Linear Programming Decoders in NLP:
Integer Programming, Message Passing, Dual Decomposition

André Martins

EMNLP 2014: Tutorials, Doha, Qatar, October 25, 2014
Slides online at http://tiny.cc/lpdnlp.
**Structured Prediction and NLP**

**Structured prediction**: a machine learning framework for predicting structured, constrained, and interdependent outputs

**NLP** deals with *structured* and *ambiguous* textual data (Smith, 2011):

- machine translation
- speech recognition
- syntactic parsing
- semantic parsing
- information extraction
- ...
Map **sentences** to their **syntactic structure**.

- A lexicalized syntactic formalism
- Grammar functions represented as lexical relationships (dependencies)

(Eisner, 1996; McDonald et al., 2005; Nivre et al., 2006; Koo et al., 2007)
Multi-Document Summarization

Map a set of related documents to a brief summary.

**CNN**

**Story Highlights**
- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar’s president
- He’s the first sitting U.S. president to visit Myanmar, also known as Burma
- Obama encourages the country to continue a “remarkable journey”
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit

*(CNN) -- Barack Obama met with Nobel Peace Prize winner Aung San Suu Kyi at her home in Myanmar on Monday, praising her “courage and determination” during a historic visit to the once repressive and secretive country.

The first sitting U.S. president to visit Myanmar, Obama urged its leaders, who have embarked on a series of far-reaching political and economic reforms since 2011, not to extinguish the “flickets of progress that we have seen.”

Obama said that his visit to the lakeside villa where the pro-democracy icon spent years under house arrest marked a new chapter between the two countries.

“Here, through so many difficult years, is where she has displayed such unbreakable courage and determination,” Obama told reporters, standing next to his fellow Nobel peace laureate. “It is here where she showed that human freedom and human dignity cannot be denied.”

The country, which is also known as Burma, was ruled by military leaders until early 2011 and for decades was politically and economically cut off from the rest of the world.

Suu Kyi acknowledged that Myanmar’s opening up would be difficult.

**The New York Times**

**YANGON, Myanmar — President Obama journeyed to this storied tropical outpost of pagodas and jungles on Monday to “extend the hand of friendship” as a land long tormented by repression and poverty begins to throw off military rule and emerge from decades of isolation.**

The visit was intended to show support for the reforms put in place by Thein Sein’s government since the end of military rule in November 2010. Activists have warned that the visit may be too hasty - political prisoners remain behind bars and ethnic conflicts in border areas are unresolved.

**BBC NEWS**
Multi-Document Summarization

Map a set of related documents to a brief summary.

**CNN**

*STORY HIGHLIGHTS*

- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar’s president
- He’s the first sitting U.S. president to visit Myanmar, also known as Burma
- Obama encourages the country to continue a "remarkable journey"
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit

(CNN) -- Barack Obama met with Nobel Peace Prize winner Aung San Suu Kyi at her home in Myanmar on Monday, praising her "courage and determination" during a historic visit to the once repressive and secretive country.

The first sitting U.S. president to visit Myanmar, Obama urged its leaders, who have embarked on a series of far-reaching political and economic reforms since 2011, not to extinguish the "flickers of progress that we have seen."

Obama said that his visit to the lakeside villa where the pro-democracy icon spent years under house arrest marked a new chapter between the two countries.

"Here, through so many difficult years, is where she has displayed such unbreakable courage and determination," Obama told reporters, standing next to his fellow Nobel peace laureate. "It is here where she showed that human freedom and human dignity cannot be denied."

The country, which is also known as Burma, was ruled by military leaders until early 2011 and for decades was politically and economically cut off from the rest of the world.

Suu Kyi acknowledged that Myanmar’s opening up would be difficult.

**The New York Times**

**[YANGON, MYANMAR]** -- President Obama journeyed to this storied tropical outpost of pagodas and jungles on Monday to “extend the hand of friendship” as a land long tormented by repression and poverty begins to throw off military rule and emerge from decades of isolation.

The visit was intended to show support for the reforms put in place by Thein Sein’s government since the end of military rule in November 2010. Activists have warned that the visit may be too hasty - political prisoners remain behind bars and ethnic conflicts in border areas are unresolved.

**BBC News**
1. Greedy algorithms can deal with rich histories, but they are suboptimal and suffer from error propagation.

2. Simple, tractable models permit exact decoding, but they make too stringent factorization assumptions.
Current State of Affairs

1 Greedy algorithms can deal with rich histories, but they are suboptimal and suffer from error propagation

2 Simple, tractable models permit exact decoding, but they make too stringent factorization assumptions

We’d like fast predictors with global features and constraints, but how?
Related Recent Tutorials

- Dual Decomposition and Lagrangian Relaxation for Inference in NLP (Rush & Collins ACL 2011)
- Structured Predictions in NLP: Constrained Conditional Models and Integer Linear Programming (Srikumar, Goldwasser & Roth NAACL 2012)
- Variational Inference in Structured NLP Models (Burkett & Klein NAACL 2012)
- Structured Belief Propagation for NLP (Gromley & Eisner ACL 2014)
This Tutorial: Linear Programming Decoders

We’ll provide a unified view over these approaches (ILPs, message-passing, dual decomposition)

We’ll focus on MAP decoding, but touch briefly on marginal decoding

We’ll illustrate with three applications:

1. Turbo Parsing
2. Compressive Summarization
3. Joint Coreference Resolution and Quote Attribution

(Companion software: AD³ toolkit)
Outline

1 Structured Prediction and Factor Graphs

2 Integer Linear Programming

3 Message-Passing Algorithms
   - Sum-Product
   - Max-Product

4 Dual Decomposition

5 Applications

6 Conclusions
Outline

1. Structured Prediction and Factor Graphs
2. Integer Linear Programming
3. Message-Passing Algorithms
   - Sum-Product
   - Max-Product
4. Dual Decomposition
5. Applications
6. Conclusions
Structured Prediction

- Input set $X$
- For each $x \in X$: a large set of candidate outputs $\mathcal{Y}(x)$
- A compatibility function $F_w(x, y)$ induced by a model $w$
  (Linear model: $F_w(x, y) = w^\top f(x, y)$)
Structured Prediction

- Input set $\mathcal{X}$
- For each $x \in \mathcal{X}$: a large set of candidate outputs $\mathcal{Y}(x)$
- A compatibility function $F_w(x, y)$ induced by a model $w$
  (Linear model: $F_w(x, y) = \mathbf{w}^\top \mathbf{f}(x, y)$)
- **Training problem**: learn the model $w$ from data $\{(x_i, y_i)\}_{i=1}^M$
- **Decoding problem (our focus)**:

$$
\hat{y} = \arg \max_{y \in \mathcal{Y}(x)} F_w(x, y)
$$
Structured Prediction

- Input set \( \mathcal{X} \)
- For each \( x \in \mathcal{X} \): a **large** set of candidate outputs \( \mathcal{Y}(x) \)
- A compatibility function \( F_w(x, y) \) induced by a model \( w \)
  (Linear model: \( F_w(x, y) = w^\top f(x, y) \))
- **Training problem:** learn the model \( w \) from data \( \{\langle x_i, y_i \rangle\}_{i=1}^M \)
- **Decoding problem (our focus):**

\[
\hat{y} = \arg \max_{y \in \mathcal{Y}(x)} F_w(x, y)
\]

- **Key assumption:** \( F_w \) decomposes into (overlapping) parts
Three Important Questions

- Representation?
- Decoding/Inference?
- Learning the parameters?
Recap: Hidden Markov Models

$F_w$ is a log-probability, factoring over emissions and transitions.

$$
P(x, y) = \prod_i P(x_i|y_i) P(y_i|y_{i-1})$$

[Diagram of HMM states and transitions]
Recap: Hidden Markov Models

$F_w$ is a log-probability, factoring over emissions and transitions.

$$P(x, y) = \prod_i \underbrace{P(x_i|y_i)}_{\text{emissions}} \underbrace{P(y_i|y_{i-1})}_{\text{transitions}}$$

$\psi_i(y_i) := P(x_i|y_i)$ (unary potentials)
Recap: Hidden Markov Models

$F_w$ is a log-probability, factoring over emissions and transitions.

$$P(x, y) = \prod_i \left( P(x_i | y_i) P(y_i | y_{i-1}) \right)$$

$\psi_{i,i-1}(y_i, y_{i-1}) := P(y_i | y_{i-1})$ (pairwise potentials)

$\psi_i(y_i) := P(x_i | y_i)$ (unary potentials)
Recap: Hidden Markov Models

$F_w$ is a log-probability, factoring over emissions and transitions.

