Two Models for Word Learning

Su Wang

DEPARTMENT OF LINGUISTICS
UNIVERSITY OF TEXAS AT AUSTIN

September 7, 2016

1 Model I

1.1 Definitions

Let u be an unknown noun, w a known noun. Let P be the set of all properties of nouns in the world, then

- [PROPERTIES OF WORD]: The properties of a word w is described by a Multinomial Distribution

  $$w_{prop}(X) \sim Multinomial(X \mid \Theta) \tag{1}$$

  over P, where $x \in X$ is the production frequency of property $prop \in P$ in the sample X drawn with the parameters Θ.

- [DISTRIBUTION OF PROPERIES]: The distribution of properties over P is

  $$\Theta \sim Dir(\alpha) \tag{2}$$

  i.e. a Dirichlet Distribution, where α is the parameter indicating the weights of properties.

- [INFORMATION UNIT]: A predicate v observed in a sentence s where the word w is one of its core arguments (i.e. subject or object) is an information unit to w such that v is associated with two sets of parameters: α_{subj} and α_{obj}, each is a vector over P where $\alpha_{prop}, prop \in P$ is the number of times property $prop$ is observed to appear in the core argument position (subj/obj) of v.

1.2 Model Description

1.2.1 Training

Let N be the set of norms from McRae et al. (2005), and V is the set of all the target predicates (i.e. for which N are the set of core arguments) from Brown

Corpus. To distinguishing the set of subject-related and object-related norms: $N = N_{subj} \cup N_{obj}$. For any single predicate v, use v_{subj} and v_{obj} to denote the set of norms that appear in the subj/obj position of v.

The training objective is that, for each $v \in V$, learn its associated property weights α_{subj} and α_{obj}.

For each $v \in V$, find its v_{subj} and v_{obj}, then find the properties related to the norms in v_{subj} and v_{obj} and update the parameters α_{subj} and α_{obj} of v, which are initialized with a vector of the near-zero number 10^{-20}, by adding to them the production frequency of the properties (as per McRae et al. 2005).

1.2.2 Learning & Updating

Let S be a set of sentences in which an unknown word w is observed, we learn the property weights β for the properties of w as follows:

1. Initialize β as a vector of the near-zero number 10^{-20},
2. For each v of w in sentence $s \in S$, use the property weights α (subj/obj) to updat β by $\beta := \beta + \alpha$,
3. Obtain the parameter Θ as the distribution over properties by sampling from $Dir(\beta)$ k times and take the average,
4. Sample l times from $Multinomial(\Theta)$, and take the average X, where $x \in X$ is the inferred frequency of corresponding property $prop \in P$.

1.3 Demo: Learning *Alligator*

See Fig. 1-4.

2 Model II

2.1 Definitions

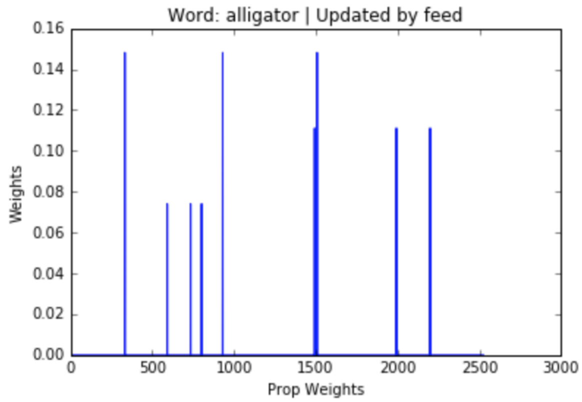
The definition of PROPERTIES OF WORD is the same as in Model I, and the rest two definitions are now as follows:

- [DISTRIBUTION OF PROPERTIES]:

$$\Theta \sim Dir(\alpha), \alpha_{prop} = p(prop \mid v) = \sum_{topic \in T} p(prop \mid topic) p(topic \mid v) \quad (3)$$

  where T is the set of all topics learned from Brown, interpreted roughly as "property clusters".

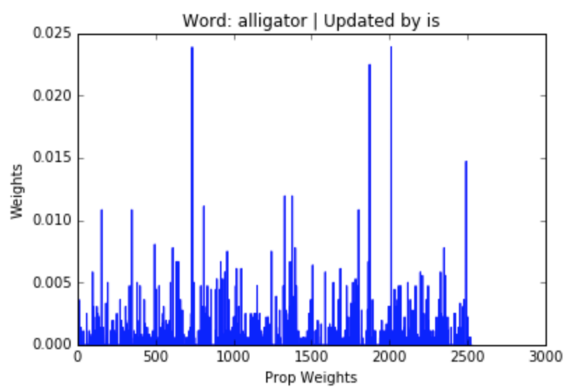
- [INFORMATION UNIT]: A predicate v observed in a sentence s where the word w is one of its core arguments (i.e. subject or object) is an information unit to w such that v is associated with two sets of parameters: α_{subj} and α_{obj}, each is a vector over P where α_{prop} the probability of property $prop \in P$ (computed as in (3)), used as a corresponding property weight.




```
Average Entropy:  2.15946923583

1th Property: an_animal (prob=0.148148%,idx=931)
2th Property: has_4_legs (prob=0.148148%,idx=1508)
3th Property: has_a_tail (prob=0.148148%,idx=333)
4th Property: beh_-_eats (prob=0.111111%,idx=1993)
5th Property: is_edible (prob=0.111111%,idx=1493)
```

