Compositionality Detection with Latent Dirichlet Allocation

Reuben Cohn-Gordon: 229 Class Project

2019

The task

- Many phrases have meanings not entirely predictable from the meanings of their parts (*non-compositionality*)
  - *big horse* (usually compositional)
  - *high horse* (usually non-compositional)
  - *cold feet* (could be either)
- **Goal 1:** Using statistical methods, can we determine when phrases are compositional in a given context? (non-compositional phrase *detection*)
- **Goal 2:** For a given phrase, can we predict across a corpus whether it tends to be compositional or not? (non-compositional phrase *induction*)
- We propose an **unsupervised** approach to both these tasks, using a probabilistic generative model

Building on a simpler task: word sense detection and induction

- To address goals 1 and 2, we extend an existing approach used in the simpler but analogous tasks of single word *sense induction and detection*
- Word sense detection is the task of **classifying** the sense of a word (e.g. distinguishing between the senses of *bug* in:
  - The bug crawled into the room.
  - The bug broke the code.
- **An unsupervised approach**: Latent Dirichlet Allocation (LDA)
- Step 1: choose a word and number of senses
- Step 2: create a corpus of sentences containing the word
- Perform LDA, with the sentences as documents, and the word senses as topics

p: the phrase in question. E.g. *high time* w_1: first word of phrase w_2: second word

D_x : corpus of sentences in which word or phrase x appears

K: number of senses. Assume shared, fixed number for word 1, word 2, and phrase

Latent Dirichlet Allocation

for i in range(K) **do**
  Choose $\phi^i_{w_1} \sim$ Dirichlet(β)
end for
for sentence d in corpus D_{w_1} **do**
  Choose $\theta^d_{w_1} \sim$ Dirichlet(α)
  for position j in d **do**
    Choose a sense index $z_j \sim$ Multinomial($\theta^d_{w_1}$)
    Choose a word $x^j_d \sim$ Multinomial($\phi^{z_j}_{w_1}$)
  end for
end for

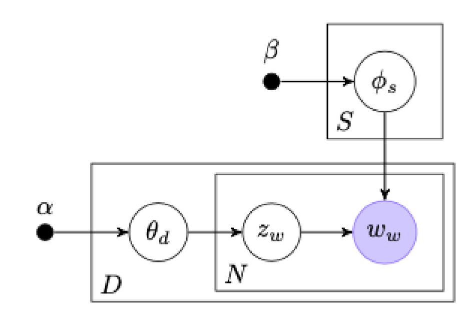

Figure: LDA applied to single word sense detection. Pseudocode and plate diagram

Compositional Latent Dirichlet Allocation

Run LDA on D_{w_1} and D_{w_2} to obtain $\{\phi^i_{w_1}\}$ and $\{\phi^i_{w_2}\}$
Choose $\kappa \sim$ Beta(γ, γ)
Choose $\omega \sim$ Beta(δ, δ)
for i in range(K) **do**
  Choose $\phi^i_p \sim$ Dirichlet(β)
end for
for sentence d in corpus D_p **do**
  Choose $\lambda_d \sim$ Bernoulli(κ)
  if λ **then**
    Choose $\theta^d_{w_1} \sim$ Dirichlet(α)
    Choose $\theta^d_{w_2} \sim$ Dirichlet(α)
    for position j in d **do**
      Choose $w \sim$ Bernoulli(ω)
      Choose a sense index $z_j \sim$ Multinomial(θ^d_w)
      Choose a word $x^j_d \sim$ Multinomial($\phi^{z_j}_w$)
    end for
  else
    Choose $\theta^d_p \sim$ Dirichlet(α)
    for position j in d **do**
      Choose a sense index $z_j \sim$ Multinomial(θ^d_p)
      Choose a word $x^j_d \sim$ Multinomial($\phi^{z_j}_p$)
    end for
  end if
end for

Corpus Collection

- Training and evaluating the model requires a corpus of common adjective-noun non-compositional phrases, like *black box* and *sour grapes*
- This is obtained by constituency parsing the British National Corpus and the Corpus of Contemporary American English and using tree-regexes to match adjective-noun phrases
- This approach ensures that the expressions are of the correct syntactic type.

Implementation

- To implement our model, we use newly developed *probabilistic programming language* Gen, embedded in Julia.
- Probabilistic programming languages allow for probabilistic primitives
- Gen in particular allows for easy specification of custom MCMC inference algorithms.

Further work

- Improved inference algorithm.
  Possibilities: collapsed Gibbs sampling, online Variational Bayes, special purpose algorithm
- Extending the approach to a non-parametric model.
  Advantage: avoids the need to prespecify a set of senses.

Audio

`https://www.youtube.com/watch?v=kAbLaVXeOpo`