$$
P(x, y) = \prod_i \left( \underbrace{P(x_i | y_i)}_{\text{emissions}} \underbrace{P(y_i | y_{i-1})}_{\text{transitions}} \right)
= \prod_i \psi_i(y_i) \prod_i \psi_{i-1}(y_i, y_{i-1})$$

\[\psi_{i-1}(y_i, y_{i-1}) := P(y_i | y_{i-1})\] (pairwise potentials)

\[\psi_i(y_i) := P(x_i | y_i)\] (unary potentials)
Recap: Hidden Markov Models

- **Representation?** Directed sequence model.
- **Decoding/Inference?** Viterbi/forward-backward algorithms.
- **Learning the parameters?** Maximum likelihood (count and normalize).
Recap: Conditional Random Fields

Same factorization, but globally normalized.

\[
P(y|x) = \frac{1}{Z(w, x)} \exp \left( \sum_i w^T f_i(x, y_i) + \sum_i w^T f_{i,i-1}(x, y_i, y_{i-1}) \right)
\]
Recap: Conditional Random Fields

Same factorization, but globally normalized.

$$\mathbb{P}(y|x) = \frac{1}{Z(w, x)} \exp \left( \sum_i w^\top f_i(x, y_i) + \sum_i w^\top f_{i, i-1}(x, y_i, y_{i-1}) \right)$$

ψᵢ(𝐲ᵢ) := exp(w^⊤f(x, yᵢ)) (unary potentials)
Recap: Conditional Random Fields

Same factorization, but globally normalized.

\[
\mathbb{P}(y|x) = \frac{1}{Z(w, x)} \exp \left( \sum_i w^\top f_i(x, y_i) + \sum_i w^\top f_{i-1}(x, y_i, y_{i-1}) \right)
\]

- \(\psi_{i,i-1}(y_i, y_{i-1}) := \exp(w^\top f(x, y_i, y_{i-1}))\) (pairwise potentials)
- \(\psi_i(y_i) := \exp(w^\top f(x, y_i))\) (unary potentials)
Recap: Conditional Random Fields

Same factorization, but globally normalized.

\[
P(y|x) = \frac{1}{Z(w,x)} \exp \left( \sum_i w^\top f_i(x,y_i) + \sum_i w^\top f_{i-1}(x,y_i,y_{i-1}) \right)
\]

\(\propto \prod_i \psi_i(y_i) \prod_i \psi_{i-1}(y_i, y_{i-1})\)

\(\psi_{i-1}(y_i, y_{i-1}) := \exp(w^\top f(x, y_i, y_{i-1}))\) (pairwise potentials)

\(\psi_i(y_i) := \exp(w^\top f(x, y_i))\) (unary potentials)
Recap: Conditional Random Fields

- **Representation?** Undirected sequence model.
- **Decoding/Inference?** Viterbi/forward-backward algorithms.
- **Learning the parameters?** Maximum conditional likelihood (convex optimization).
HMMs and CRFs are two instances of *graphical models*.

In general, graphical models come in two flavours:

- Directed (Bayesian Networks)
- Undirected (Markov Networks)
Bayesian Networks

Useful to express causality relations.

Factors are conditional probability tables.

\[ P(y) = P(y_1)P(y_2|y_1, y_4)P(y_3|y_2, y_5)P(y_4|y_5)P(y_5) \]

\[ P(y) = \prod_i P(y_i|\text{parents}(y_i)) \]
Bayesian Networks

Useful to express causality relations.

Factors are conditional probability tables.

\[ P(y) = P(y_1)P(y_2|y_1, y_4)P(y_3|y_2, y_5)P(y_4|y_5)P(y_5) \]

\[ P(y) = \prod_{i} P(y_i|\text{parents}(y_i)) \]
Markov Networks

Useful to express **correlations between variables**.

Factors correspond to **cliques of the graph**.

\[
P(y) = \frac{1}{\mathcal{Z}} \psi_{124}(y_1, y_2, y_4) \psi_{235}(y_2, y_3, y_5) \psi_{245}(y_2, y_4, y_5)
\]
Markov Networks

Useful to express correlations between variables.

Factors correspond to cliques of the graph.

\[
\mathbb{P}(y) = \frac{1}{Z} \psi_{124}(y_1, y_2, y_4) \psi_{235}(y_2, y_3, y_5) \psi_{245}(y_2, y_4, y_5)
\]
Conditional Independence

Graphical models are a great tool for modeling conditional independence. They link properties of the probability distribution with properties of the graph (reachability, D-separation, etc.).

Lots of literature about this: Pearl (1988); Lauritzen (1996); Koller and Friedman (2009)
An Intermediate Representation: Factor Graph

A bipartite graph with **variable nodes** and **factor nodes**

It makes *explicit* the factors of the distribution

\[
\mathbb{P}(y) \propto \prod_{i} \psi_i(y_i) \times \prod_{s} \psi_s(y_s)
\]

With unary potentials only, all variables would be independent

Higher-order potentials can model correlations, impose soft/hard constraints, etc.
Example: Low-Density Parity Check Codes

A message is transmitted through a noisy channel, corrupting some bits. Redundancy can help decoding the message, e.g. via additional parity check bits that can detect/correct errors (error-correcting codes). High-level idea: increase redundancy to build more accurate decoders.

(Adapted from MacKay 2003.)
Inference/Decoding

\[ P_\psi(y|x) = \frac{1}{Z(\psi, x)} \times \prod_i \psi_i(y_i) \times \prod_s \psi_s(y_s) \]

Two decoding problems:

- **MAP decoding:** compute \( \hat{y} = \arg \max_y P_\psi(y|x) \)
- **Marginal decoding:** compute every \( P_\psi(y_i|x) \) and \( P_\psi(y_s|x) \); and evaluate the partition function \( Z(\psi, x) \)

Sometimes easy, in general intractable...
When is Decoding Easy?

- independent variables (trivial)
- sequence models (Viterbi, forward-backward)
- graphical models without cycles (variable elimination, belief propagation)
- graphical models with low treewidth (junction tree algorithm)
When is Decoding Easy?

- independent variables (trivial)
- sequence models (Viterbi, forward-backward)
- graphical models without cycles (variable elimination, belief propagation)
- graphical models with low treewidth (junction tree algorithm)

In general, for graphs with cycles, MAP decoding is NP-hard and marginal decoding is \#P-hard
When is Decoding Easy?

- independent variables (trivial)
- sequence models (Viterbi, forward-backward)
- graphical models without cycles (variable elimination, belief propagation)
- graphical models with low treewidth (junction tree algorithm)

In general, for graphs with cycles, MAP decoding is NP-hard and marginal decoding is \#P-hard

Note: tractability depends not only on the topology, but also on the potentials
Example: Ising and Potts Models

Ising/Potts grid  Ernst Ising, 1900–1998  Ren Potts, 1925–2005

All factors are pairwise, variables are binary (Ising) or multi-class (Potts)
Example: Ising and Potts Models

All factors are pairwise, variables are binary (Ising) or multi-class (Potts)
MAP decoding is tractable for attractive Ising models (i.e. Ising models with supermodular log-potentials):

$$\log \psi_{ij}(1, 1) + \log \psi_{ij}(0, 0) \geq \log \psi_{ij}(0, 1) + \log \psi_{ij}(1, 0)$$

Good approximations for attractive Potts models
Example: Ising and Potts Models

All factors are pairwise, variables are binary (Ising) or multi-class (Potts)

MAP decoding is tractable for attractive Ising models (i.e. Ising models with supermodular log-potentials):

$$\log \psi_{ij}(1, 1) + \log \psi_{ij}(0, 0) \geq \log \psi_{ij}(0, 1) + \log \psi_{ij}(1, 0)$$

Good approximations for attractive Potts models

... but the general case is NP-hard and hard to approximate
Example: Skip-Chain CRFs

Skip-chain CRFs are useful to model long-range dependencies.

Skip-chains introduce cycles, making decoding more expensive.

We could write this information in the “state” and still decode with dynamic programming, but that would blow up the number of states.
Beyond Graphical Models

Some NLP problems (e.g. parsing) require representations beyond graphical models.

Dynamic programming algorithms (CKY, inside-outside) still work for those representations.

Example: case-factor diagrams (McAllester et al., 2008).

Other problems (e.g. matching, spanning trees) can be solved with combinatorial algorithms not related with dynamic programming.

All these can still be represented as GMs by “generalizing” the notion of factor.
Factors as Machines

\[ Y_1 \rightarrow Y_2 \rightarrow Y_3 \rightarrow Y_4 \rightarrow Y_5 \]
Factors as Machines

\[ Y_1 \quad Y_2 \quad Y_3 \quad Y_4 \quad Y_5 \]
Three Kinds of Factors

Let $N(s)$ denote the set of variables that are *neighbors* of factor $s$. (Its cardinality $|N(s)|$ is called the *degree* of $s$.)