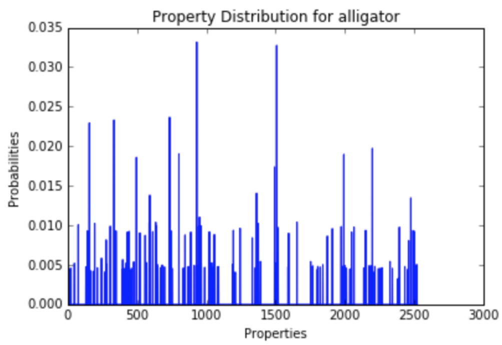
Figure 1: Single Update: Feed




```
Average Entropy:  5.699702862

1th Property: made_of_wood (prob=0.023876%,idx=2012)
2th Property: is_large (prob=0.023876%,idx=735)
3th Property: made_of_metal (prob=0.022488%,idx=1874)
4th Property: is_small (prob=0.014714%,idx=2493)
5th Property: has_doors (prob=0.011938%,idx=1377)
```

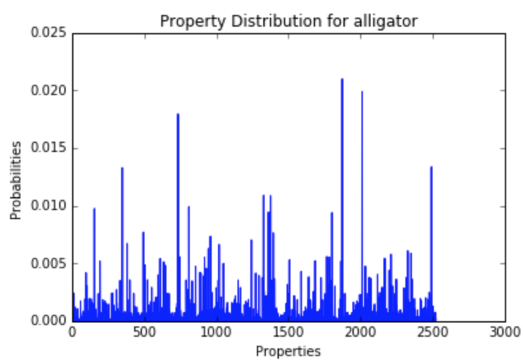
Figure 2: Single Update: Is




```
Average Entropy: 4.680241

1th Property: has_4_legs (wgt=7.000000,idx=1508)
2th Property: an_animal (wgt=7.000000,idx=931)
3th Property: used_for_transportation (wgt=5.000000,idx=155)
4th Property: is_large (wgt=5.000000,idx=735)
5th Property: has_a_tail (wgt=5.000000,idx=333)
```

Figure 3: Multi Update: Feed, Catch, Roam




```
Average Entropy: 6.167238

1th Property: made_of_metal (wgt=122.000000,idx=1874)
2th Property: made_of_wood (wgt=115.000000,idx=2012)
3th Property: is_large (wgt=106.000000,idx=735)
4th Property: is_small (wgt=79.000000,idx=2493)
5th Property: different_colours (wgt=75.000000,idx=348)
```

Figure 4: Multi Update: Is, Have, Get

2.2 Model Description

2.2.1 Training

First make predicate-argument (subject/object) cooccurrence matrices $S_{|V|\times|N_{subj}|}$ and then $O_{|V|\times|N_{obj}|}$, and derive predicate-property cooccurrence matrices $S'_{|V|\times|P_{subj}|}$ and $O'_{|V|\times|P_{obj}|}$ by looking up v_{subj} and v_{obj} for each $v \in V$ from McRae et al. (2005). Note that the cells in the predicate-property cooccurrence matrices are the production frequencies.

Use the predicate-property matrices to create pseudo-documents (as per Dinu & Lapata 2010): each row corresponds to a predicate v, and the corresponding pseudo-document is generated using the frequency of properties in the cells – if $p =$ `an_animal` appears 11 times in a core argument position of $v =$ `pet`, then add 11 `an_animal` as words to the document.

Finally make a predicate-topic model using the pseudo-documents, from which we obatin two distributions $p(\mathit{prop} \mid \mathit{topic})$ and $p(\mathit{topic} \mid \mathit{predicate})$.

