Distributed MAP Inference for Undirected Graphical Models

Sameer Singh†, Amarnag Subramanya§, Fernando Pereira§, Andrew McCallum†

† University of Massachusetts, Amherst MA § Google Research, Mountain View CA

sameer@cs.umass.edu, asubram@google.com, pereira@google.com, mccallum@cs.umass.edu

Google

IESL Information Extraction and Synthesis Laboratory

Motivation

There have been recent advances in approximate inference and learning methods:
- FACTORIE [MCallum et al NIPS 2009]
- Lazy Inference [Poon et al AAAI 2008]
- Dual Decomposition [Rush et al EMNLP 2010]
- LP relaxation [Martins et al EMNLP 2010]

These methods allow inference over models with global dependencies:
- Coreference Models [Culotta et al NAACL 2007]
- Relation Extraction [Riedel et al EMNLP 2010]
- Joint Inference of Multiple Tasks [Singh et al ECML 2009; Finkel & Manning NAACL 2009]

However, without parallelization, the scalability of these methods is severely limited.
Our Contribution: Distribute MCMC-based MAP Inference using Map-Reduce

Related Work:
- GraphLab [Low et al UAI 2010]
- Splashing [Gonzalez et al UAI 2009]
- Topic Models [Smola & Narayanamurthy VLDB 2010; Asuncion et al NIPS 2009]

Our method allows ease in specifying structure, distribution strategy and proposal function, which enables faster convergence.

Factor Graphs

Undirected graphical models that represent distribution over assignments y of the unobserved variables (Y) given observed variables (x) using factors (Ψ):

$$p(y|x) \propto \exp \sum_{y_c \subseteq y} \psi_c(y_c)$$

Note: This formulation is slightly different from usual in that the set of factors depends on the assignment y.

For most applications, we want the ***maximum a posteriori (MAP)*** configuration:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} p(y|x) = \arg\max_{y \in \mathcal{Y}} \exp \sum_{y_c \subseteq y} \psi_c(y_c)$$

MCMC Based MAP Inference

Finding $\hat{y}$ takes a long time since the number of configurations can be exponential.
We use MCMC-based algorithms to efficiently explore the space.

Proposal Function q:
Proposes a *small* change to the current configuration y to result in y'.

Acceptance Ratio α:

$$\alpha(y, y') = \min\left(1, \left(\frac{p(y')}{p(y)}\right)^{1/t} \frac{q(y)}{q(y')}\right)$$

where t is the temperature, which is slowly reduced as inference progresses.

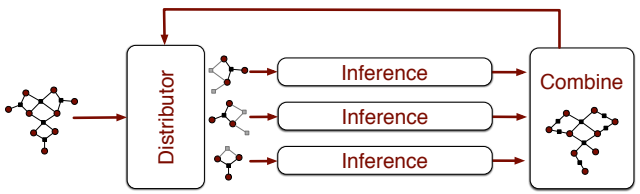
Calculating $p(y')$ / $p(y)$ is easy since the normalization constant cancels.
Furthermore, if the change is small, calculating this ratio can be very efficient since it only involves factors that have *changed* between y and y':

$$\frac{p(y')}{p(y)} = \exp\left\{\sum_{y'_c \subseteq y'} \psi_c(y'_c) - \sum_{y_c \subseteq y} \psi_c(y_c)\right\}$$

Distributed MAP Inference

Consider two jumps ($y \rightarrow y_a$) and ($y \rightarrow y_b$) such that they are *mutually exclusive*, i.e. there is no overlap in the variables and factors used to compute $\alpha(y, y_a)$ and $\alpha(y, y_b)$.

These two jumps can be evaluated independently, and can be used to update the current configuration independently. This observation provides an opportunity to parallelize.



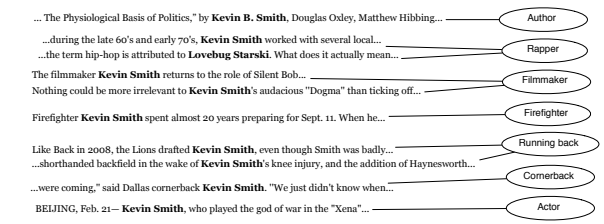
The distributor divides the variables amongst the inference workers.
Propose changes that do not require evaluation of factors across workers.

Thus, proposals are *mutually exclusive* , and correctly compute the acceptance score.