1. **Dense factors:** $\psi_s(y_s)$ has all $O(\exp(|N(s)|))$ degrees of freedom
2. **Structured factors:** $\psi_s(y_s)$ has internal structure
3. **Hard constraint factors:**

\[
\psi_s(y_s) := \begin{cases} 
1, & \text{if } y_s \in Y_s \\
0, & \text{otherwise.}
\end{cases}
\]
Examples of Structured Factors

- a factor for bipartite matching (Duchi et al., 2007)
- combining a sequential model (POS tagger) with a PCFG (Rush et al., 2010)
- combining CCG parsing and supertagging (Auli and Lopez, 2011)
- dependency parsing with head automata (Smith and Eisner, 2008; Koo et al., 2010)
- handling string-valued variables with factors that are finite state transducers (Dreyer and Eisner, 2009)
- inversion transduction grammar constraint (Burkett and Klein, 2012)
Examples of Hard Constraint Factors

Logic factors: can express arbitrary FOL constraints

- Applications: Markov logic networks (Richardson and Domingos, 2006), constrained conditional models (Roth and Yih, 2004)

Knapsack factors: can express budget constraints

- Applications: summarization, diversity problems,...

(Martins et al., 2011b; Almeida and Martins, 2013; Martins et al., 2014)
GMs, $\psi \geq 0$
(dense, structured,
hard-constraint factors)

tractable problems

GMs $\psi > 0$
(dense factors)

problems solvable with
dynamic programming

non-projective
dependency
grammars

GMs $\psi > 0$, no cycles

projective
dependency
grammars

PCFGs
Approximate Decoding

What to do when exact decoding is intractable?

- Sampling methods (MCMC, etc.)
- Mean field algorithms
- LP relaxations
- Message-passing
- Dual decomposition

We’ll highlight connections between several of these methods.
Approximate Decoding

What to do when exact decoding is intractable?

- Sampling methods (MCMC, etc.)
- Mean field algorithms
- LP relaxations
- Message-passing
- Dual decomposition

We’ll highlight connections between several of these methods.
“Local” denotes independent problems within the scope of each factor

“Global” involves a global assignment of variables, consistent across factors

Key idea: “glue” the local evidence at the factors to obtain a global assignment

Our assumption: local decoding is easy, for every factor

We want to build a good (approximate) global decoder by invoking the local decoders.
## What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD$^3$ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>
Outline

1. Structured Prediction and Factor Graphs
2. Integer Linear Programming
3. Message-Passing Algorithms
   - Sum-Product
   - Max-Product
4. Dual Decomposition
5. Applications
6. Conclusions
Linear Programming
(Kantorovich, 1940; Dantzig, 1947)

\[
\begin{align*}
\max \quad & s^\top z \\
\text{s.t.} \quad & a_i^\top z \leq b_i, \quad i = 1, \ldots, N.
\end{align*}
\]

Linear objective

Linear constraints
Linear Programming
(Kantorovich, 1940; Dantzig, 1947)

\[
\max_z s^\top z \\
s.t. \quad a_i^\top z \leq b_i, \ i = 1, \ldots, N.
\]
max \limits_{z} \mathbf{s}^\top z \quad \text{Linear objective}

\text{s.t.} \quad \mathbf{a}_i^\top z \leq b_i, \quad i = 1, \ldots, N. \quad \text{Linear constraints}
Linear Programming
(Kantorovich, 1940; Dantzig, 1947)

\[
\begin{align*}
\max_{z} & \quad s^\top z \\
\text{s.t.} & \quad a_i^\top z \leq b_i, \quad i = 1, \ldots, N.
\end{align*}
\]

Linear objective
Linear constraints
Linear Programming
(Kantorovich, 1940; Dantzig, 1947)

\[
\begin{align*}
\max_z \quad & s^\top z \\
\text{s.t.} \quad & a_i^\top z \leq b_i, \quad i = 1, \ldots, N.
\end{align*}
\]
Linear Programming
(Kantorovich, 1940; Dantzig, 1947)

- If feasible and bounded, the solution is always attained at a vertex
- Can be solved in \textit{polynomial time} (Khachiyan, 1980)
- Lots of off-the-shelf solvers (CPLEX, Gurobi, GLPK, LP-Solve, etc.)
**Integer Linear Programming**

\[
\max_z \quad s^\top z \\
\text{s.t.} \quad a_i^\top z \leq b_i, \quad i = 1, \ldots, N, \\
\quad z \text{ integer.}
\]

Linear objective

Linear constraints

\[a_i^\top z \leq b_i\]

\[z^* \text{ (solution of the LP)}\]

\[s \text{ (score vector)}\]
**Integer Linear Programming**

\[
\begin{align*}
\max_z & \quad \mathbf{s}^\top \mathbf{z} \\
\text{s.t.} & \quad \mathbf{a}_i^\top \mathbf{z} \leq b_i, \quad i = 1, \ldots, N, \\
& \quad \mathbf{z} \text{ integer.}
\end{align*}
\]
**Integer Linear Programming**

\[
\begin{align*}
\max_{z} & \quad s^\top z \\
\text{s.t.} & \quad a_i^\top z \leq b_i, \quad i = 1, \ldots, N, \\
& \quad z \text{ integer.}
\end{align*}
\]
Integer Linear Programming

\[
\begin{align*}
\max_z & \quad s^\top z \\
\text{s.t.} & \quad a_i^\top z \leq b_i, \quad i = 1, \ldots, N, \\
& \quad z \text{ integer.}
\end{align*}
\]

Linear objective
Linear constraints

André Martins (Priberam/IT)
LP Decoders in NLP
http://tiny.cc/lpdnlp
**Integer Linear Programming**

\[
\max_z \quad s^\top z \\
\text{s.t.} \quad a_i^\top z \leq b_i, \quad i = 1, \ldots, N, \\
z \text{ integer.}
\]

Linear objective

Linear constraints

André Martins (Priberam/IT)

LP Decoders in NLP

http://tiny.cc/lpdnlp
Integer Linear Programming

In general, NP-hard (Karp, 1972)

Existing solvers are effective for small instances, but don’t scale

**LP relaxation:** drops the integer constraints

- Gives an upper bound of the solution of the ILP
- A common first step in exact algorithms (branch-and-bound, cutting plane, branch-and-cut)

Here’s a very simple approximate algorithm:

1. Solve the LP relaxation
2. If the solution is integer, then it *is* the solution of the ILP
3. Otherwise, apply a rounding heuristic (problem-dependent)
Two Representations of Polytopes

Intersection of half-spaces (H-representation) or convex hull of a set of vertices (V-representation)

\[ \mathbf{a}_i^T \mathbf{z} \leq b_i \]

To call a solver, we need to specify a concise H-representation.

However, it may be difficult or impossible to obtain one if all we have is a V-representation.

We next show how this relates to MAP decoding...
Structured Outputs as Bit-Vectors

One indicator \( p_i(y_i) \) per each variable state

One indicator \( q_s(y_s) \) per each factor configuration

Overall: each global output \( y \in \mathcal{Y}(x) \) is mapped to a bit-vector

Note: not all bit vectors are valid (they must be consistent)

André Martins (Priberam/IT)

LP Decoders in NLP

http://tiny.cc/lpdnlp
Structured Outputs as Bit-Vectors

- One indicator $p_i(y_i)$ per each variable state
Structured Outputs as Bit-Vectors

One indicator $p_i(y_i)$ per each variable state

Overall: each global output $y \in Y(x)$ is mapped to a bit-vector

Note: not all bit vectors are valid (they must be consistent)

- One indicator $p_i(y_i)$ per each variable state
Structured Outputs as Bit-Vectors

- One indicator $p_i(y_i)$ per each variable state
- One indicator $q_s(y_s)$ per each factor configuration
Structured Outputs as Bit-Vectors

One indicator $p_i(y_i)$ per each variable state

One indicator $q_s(y_s)$ per each factor configuration

Note: not all bit vectors are valid (they must be consistent)
Structured Outputs as Bit-Vectors

- One indicator $p_i(y_i)$ per each variable state
- One indicator $q_s(y_s)$ per each factor configuration
Structured Outputs as Bit-Vectors

- One indicator \( p_i(y_i) \) per each variable state
- One indicator \( q_s(y_s) \) per each factor configuration
- Overall: each global output \( y \in \mathcal{Y}(x) \) is mapped to a bit-vector
Structured Outputs as Bit-Vectors

- One indicator $p_i(y_i)$ per each variable state
- One indicator $q_s(y_s)$ per each factor configuration
- Overall: each global output $y \in \mathcal{Y}(x)$ is mapped to a bit-vector
Structured Outputs as Bit-Vectors

- One indicator $p_i(y_i)$ per each variable state
- One indicator $q_s(y_s)$ per each factor configuration
- Overall: each global output $y \in \mathcal{Y}(x)$ is mapped to a bit-vector
- **Note:** not all bit vectors are valid (they must be consistent)
Marginal Polytope (Wainwright and Jordan, 2008)

Points of $\text{MARG}(G)$ correspond to realizable marginals (more later).