2.2.2 Learning & Updating

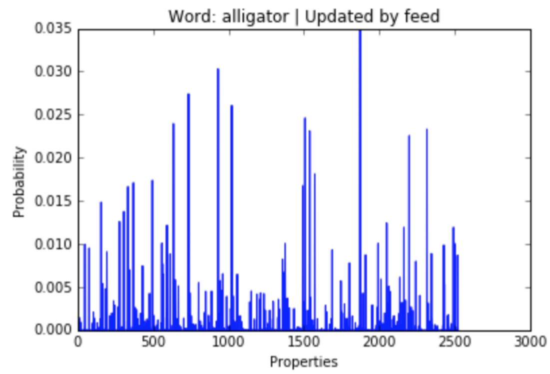
Let S be a set of sentences in which an unknown word w is observed, we learn the property weights β for the properties of w as follows:

1. Initialize β as a vector of the near-zero number 10^{-20},
2. For each v of w in sentence $s \in S$, compute property weights α, a distribution over P, by $p(\mathit{prop} \mid v) = \sum_{\mathit{topic} \in T} p(\mathit{prop} \mid \mathit{topic}) p(\mathit{topic} \mid v), \forall \mathit{prop} \in P$, then update β by $\beta := \beta + \lambda \cdot \alpha$[1]
3. Obtain the parameter Θ as the distribution over properties by sampling from $\mathit{Dir}(\beta)$ k times and take the average,
4. Sample l times from $\mathit{Multinomial}(\Theta)$, and take the average X, where $x \in X$ is the inferred frequency of corresponding property $\mathit{prop} \in P$.

2.3 Demo: Learning *Alligator*

See Fig. 5-8.

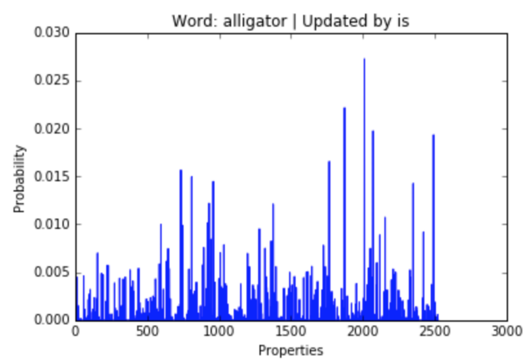
[1] λ is the hyperparameter that models the *learning speed* of the agent, s.t. when λ is large, the spikes in α get amplified.




```
4.94428473795

1th Property: made_of_metal (prob=0.034918%,idx=1874)
2th Property: an_animal (prob=0.030312%,idx=931)
3th Property: is_large (prob=0.027420%,idx=735)
4th Property: used_by_riding (prob=0.026073%,idx=1022)
5th Property: has_4_legs (prob=0.024619%,idx=1508)
```

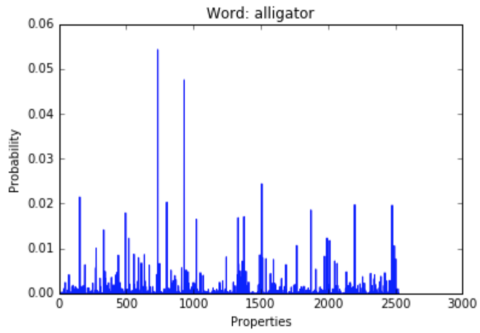
Figure 5: Single Update: Feed




```
5.43274610466

1th Property: made_of_wood (prob=0.027258%,idx=2012)
2th Property: made_of_metal (prob=0.022159%,idx=1874)
3th Property: used_for_living_in (prob=0.019740%,idx=2072)
4th Property: is_small (prob=0.019333%,idx=2493)
5th Property: requires_drivers (prob=0.016551%,idx=1767)
```

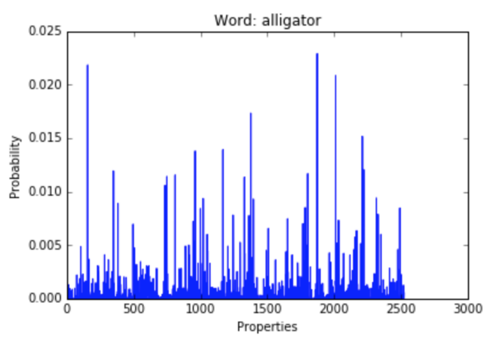
Figure 6: Single Update: Is




```
5.21266372159

1th Property: is_large (prob=0.054280%,idx=735)
2th Property: an_animal (prob=0.047534%,idx=931)
3th Property: has_4_legs (prob=0.024407%,idx=1508)
4th Property: used_for_transportation (prob=0.021455%,idx=155)
5th Property: a_mammal (prob=0.020295%,idx=802)
```

Figure 7: Single Update: Feed, Catch, Roam




```
5.80263256205

1th Property: made_of_metal (prob=0.022895%,idx=1874)
2th Property: used_for_transportation (prob=0.021831%,idx=155)
3th Property: made_of_wood (prob=0.020837%,idx=2012)
4th Property: has_doors (prob=0.017335%,idx=1377)
5th Property: has_4_wheels (prob=0.015165%,idx=2212)
```

Figure 8: Single Update: Is, Have, Get