EM based clustering for learning verb and noun classes

Tejaswini Deoskar

1 Introduction

In class, we noticed an interesting linguistic fact about verbs: verbs seem to fall into various semantic classes that show different syntactic alternations. For instance, verbs in the BREAK class can have the following alternation, but not the verbs in the HIT class

John broke the window.	John hit the window.
The window broke.	*The window hit.

We will use the EM algorithm to discover these classes from unlabeled data. This a fairly straightforward application of EM (EM has been used successfully for text classification, for instance), as compared to using EM to learn more structured models like PCFG/POS induction (where EM has proved to not be very successful). Although technically simple, here we explore its use for a fairly non-standard and semantically interesting task.

The main reference for the project is the paper "Inducing a Semantically Annotated Lexicon via EM-Based Clustering", a paper that was ahead of its time and has largely been overlooked since, despite having some interesting ideas on inducing semantic representations [1] We will try to replicate the results in this paper.

You might also want to look over an EM tutorial: A good one for the classification case is by Michael Collins `http://www.cs.columbia.edu/~mcollins/em.pdf`.

There are two main tasks:

- The first task is to learn classes of **verbs**. The training data for this task consists of pairs of verbs and nouns (v, n) that are extracted from a large corpus. The corpus is first parsed, and for every verb, the head noun of the subject NP or object NP is extracted. The verb further

[1] These are *categorical* representations

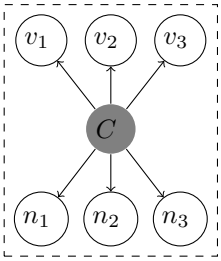

Figure 1: The Graphical Model

indicates whether the noun was in the subject/object position. Thus the data is something like:

V.in.s N
V.tr.s N
V.tr.o N
where *V.in.s* N indicates that N is the subject of an intransitive verb. *V.tr.s* indicates subject of a transitive verb, and *V.tr.o* object of a transitive verb (I am using a more mnemonic notation than in the paper)

- The second part consists of using the learnt classes of verbs (and keeping them fixed), and further inducing classes on *nouns* that are either in the subject or object positions of these verbs.

2 Part 1 : verb classes

This is a simple model (represented in graphical terms in Fig 1) :
$p(c, v, n) = p(c) \cdot p(v|c) \cdot p(n|c)$
Each of these are categorical/discrete distributions. Thus,

$$P(C) = Cat(\sigma_1 \dots \sigma_k)$$

$$P(V) = Cat(\phi_1 \dots \phi_{v_n})$$

$$P(N) = Cat(\lambda_1 \dots \lambda_{n_n})$$

where k is the number of classes, v_n is the number of verbs, i.e. $|V|$, , n_n is the number of nouns, $|N|$.
Parameters are:

$$\theta = \{\sigma_c, \phi_{c,v}, \lambda_{c,n} : \forall c, \forall v, \forall n\}$$

$P(N|C)$ params would be a table of dimensions $kX|N|$, where each row should sum to 1, and similarly for $P(V|C)$.

	n_1	n_2	...	$n_{|N|}$
c_1				
c_2				
.				
.				
c_k				

$$P_\theta(c, v, n) = P_\theta(c)P_\theta(v|c)P_\theta(n|c)$$

$$\begin{aligned} P_\theta(c|v, n) &= \frac{P_\theta(c, v, n)}{P_\theta(v, n)} \\ &= \frac{P_\theta(c)P_\theta(v|c)P_\theta(n|c)}{\sum_{c'} P_\theta(c')P_\theta(v|c')P_\theta(n|c')} \end{aligned}$$

For evaluation of this part, we will focus on the evalution in section 3.1 of the paper. Sec 3.2 is optional.

3 Part 2 : Subject/Object classes

In this part, the first goal is to induce the latent classes for subjects of a fixed intransitive verb.

This is how the probability of an arbitrary subject is calculated:

$$P(n) = \sum_{c \in C} p(c, n) = \sum_{c \in C} p(c)PLC(n|c)$$

$PLC(.)$ is a latent class model for verb-noun pairs.

$$M(\theta_c) = \frac{\sum_{n \in N} f(n)p_\theta(c|n)}{\sum_{n \in N} f(n)}$$

$$\theta = \{\theta_c | c \in C\}$$

A similar approach can be applied to induce latent semantic annotation for transitive verb frames.

$$P(n_1, n_2) = \sum_{c_1, c_2 \in C} p(c_1, c_2, n_1, n_2) = \sum_{c_1, c_2 \in C} p(c_1, c_2) PLC(n_1|c_1) PLC(n_2|c_2)$$

$$M(\theta_{c1c2}) = \frac{\sum_{n_1, n_2 \in N} f(n_1, n_2) p_\theta(c_1, c_2|n_1, n_2)}{\sum_{n_1, n_2 \in N} f(n_1, n_2)}$$

$$\theta = \{\theta_{c_1, c_2} | c_1, c_2 \in C\}$$