This is a V-representation, what about an H-representation?

34 bits

$y(x)$
Marginal Polytope (Wainwright and Jordan, 2008)

Vertices of $\text{MARG}(G)$ correspond to outputs $Y(x)$.

Points of $\text{MARG}(G)$ correspond to realizable marginals (more later).

This is a V-representation, what about an H-representation?
Marginal Polytope (Wainwright and Jordan, 2008)

Vertices of $\text{MARG}(G)$ correspond to outputs $y(x)$

Points of $\text{MARG}(G)$ correspond to realizable marginals (more later)

This is a V-representation, what about an H-representation?
Marginal Polytope (Wainwright and Jordan, 2008)

Vertices of $\text{MARG}(G)$ correspond to outputs $y(x)$

Points of $\text{MARG}(G)$ correspond to realizable marginals (more later)

- Vertices of $\text{MARG}(G)$ correspond to outputs $y(x)$
- Points of $\text{MARG}(G)$ correspond to realizable marginals (more later)
Vertices of $\text{MARG}(G)$ correspond to outputs $y(x)$
Points of $\text{MARG}(G)$ correspond to realizable marginals (more later)
Marginal Polytope (Wainwright and Jordan, 2008)

- Vertices of $\text{MARG}(G)$ correspond to outputs $y(x)$
- Points of $\text{MARG}(G)$ correspond to realizable marginals (more later)
- This is a V-representation, what about an H-representation?
H-Representation With Integer Constraints

In general, there’s no concise H-representation for \( \text{MARG}(G) \) ... but we can represent its vertices if \emph{integer constraints} are permitted:

\[
\sum_{y_s} q_s(y_s) = 1, \quad q_s(y_s) \geq 0, \quad \forall y_s \in Y_s \quad \text{(normalization)}
\]

\[
p_i(y_i) = \sum_{y_s \sim y_i} q_s(y_s), \quad \forall i \in N(s) \quad \text{(marginalization)}
\]

\( q \) is integer \quad \text{(integer constraints)}

This will open the door for formulating MAP decoding as an ILP.
MAP Decoding as an ILP

Recall the MAP decoding problem:

\[ \hat{y} = \arg \max_{y \in \mathcal{Y}(x)} P_\psi(y|x) \]

\[ = \arg \max_{y \in \mathcal{Y}(x)} \frac{1}{Z(\psi, x)} \prod_i \psi_i(y_i) \prod_s \psi_s(y_s) \]

\[ = \arg \max_{y \in \mathcal{Y}(x)} \sum_i \theta_i(y_i) + \sum_s \theta_s(y_s), \]

where \( \theta_i(y_i) := \log \psi_i(y_i) \) and \( \theta_s(y_s) := \log \psi_s(y_s) \)

We can rewrite this as an ILP:

\[
\text{maximize } \sum_i \sum_{y_i} \theta_i(y_i)p_i(y_i) + \sum_s \sum_{y_s} \theta_s(y_s)q_s(y_s)
\]

subject to \((p, q) \in \text{MARG}(G)\)
Local Polytope

Obtained by relaxing the integer constraints

Regard $p_i$ and $q_s$ as probability distributions that must be **locally consistent**:

\[
\sum_{y_s} q_s(y_s) = 1, \quad q_s(y_s) \geq 0, \quad \forall y_s \in y_s \quad \text{(normalization)}
\]

\[
p_i(y_i) = \sum_{y_s \sim y_i} q_s(y_s), \quad \forall i \in N(s) \quad \text{(marginalization)}
\]

$q$ is integer \hspace{1cm} \text{(integer-constraints)}

The feasible points are **pseudo-marginals** (not necessarily valid marginals)
Local and Marginal Polytopes

\[ \text{LOCAL}(G) \] is an outer bound of \[ \text{MARG}(G) \]. It contains all the integer vertices of \[ \text{MARG}(G) \], plus spurious fractional vertices. If the graph has no cycles, then \[ \text{LOCAL}(G) = \text{MARG}(G) \].
**Local and Marginal Polytopes**

- LOCAL($G$) is an **outer bound** of MARG($G$)
- It contains all the integer vertices of MARG($G$), plus spurious **fractional vertices**
- **If the graph has no cycles, then** LOCAL($G$) = MARG($G$)
Solves a linear relaxation of MAP decoding, replacing MARG(G) by LOCAL(G):

\[
\text{maximize } \sum_i \sum_{y_i} \theta_i(y_i)p_i(y_i) + \sum_s \sum_{y_s} \theta_s(y_s)q_s(y_s)
\]

subject to \((p, q) \in \text{LOCAL}(G)\)

If the solution is integer, we solved the problem exactly; otherwise, apply a rounding heuristic

**Runtime is polynomial, but the procedure is only approximate.**
What About Hard Constraint Factors?

Logic and knapsack/budget constraints can also be expressed *linearly*
Logic/Budget Constraints

Assume $z_1, z_2, \ldots \in \{0, 1\}$ (binary variables)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Statement</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implication</td>
<td>if $z_1$ then $z_2$</td>
<td>$z_1 \leq z_2$</td>
</tr>
<tr>
<td>Negation</td>
<td>$z_1$ iff $\neg z_2$</td>
<td>$z_1 = 1 - z_2$</td>
</tr>
<tr>
<td>OR</td>
<td>$z_1$ or $z_2$ or $z_3$</td>
<td>$z_1 + z_2 + z_3 \geq 1$</td>
</tr>
<tr>
<td>XOR</td>
<td>$z_1$ xor $z_2$ xor $z_3$</td>
<td>$z_1 + z_2 + z_3 = 1$</td>
</tr>
<tr>
<td>OR-OUT</td>
<td>$z_{12} = z_1 \lor z_2$</td>
<td>$z_{12} \geq z_1, z_{12} \geq z_2,$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$z_{12} \leq z_1 + z_2$</td>
</tr>
<tr>
<td>AND-OUT</td>
<td>$z_{12} = z_1 \land z_2$</td>
<td>$z_{12} \leq z_1, z_{12} \leq z_2,$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$z_{12} \geq z_1 + z_2 - 1$</td>
</tr>
<tr>
<td>Budget</td>
<td>at most $B$ active units</td>
<td>$\sum_i z_i \leq B$</td>
</tr>
<tr>
<td>Knapsack</td>
<td>at most $B$ total weight</td>
<td>$\sum_i w_i z_i \leq B$</td>
</tr>
</tbody>
</table>

More complex expressions via composition and De Morgan’s laws
Summing Up ILPs

- MAP decoding can be expressed as an Integer Linear Program (ILP)
- Logic constraints can be incorporated easily
- Structured factors are harder (they need to be disassembled)
- The ILP can be relaxed for approximate decoding (LP-MAP)
- Geometrically: an outer bound of the *marginal polytope*
- The relaxation is tight if the graph $G$ does not have cycles
- **Disadvantage:** an off-the-shelf LP solver won't exploit the *modularity* of the problem
- **Algorithms that exploit the structure of the LP will be the topic of the remaining sections**
Outline

1. Structured Prediction and Factor Graphs
2. Integer Linear Programming
3. Message-Passing Algorithms
   - Sum-Product
   - Max-Product
4. Dual Decomposition
5. Applications
6. Conclusions
Motivating Example: Counting Soldiers

(Adapted from MacKay 2003 and Gormley & Eisner ACL’14 tutorial.)
Motivating Example: Counting Soldiers

(Adapted from MacKay 2003 and Gormley & Eisner ACL’14 tutorial.)
Motivating Example: Counting Soldiers

(Adapted from MacKay 2003 and Gormley & Eisner ACL’14 tutorial.)
Motivating Example: Counting Soldiers

(Adapted from MacKay 2003 and Gormley & Eisner ACL’14 tutorial.)
Motivating Example: Counting Soldiers

... and there's 1 of me. Belief: must be 11+1+1+1 of us.