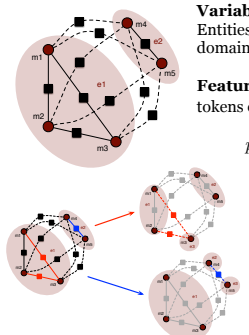
Cross-Document Coreference

Input: set of text *mention* strings from a large corpus, with the words in their contexts

Output: *Clustering* of these mentions such that mentions in each cluster refers to the same *underlying entity*.



Pairwise Model for Coreference



Variables: Mentions m, and Entities e
Entities can take any set of mentions as a value, i.e. domain of an entity variable is a set of all possible sets

Features: Φij is the shifted cosine similarity of the context tokens of mentions i and j. $\phi_{ij} = \cos(m_i^x, m_j^x) - b$

$$p(\vec{e}) \propto w_a \sum_{m_i \underset{e}{\sim} m_j} \phi(m_i, m_j) - w_r \sum_{m_i \underset{e}{\not\sim} m_j} \phi(m_i, m_j)$$

e.g. $p(e_1, e_2) \propto w_a(\phi_{12} + \phi_{13} + \phi_{23} + \phi_{45}) - w_r(\phi_{15} + \phi_{25} + \phi_{35} + \phi_{14} + \phi_{24} + \phi_{34})$

For a proposal that moves mention m from e_1 to e_3, only the factors with mentions in e_1 & e_3 are used.

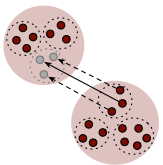
When distributing variables, we ensure:
1) mentions of an entity are on the same machine
2) propose moves amongst entities on same machine

Latent Variables for Distribution and Proposals

Random distribution helps *mixing*, but can lead to wasteful proposals.
Furthermore, within each worker there may be unnecessary proposals.

Sub-Entities

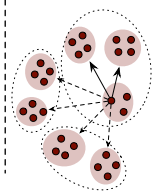
For better within-machine proposals



- clusters amongst coreferent mentions
- proposals move whole *sub-entity* together.

Super-Entities

For better distribution amongst workers



- clusters of entities that are similar
- distribution moves entities in the same *super-entity* to the same worker.

These two sets of variables and factors can be used in a *combined* model. The potentials for these additional variables are set similar to pairwise, with a different b.

Inference cycles between the stages, fixing variables of two stages and sampling the third.

Person-X Evaluation

It is difficult to manually label data for large-scale cross-document coreference. We will *hide* mention strings and use them as true entity labels. For example:

And we can see **Barack Obama** walking slowly, behind Speaker Nancy Pelosi and Senator Dianne Feinstein...

... the economy by unleashing another frustrated tirade against **Barack Obama**," Tommy Victor, an Obama campaign spokesman...

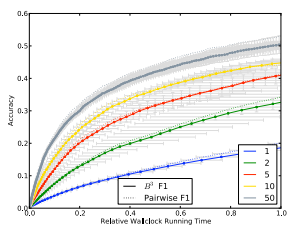
People who heard President **Barack Obama** out on the subject here last month left a meeting divided...

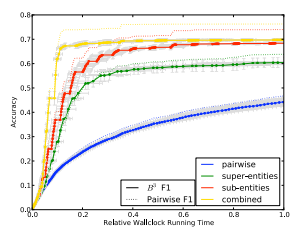
Secretary of State **Hillary Rodham Clinton**, just last week, proclaimed that the world was entering...

Hillary Rodham Clinton speaks no foreign languages, but has visited 90 countries...

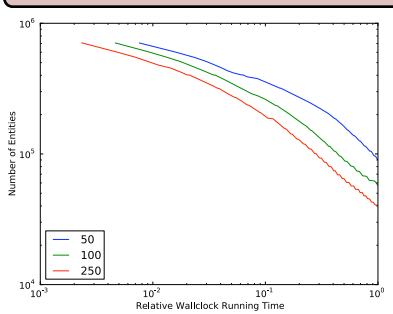
... played a central role, with Secretary of State **Hillary Rodham Clinton** and Mr. Mitchell taking part in the meetings.

25,000 mentions with 50 unique mention strings (entities), averaged over 5 runs





Preliminary Large-Scale Experiments



Person-X evaluation is unreliable as the number of mentions grows:
- same string ≠ same entity
- different string ≠ different entities

Instead, we explore speed of convergence of inference for 1 million mentions form NYT.