. Anderson. Multimodel inference understanding aic and bic in model selection. *Sociological methods & research*, 2004.

[4] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *WSDM*, 2010.

[5] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. ERD 2014: Entity recognition and disambiguation challenge. *SIGIR Forum*, 2014.

[6] G. Celeux and G. Govaert. A classification em algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 1992.

[7] W. W. Cohen. Random-walk approaches to entity discovery and linking. In *KBP Entity Linking Task*, 2014.

[8] B. Dalvi and W. W. Cohen. Hierarchical semi-supervised classification with incomplete class hierarchies. In *under submission*, 2014.

[9] B. Dalvi and W. W. Cohen. Multi-view hierarchical semi-supervised learning by optimal assignment of sets of labels to instances. In *in preparation*, 2014.

[10] B. Dalvi, W. W. Cohen, and J. Callan. Classifying entities into an incomplete ontology. In *AKBC*, 2013.

[11] B. Dalvi, W. W. Cohen, and J. Callan. Exploratory learning. In *ECML*, 2013.

[12] B. Dalvi, E. Minkov, P. P. Talukdar, and W. W. Cohen. Automatic gloss finding for a knowledge base using ontological constraints. In *WSDM*, 2015.

[13] B. Dalvi, C. Xiong, and J. Callan. A language modeling approach to entity recognition and disambiguation for search queries. In *ERD, Entity Recognition and Disambiguation Challenge at SIGIR*, 2014.

[14] N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural EM algorithm for phylogenetic inference. *Journal of Computational Biology*, 2002.

[15] T. P. Mohamed, E. R. Hruschka Jr, and T. M. Mitchell. Discovering relations between noun categories. In *EMNLP'11*.

[16] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 2000.

[17] A. Pal, N. N. Dalvi, and K. Bellare. Discovering hierarchical structure for sources and entities. In *AAAI*, 2013.

[18] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *ACL*, 2006.