1 here

11 here

Commander

(Adapted from MacKay 2003 and Gormley & Eisner ACL’14 tutorial.)
Motivating Example: Counting Soldiers

(Adapted from MacKay 2003 and Gormley & Eisner ACL’14 tutorial.)
Motivating Example: Counting Soldiers

(Adapted from MacKay 2003 and Gormley & Eisner ACL’14 tutorial.)
Outline

1 Structured Prediction and Factor Graphs

2 Integer Linear Programming

3 Message-Passing Algorithms
   - Sum-Product
   - Max-Product

4 Dual Decomposition

5 Applications

6 Conclusions
Sum-Product Belief Propagation

Recall that $\mathbb{P}_\psi(y|x) := \frac{1}{Z(\psi, x)} \times \prod_i \psi_i(y_i) \times \prod_s \psi_s(y_s)$

Alternate between computing two kinds of messages:

- **Variable-to-factor:** $m_{i\rightarrow s}(y_i) = \psi_i(y_i) \prod_{s' \in N(i) \setminus \{s\}} n_{s'\rightarrow i}(y_i)$

- **Factor-to-variable:** $n_{s\rightarrow i}(y_i) = \sum_{y_s \sim y_i} \psi_s(y_s) \prod_{j \in N(s) \setminus \{i\}} m_{j\rightarrow s}(y_j)$
Given the messages, we compute local beliefs:

- **Variable beliefs:**

  \[
  p_i(y_i) \propto \psi_i(y_i) \prod_{s \in N(i)} n_{s \rightarrow i}(y_i)
  \]

- **Factor beliefs:**

  \[
  q_s(y_s) \propto \psi_s(y_s) \prod_{i \in N(s)} m_{i \rightarrow s}(y_i)
  \]

If the graph has no cycles, these beliefs converge to the true marginals

\[
  p_i(y_i) \rightarrow \mathbb{P}_\psi(y_i|x), \quad q_s(y_s) \rightarrow \mathbb{P}_\psi(y_s|x)
\]

**Otherwise: loopy BP** (later)
Belief Propagation as Calibration

- **Variable-to-factor messages:**

  \[ m_{i \rightarrow s}(y_i) = \psi_i(y_i) \prod_{s' \in N(i) \setminus \{s\}} n_{s' \rightarrow i}(y_i) = \frac{p_i(y_i)}{n_{s \rightarrow i}(y_i)} \]

- **Factor-to-variable messages:**

  \[ n_{s \rightarrow i}(y_i) = \sum_{y_s \sim y_i} \psi_s(y_s) \prod_{j \in N(s) \setminus \{i\}} m_{j \rightarrow s}(y_j) = \frac{\sum_{y_s \sim y_i} q_s(y_s)}{m_{i \rightarrow s}(y_i)} \]

- **Calibration equations (attained at convergence):**

  \[ p_i(y_i) = \sum_{y_s \sim y_i} q_s(y_s) \]

**Punchline:** to run sum-product BP, we only need *local marginals*
**What Kind of Local Decoding Do We Need?**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD³ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>

![Graphical representation of local decoding operations](image)
What if the graph has cycles?
Loopy Belief Propagation

What if the graph has cycles?

We’ll see that marginal decoding corresponds to optimizing a free energy objective over the marginal polytope.

Sum-product “loopy” BP entails two approximations:

1. Replaces MARG\( (G) \) by LOCAL\( (G) \)
2. Optimizes a Bethe free energy approximation
For any $\psi$, there are marginals $p, q$ in $\text{MARG}(G)$ that parametrize $\mathbb{P}_\psi$.

E.g. if the graph has no cycles:

$$
\mathbb{P}_\psi(y|x) = \frac{1}{Z(\psi, x)} \prod_i \psi_i(y_i) \times \prod_s \psi_s(y_s)
$$

$$
= \prod_i p_i(y_i) 1^{-|N(i)|} \times \prod_s q_s(y_s)
$$

$$
:= \mathbb{P}_{p,q}(y|x)
$$

Therefore: a distribution can be represented as a point in $\text{MARG}(G)$

$\theta := \log(\psi)$ are called \textit{canonical parameters}, and $(p, q)$ \textit{mean parameters}.
Assume a tree-shaped Bayes net (each variable $i$ has a single parent $\pi_i$)

\[
\mathbb{P}(y) = \mathbb{P}(y_0) \prod_{i \neq 0} \mathbb{P}(y_i | y_{\pi_i}) \\
= \mathbb{P}(y_0) \prod_{i \neq 0} \frac{\mathbb{P}(y_i, y_{\pi_i})}{\mathbb{P}(y_{\pi_i})} \\
= \mathbb{P}(y_0) \prod_s \mathbb{P}(y_s) \\
\prod_j \mathbb{P}(y_j) \prod_{i : j = \pi_i} \\
= \frac{\mathbb{P}(y_0) \prod_s \mathbb{P}(y_s)}{\mathbb{P}(y_0)^{|N(0)|} \prod_{j \neq 0} \mathbb{P}(y_j)^{|N(j) - 1|}} \\
= \prod_s \mathbb{P}(y_s) \\
\prod_j \mathbb{P}(y_j)^{|N(j)| - 1} \\
= \prod_i p_i(y_i)^{1 - |N(i)|} \times \prod_s q_s(y_s).
Step #2: Entropy and Log-Partition Function

Entropy of a probability distribution: \( H(P) = - \sum_y P(y) \log P(y) \)

**Definition:** the Fenchel dual of a convex function \( f : \mathbb{R}^D \rightarrow \mathbb{R} \cup \{ +\infty \} \) is the convex function \( f^* : \mathbb{R}^D \rightarrow \mathbb{R} \cup \{ +\infty \} \) defined pointwise as
\[
f^*(v) := \sup_u \left( v^T u - f(u) \right)
\]
Step #2: Entropy and Log-Partition Function

Entropy of a probability distribution: \[ H(\mathbb{P}) = - \sum_y \mathbb{P}(y) \log \mathbb{P}(y) \]

**Definition:** the Fenchel dual of a convex function \( f : \mathbb{R}^D \to \mathbb{R} \cup \{+\infty\} \) is the convex function \( f^* : \mathbb{R}^D \to \mathbb{R} \cup \{+\infty\} \) defined pointwise as
\[ f^*(v) := \sup_u \left( v^\top u - f(u) \right) \]

**Theorem (I):** the log-partition function \( \log Z(\theta) \) and the negative entropy \(-H(\mathbb{P}_{p,q})\) are Fenchel dual:
\[ \log Z(\theta) = \max_{(p,q) \in \text{MARG}(G)} \left( \sum_i \theta_i^\top p_i + \sum_s \theta_s^\top q_s + H(\mathbb{P}_{p,q}) \right) \]

This underlies the well-known duality between maximum likelihood in log-linear models and maximum entropy.
Step #3: Loopy BP as Variational Inference

**Theorem (II):** The maximizers \((p^*, q^*)\) are the true marginals of \(P_\theta\):

\[
(p^*, q^*) = \arg \max_{(p,q) \in \text{MARG}(G)} \sum_i \theta_i^T p_i + \sum_s \theta_s^T q_s + H(P_{p,q})
\]
Step #3: Loopy BP as Variational Inference

Theorem (II): The maximizers \((p^*, q^*)\) are the true marginals of \(\mathbb{P}_\theta\):

\[
(p^*, q^*) = \arg\max_{(p,q) \in \text{MARG}(G)} \sum_i \theta_i^\top p_i + \sum_s \theta_s^\top q_s + H(\mathbb{P}_{p,q})
\]

Drawback: in general, \(\text{MARG}(G)\) and \(H(\mathbb{P}_{p,q})\) are both intractable
Step #3: Loopy BP as Variational Inference

**Theorem (II):** The maximizers \((p^*, q^*)\) are the true marginals of \(P_\theta\):

\[
(p^*, q^*) = \arg \max_{(p, q) \in \text{MARG}(G)} \sum_i \theta_i^\top p_i + \sum_s \theta_s^\top q_s + H(P_{p, q})
\]

**Drawback:** in general, \(\text{MARG}(G)\) and \(H(P_{p, q})\) are both intractable

Yedidia et al. (2001) showed that loopy BP entails two approximations:

1. Replace \(\text{MARG}(G)\) by \(\text{LOCAL}(G)\)
2. Approximate \(H(P_{p, q})\) by the Bethe entropy \(H_{\text{Bethe}}(P_{p, q})\)

Both are exact when the graph does not have cycles
Bethe Entropy Approximation

Derived by “pretending” the graph has no cycles
We have seen

\[ P_\psi(y|x) \approx \prod_i p_i(y_i)^{1-|N(i)|} \times \prod_s q_s(y_s) \]

From which we can derive

\[ H(P_{p,q}) \approx H_{\text{Bethe}}(P_{p,q}) = \sum_i (1 - |N(i)|)H_i(p_i) + \sum_s H_s(q_s) \]

A linear combination of local entropies:

\[ H_i(p_i) = -\sum_{y_i} p_i(y_i) \log p_i(y_i), \quad H_s(q_s) = -\sum_{y_s} q_s(y_s) \log q_s(y_s) \]

Not concave in general!

Hans Bethe, 1906–2005
Geometric Illustration

If loopy BP converges, it reaches a stationary point of the approximate variational problem \( H^{Bethe}(p, q) \) is non-concave in general \( \Rightarrow \) local minima

André Martins (Priberam/IT)

LP Decoders in NLP http://tiny.cc/lpdnlp 64 / 149
If loopy BP converges, it reaches a stationary point of the approximate variational problem $H_{\text{Bethe}}(P_{p,q})$ is non-concave in general $\Rightarrow$ local minima.
If loopy BP converges, it reaches a stationary point of the approximate variational problem $H_{\text{Bethe}}(P_{p,q})$ is non-concave in general $\Rightarrow$ local minima.
If loopy BP converges, it reaches a stationary point of the approximate variational problem $H_{\text{Bethe}}(p, q)$ is non-concave in general $\Rightarrow$ local minima.
If loopy BP converges, it reaches a stationary point of the approximate variational problem $H_{\text{Bethe}}(P_p, q)$ is non-concave in general $\Rightarrow$ local minima

André Martins (Priberam/IT)

LP Decoders in NLP

http://tiny.cc/lpdnlp
If loopy BP converges, it reaches a stationary point of the approximate variational problem $H_{\text{Bethe}}(p,q)$ is non-concave in general $\Rightarrow$ local minima.
If loopy BP converges, it reaches a stationary point of the approximate variational problem

\[ H_{\text{Bethe}}(P_{p,q}) \]
is non-concave in general \( \Rightarrow \) local minima
Summary of Loopy BP

**Advantages:**

- Simple to implement
- Handles structured and logic factors (only need to compute local marginals)
- Often works well in practice (if cycles are not very influential)
- Often yields a reasonable approximation of log $Z$ and $H$

**Disadvantages:**

- Doesn’t give an upper/lower bound of log $Z$ and $H$
- Entropy approximation is not concave (local minima)
- May not converge
- The final beliefs may not be realizable marginals
Tree Reweighted BP (Wainwright et al., 2005)

**Key idea:** cover the graph with a set of trees
**Key idea:** cover the graph with a set of trees
**Tree Reweighted BP (Wainwright et al., 2005)**

**Key idea:** cover the graph with a set of trees

![Diagram](http://tiny.cc/lpdnlp)

\[
H_{\text{TRBP}}(p, q) = \sum_i \left(1 - \sum_s \in N(i) c_{is}\right) H_i(p_i) + \sum_s H_s(q_s)
\]

(Note: if all \(c_{is} = 1\), this would revert to the Bethe approximation)
Tree Reweighted BP (Wainwright et al., 2005)

**Key idea:** cover the graph with a set of trees

Count the appearance probability $c_{is} > 0$ of each edge

$$H_{TRBP}(P, q) = \sum_i \left( 1 - \sum_s \in N(i) c_{is} \right) H_i(p_i) + \sum_s H_s(q_s)$$

(Note: if all $c_{is} = 1$ this would revert to the Bethe approximation)
Tree Reweighted BP (Wainwright et al., 2005)

**Key idea:** cover the graph with a set of trees

Count the appearance probability $c_{is} > 0$ of each edge

$H_{TRBP}(p, q) = \sum_i (1 - \sum_{s \in N(i)} c_{is}) H_i(p_i) + \sum_s H_s(q_s)$

(Note: if all $c_{is} = 1$ this would revert to the Bethe approximation)
Tree Reweighted BP (Wainwright et al., 2005)

**Key idea:** cover the graph with a set of trees

Count the appearance probability $c_{is} > 0$ of each edge

**This results in a convex upper bound of $-H$ and $\log Z$:**

$$H_{\text{TRBP}}(p, q) = \sum_i (1 - \sum_{s \in N(i)} c_{is}) H_i(p_i) + \sum_s H_s(q_s)$$

(Note: if all $c_{is} = 1$ this would revert to the Bethe approximation)
TRBP Messages

■ Variable-to-factor messages:

\[ m_{i \rightarrow s}(y_i) = \frac{\psi_i(y_i) \prod_{s' \in N(i)} n_{s' \rightarrow i}^{c_{is'}}(y_i)}{n_{s \rightarrow i}(y_i)} \]

■ Factor-to-variable messages:

\[ n_{s \rightarrow i}(y_i) = \sum_{y_s \sim y_i} \frac{\psi_s(y_s) \prod_{j \in N(s)} m_{j \rightarrow s}^{c_{js}}(y_j)}{m_{i \rightarrow s}(y_i)} \]

■ Variable beliefs:

\[ p_i(y_i) \propto \psi_i(y_i) \prod_{s \in N(i)} n_{s \rightarrow i}^{c_{is}}(y_i) \]

■ Factor beliefs:

\[ q_s(y_s) \propto \psi_s(y_s) \prod_{i \in N(s)} m_{i \rightarrow s}^{c_{is}}(y_i) \]
Summary of TRBP

Advantages:

- Still simple to implement
- Entropy approximation is concave (no local minima)
- Gives an upper bound on $-H$ and $\log Z$
- Lots of knobs (the appearance probabilities)

Disadvantages:

- Lots of knobs (the appearance probabilities)
- Typically it’s a very loose bound
- May not converge (but in practice always does, with dampening)
- The final beliefs may not be realizable marginals
### What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th><strong>Algorithm</strong></th>
<th><strong>Local Operation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD³ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>
Norm-Product BP (Hazan and Shashua, 2010)

Subsumes loopy BP and TRBP

Relies on a convex approximation to the entropy using \textit{counting numbers} \( c_i \geq 0 \) and \( c_s > 0 \) (in its simpler variant)

\[
H_{\text{NPBP}}(p, q) = \sum_i c_i H_i(p_i) + \sum_s c_s H_s(q_s)
\]

Messages will become \textit{norms}

Recall the definition of \( \ell_p \)-norm: 
\[
\|x\|_p = \left( \sum_i |x_i|^p \right)^{1/p}
\]
NPBP Messages

- **Variable-to-factor messages:**

\[
m_{i \rightarrow s}(y_i) = \frac{\left(\psi_i(y_i) \prod_{s' \in N(i)} n_{s' \rightarrow i}(y_i)\right)^{c_s/(c_i + \sum_{s' \in N(i)} c'_s)}}{n_{s \rightarrow i}(y_i)}
\]

- **Factor-to-variable messages:**

\[
n_{s \rightarrow i}(y_i) = \left(\sum_{y_s \sim y_i} \left(\psi_s(y_s) \prod_{j \in N(s) \setminus \{i\}} m_{j \rightarrow s}(y_j)\right)^{1/c_s}\right)^{c_s}
\]

- **Variable beliefs:**

\[
p_i(y_i) \propto \left(\psi_i(y_i) \prod_{s \in N(i)} n_{s \rightarrow i}(y_i)\right)^{1/(c_i + \sum_{s' \in N(i)} c'_s)}
\]

- **Factor beliefs:**

\[
q_s(y_s) \propto \left(\psi_s(y_s) \prod_{i \in N(s)} m_{i \rightarrow s}(y_i)\right)^{c_s}
\]
Summary of NPBP

Advantages:

- Still simple to implement
- Entropy approximation is concave (no local minima)
- Always converges (primal-dual block ascent)
- Lots of knobs (the counting numbers)

Disadvantages:

- Lots of knobs (the counting numbers)
- Messages are not computed in parallel (otherwise, may not converge)
- The final beliefs may not be realizable marginals
### What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD³ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>

Diagram:
- $Y_1$, $Y_2$, $Y_3$, $Y_4$, $Y_5$
Outline

1. Structured Prediction and Factor Graphs
2. Integer Linear Programming
3. Message-Passing Algorithms
   - Sum-Product
   - Max-Product
4. Dual Decomposition
5. Applications
6. Conclusions
Define $Z_\epsilon$ where $\epsilon$ is a temperature parameter:

$$Z_\epsilon(\psi, x) = \left( \sum_{y \in \mathcal{Y}(x)} \prod_{i} \psi_i(y_i)^{1/\epsilon} \prod_{s} \psi_s(y_s)^{1/\epsilon} \right)^\epsilon$$

If $\epsilon = 1$, this becomes the partition function $Z(\psi, x)$

If $\epsilon \to 0$, this becomes the mode of $\mathbb{P}_\psi(y|x)$

Note that $Z_\epsilon(\psi, x) = Z(\psi^{1/\epsilon}, x)^\epsilon$ for any $\epsilon$, i.e., $Z_\epsilon$ can be computed by the same means as the partition function by scaling the potentials

By choosing a small enough $\epsilon$, any sum-product message-passing algorithm can be used to approximate the MAP

There is a trade-off between precision and numerical stability
Max-Product Belief Propagation

- For MAP decoding instead of marginal decoding
- Only change: factor-to-variable messages (max instead of $\sum$)

$$n_{s \rightarrow i}(y_i) = \max_{y_s \sim y_i} \left( \psi_s(y_s) \prod_{j \in N(s) \setminus \{i\}} m_{j \rightarrow s}(y_j) \right) = \frac{\max_{y_s \sim y_i} q_s(y_s)}{m_{i \rightarrow s}(y_i)}$$

- If the graph has no cycles, beliefs will converge to max-marginals:

$$p_i(y_i) \rightarrow \max_{y \sim y_i} P_{\psi}(y|x), \quad q_s(y_s) \rightarrow \max_{y \sim y_s} P_{\psi}(y|x)$$

- Decoding the best max-marginal at each variable node gives the MAP
- With cycles: not guaranteed to converge, and even if it does, no relation with LP-MAP
What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD³ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>
TRW-S (Kolmogorov, 2006)

Same rationale as sum-product TRBP: cover the graph with spanning trees, and compute messages using edge appearance probabilities

Only differences:

- Replace $\sum$ with $\max$
- Messages need to be computed *sequentially* for convergence

As max-product loopy BP, all is required is to compute *local max-marginals*

Under mild assumptions, gives the solution of LP-MAP
### What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD³ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>
Max-Product LP (Globerson and Jaakkola, 2008)

Derived by writing the dual of LP-MAP, and solving it with a block coordinate descent algorithm.

The message updates need to be computed in a sequential schedule.

**Progress in the dual objective is monotonic**

**Drawback:** Since the dual is non-smooth, we may get stuck at a suboptimal point.

![Figure 6.3.6.](image)

(From Bertsekas et al. (1999))
MPLP Messages

- **Variable-to-factor messages:**

\[
m_{i \rightarrow s}(y_i) = \psi_i(y_i) \prod_{s' \in N(i) \setminus \{s\}} n_{s' \rightarrow i}(y_i)
\]

- **Factor-to-variable messages:**

\[
n_{s \rightarrow i}(y_i) = \max_{y_s \sim y_i} \left( \prod_{j \in N(s)} m_{j \rightarrow s}(y_j)^{1/|N(s)|} \right)^{1/|N(s)|}
\]

\[
= \left( \prod_{j \in N(s)} m_{j \rightarrow s}(y_j)^{1/|N(s)|} \right)^{1/|N(s)|}
\]
Summary of MPLP

Advantages:

- Very simple to implement
- Handles structured and logic factors (only need to compute local max-marginals)
- Monotonically improves the dual
- No parameters to tune

Disadvantages:

- Can get stuck at a suboptimal solution (general problem with nonsmooth coordinate ascent)
- Messages are not computed in parallel (otherwise, may not converge)
## What Kind of Local Decoding Do We Need?

### Algorithm Local Operation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD$^3$ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>

![Diagram of dependency network with nodes $Y_1$ to $Y_5$]
BP algorithms and their variants can be used both for MAP and marginal decoding.

They need to compute local *marginals* (sum-product) or *max-marginals* (max-product).

Always exact if the graph has no cycles; approximate otherwise.

They correspond to minimizing an energy approximation over the local polytope.

Some variants do convex approximations or compute upper bounds.

Two views of MAP decoding: (1) the near-zero temperature limit of marginal decoding; (2) a non-smooth optimization problem.
Outline

1. Structured Prediction and Factor Graphs
2. Integer Linear Programming
3. Message-Passing Algorithms
   - Sum-Product
   - Max-Product
4. Dual Decomposition
5. Applications
6. Conclusions
Dual Decomposition

- Old idea in optimization (Dantzig and Wolfe, 1960; Everett III, 1963)
- First proposed by Komodakis et al. (2007) in computer vision
- Introduced in NLP by Rush et al. (2010) for model combination
- Successful in syntax, semantics, MT: Koo et al. (2010); Chang and Collins (2011); Martins et al. (2011b); Almeida et al. (2014); Martins and Almeida (2014), and many others.
Dual Decomposition

- Old idea in optimization (Dantzig and Wolfe, 1960; Everett III, 1963)
- First proposed by Komodakis et al. (2007) in computer vision
- Introduced in NLP by Rush et al. (2010) for model combination
- Successful in syntax, semantics, MT: Koo et al. (2010); Chang and Collins (2011); Martins et al. (2011b); Almeida et al. (2014); Martins and Almeida (2014), and many others.
Dual Decomposition

- Old idea in optimization (Dantzig and Wolfe, 1960; Everett III, 1963)
- First proposed by Komodakis et al. (2007) in computer vision
- Introduced in NLP by Rush et al. (2010) for model combination
- Successful in syntax, semantics, MT: Koo et al. (2010); Chang and Collins (2011); Martins et al. (2011b); Almeida et al. (2014); Martins and Almeida (2014), and many others.
Dual Decomposition

- Old idea in optimization (Dantzig and Wolfe, 1960; Everett III, 1963)
- First proposed by Komodakis et al. (2007) in computer vision
- Introduced in NLP by Rush et al. (2010) for model combination
- Successful in syntax, semantics, MT: Koo et al. (2010); Chang and Collins (2011); Martins et al. (2011b); Almeida et al. (2014); Martins and Almeida (2014), and many others.
Recall the LP-MAP problem:

$$\text{maximize } \sum_{i} \theta_i^\top p_i + \sum_{s} \theta_s^\top q_s$$

subject to

$$q_s \in \Delta^{|y_s|}, \forall s$$

$$p_i = M_{is} q_s, \forall i, s.$$ (local polytope)

Matrix $M_{is} \in \{0, 1\}^{|y_i| \times |y_s|}$ represents the constraints $p_i(y_i) = \sum_{y_s \sim y_i} q_s(y_s)$

We’ll reformulate this problem by:

1. Introducing copy variables $q_{is} = p_i$
2. Defining $\theta_{is} := \theta_i/|N(i)|$
Reformulation of LP-MAP

The problem becomes:

$$\text{maximize} \quad \sum_s \left( \theta_s^\top q_s + \sum_{i \in N(s)} \theta_{is}^\top q_{is} \right)$$

subject to

$$q_s \in \Delta^{|y_s|}, \quad \forall s$$

$$q_{is} = M_{is} q_s, \quad \forall i, s \quad \text{(local polytope)}$$

$$q_{is} = p_i, \quad \forall i, s.$$

By introducing Lagrange multipliers for the last constraints, we get the following **Lagrangian function**:

$$\mathcal{L}(p, q, \lambda) = \sum_s \left( \theta_s^\top q_s + \sum_{i \in N(s)} \theta_{is}^\top q_{is} \right) + \sum_{is} \lambda_{is}^\top (p_i - q_{is})$$
Dual of LP-MAP

The dual problem is

$$\text{minimize } \sum_s g_s(\lambda) \quad \text{subject to } \lambda \in \Lambda := \left\{ \lambda \left| \sum_{s \in N(i)} \lambda_is = 0 \right. \right\}$$

where the $g_s(\lambda)$ are local subproblems,

$$g_s(\lambda) := \max_{\bar{q}_s \in Q_s} \left( \theta_s^\top q_s + \sum_{i \in N(s)} (\theta_is + \lambda_is)^\top q_is \right)$$

$$= \max_{y_s \in y_s} \left( \theta_s(y_s) + \sum_{i \in N(s)} (\theta_is(y_i) + \lambda_is(y_i)) \right)$$

and $\bar{q}_s \in Q_s$ encodes the constraints

$$\left\{ \begin{array}{l} q_s \in \Delta_{|y_s|} \\ q_is = M_is q_s, \forall i \in N(s). \end{array} \right.$$
initialize penalties $\lambda$ to zero
repeat

$\bar{q}_s \leftarrow \arg\max_{\bar{q}_s \in Q} \theta_s^\top q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^\top q_{is}$

end for

$p_i \leftarrow 1 / |N(i)| \sum_{s \in N(i)} q_{is}$

$\lambda_{is} \leftarrow \lambda_{is} - \eta (q_{is} - p_i)$

until consensus (all $q_{is} = p_i$) or maximum number of iterations reached

Guaranteed to converge to an $\epsilon$-accurate solution after at most $O(\frac{1}{\epsilon^2})$ iterations

Problem: too slow when there are many factors (Martins et al., 2011b)
initialize penalties $\lambda$ to zero
repeat
  for each factor $s$ do
    $\tilde{q}_s \leftarrow \arg \max_{\bar{q}_s \in Q_s} \theta_s^\top q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^\top q_{is}$
  end for
until consensus (all $q_{is} = p_i$) or maximum number of iterations reached
Projected Subgradient (Komodakis et al., 2007)

initialize penalties $\lambda$ to zero

repeat

for each factor $s$ do

$\tilde{q}_s \leftarrow \arg \max_{\bar{q}_s \in Q_s} \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is}$

end for

$p_i \leftarrow \frac{1}{|N(i)|} \sum_{s \in N(i)} q_{is}$

until consensus (all $q_{is} = p_i$) or maximum number of iterations reached
initialize penalties $\lambda$ to zero
repeat
  for each factor $s$ do
    $\bar{q}_s \leftarrow \arg \max_{q_s \in \Omega_s} \theta_s^\top q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^\top q_{is}$
  end for
  $p_i \leftarrow \frac{1}{|N(i)|} \sum_{s \in N(i)} q_{is}$
  $\lambda_{is} \leftarrow \lambda_{is} - \eta (q_{is} - p_i)$
until consensus (all $q_{is} = p_i$) or maximum number of iterations reached

Guaranteed to converge to an $\epsilon$-accurate solution after at most $O\left(\frac{1}{\epsilon^2}\right)$ iterations

Problem: too slow when there are many factors (Martins et al., 2011b)
Projected Subgradient (Komodakis et al., 2007)

initialize penalties $\lambda$ to zero

repeat

for each factor $s$ do

$\bar{q}_s \leftarrow \arg \max_{\bar{q}_s \in Q_s} \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is}$

end for

$p_i \leftarrow \frac{1}{|N(i)|} \sum_{s \in N(i)} q_{is}$

$\lambda_{is} \leftarrow \lambda_{is} - \eta (q_{is} - p_i)$

until consensus (all $q_{is} = p_i$) or maximum number of iterations reached

- Guaranteed to converge to an $\epsilon$-accurate solution after at most $O(1/\epsilon^2)$ iterations
- **Problem:** too slow when there are many factors (Martins et al., 2011b)
### What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD$^3$ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>
Accelerating Consensus

Two fundamental problems with the subgradient algorithm:

1. The dual objective $\sum_s g_s(\lambda)$ is non-smooth
2. Consensus is promoted only through updating $\lambda$ (no memory about past updates)
Accelerating Consensus

Two fundamental problems with the subgradient algorithm:

1. The dual objective $\sum_s g_s(\lambda)$ is non-smooth
2. Consensus is promoted only through updating $\lambda$ (no memory about past updates)

How can dual decomposition be accelerated?
Accelerating Consensus

Two fundamental problems with the subgradient algorithm:

1. The dual objective $\sum_s g_s(\lambda)$ is non-smooth
2. Consensus is promoted only through updating $\lambda$ (no memory about past updates)

How can dual decomposition be accelerated?

- Jojic et al. (2010) smooth the objective and use gradient methods
- Martins et al. (2011a): augmented Lagrangian
Accelerating Consensus

Two fundamental problems with the subgradient algorithm:

1. The dual objective \( \sum_s g_s(\lambda) \) is non-smooth
2. Consensus is promoted only through updating \( \lambda \) (no memory about past updates)

How can dual decomposition be accelerated?

- Jojic et al. (2010) smooth the objective and use gradient methods
- Martins et al. (2011a): augmented Lagrangian
Accelerated Gradient (Jojic et al., 2010)

Basic idea: make the dual objective smooth by adding an entropic perturbation with a near-zero $\epsilon$ temperature (also Johnson (2008))

The subproblems become local *marginal* computations instead of maximizations

With Nesterov’s accelerated gradient method (Nesterov, 2005), the iteration bound goes from $O(1/\epsilon^2)$ to $O(1/\epsilon)$

However: very sensitive to the temperature parameter

With low temperatures, may face numerical issues (in particular for some hard-constraint factors)

In practice, quite slow to take off (we’ll see some plots later)
Accelerated Gradient (Jojic et al., 2010)

Basic idea: make the dual objective smooth by adding an entropic perturbation with a near-zero $\epsilon$ temperature (also Johnson (2008))

The subproblems become local *marginal* computations instead of maximizations

With Nesterov’s accelerated gradient method (Nesterov, 2005), the iteration bound goes from $O(1/\epsilon^2)$ to $O(1/\epsilon)$

**However: very sensitive to the temperature parameter**

With low temperatures, may face numerical issues (in particular for some hard-constraint factors)

In practice, quite slow to take off (we’ll see some plots later)
### What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD³ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>

**Diagram:**

- $Y_1$, $Y_2$, $Y_3$, $Y_4$, $Y_5$
Accelerating Consensus

Two fundamental problems with the subgradient algorithm:

1. The dual objective \( \sum_s g_s(\lambda) \) is non-smooth
2. Consensus is promoted only through updating \( \lambda \) (no memory about past updates)

How can dual decomposition be accelerated?

- Jojic et al. (2010) smooth the objective and use gradient methods ✓
- Martins et al. (2011a): augmented Lagrangian
Accelerating Consensus

Two fundamental problems with the subgradient algorithm:

1. The dual objective $\sum_s g_s(\lambda)$ is non-smooth
2. Consensus is promoted only through updating $\lambda$ (no memory about past updates)

How can dual decomposition be accelerated?

- Jojic et al. (2010) smooth the objective and use gradient methods ✓
- Martins et al. (2011a): augmented Lagrangian
Alternating Directions Dual Decomposition ($\text{AD}^3$)

Based on the \textbf{alternating direction method of multipliers (ADMM)}:

- an old method in optimization inspired by augmented Lagrangians
  (Gabay and Mercier, 1976; Glowinski and Marroco, 1975)
- a natural fit to consensus problems
- a natural “upgrade” of the subgradient algorithm (Boyd et al., 2011)
Augmented Lagrangian and ADMM

Basic idea: augment the Lagrangian function with a **quadratic penalty**

\[
\mathcal{L}_\eta(p, q, \lambda) = \sum_s \left( \theta_s^\top q_s + \sum_{i \in N(s)} \theta_{is}^\top q_{is} \right) + \sum_{is} \lambda_{is}^\top (p_i - q_{is}) - \frac{\eta}{2} \sum_{is} \|q_{is} - p_i\|^2
\]

Method of multipliers (super-linear convergence):

1. Maximize \(\mathcal{L}_\eta(p, q, \lambda)\) jointly w.r.t. \(p\) and \(q\) (challenging)
2. Multiplier update: \(\lambda_{is} \leftarrow \lambda_{is} - \eta(q_{is} - p_i)\)

**Alternating direction method of multipliers:** replace step 1 by separate maximizations (first w.r.t. \(q\), then \(p\))
initialize penalties $\lambda$ to zero

repeat
  for each factor $s = 1, \ldots, S$ do
    $\bar{q}_s \leftarrow \arg \max_{\bar{q}_s \in Q_s} \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is}$
  end for

  $p_i \leftarrow \frac{1}{|N(i)|} \sum_{s \in N(i)} q_{is}$
  $\lambda_{is} \leftarrow \lambda_{is} - \eta (q_{is} - p_i)$

until consensus (all $q_{is} = p_i$) or maximum number of iterations reached
initialize penalties $\lambda$ to zero

repeat

for each factor $s = 1, \ldots, S$ do

\[ \bar{q}_s \leftarrow \arg \max_{\bar{q}_s \in \Omega_s} \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \|q_{is} - p_i\|^2 \]

end for

\[ p_i \leftarrow \frac{1}{|N(i)|} \sum_{s \in N(i)} q_{is} \]

\[ \lambda_{is} \leftarrow \lambda_{is} - \eta (q_{is} - p_i) \]

until consensus (all $q_{is} = p_i$) or maximum number of iterations reached
initialize penalties $\lambda$ to zero

repeat
  for each factor $s = 1, \ldots, S$ do
    $\bar{q}_s \leftarrow \arg\max_{q_s \in Q_s} \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \|q_{is} - p_i\|^2$
  end for

  $p_i \leftarrow \frac{1}{|N(i)|} \sum_{s \in N(i)} q_{is}$
  $\lambda_{is} \leftarrow \lambda_{is} - \eta (q_{is} - p_i)$

until consensus (all $q_{is} = p_i$) or maximum number of iterations reached

- **faster consensus**: regularize $q$-step towards average votes in $p$
From Subgradient to AD$^3$ (Martins et al., 2011a)

```latex
initialize penalties $\lambda$ to zero
repeat
    for each factor $s = 1, \ldots, S$ do
        $\bar{q}_s \leftarrow \arg \max_{q_s \in Q_s} \theta_s^T q_s + \sum_{i \in N(s)} \left( \theta_{is} + \lambda_{is} \right)^T q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \| q_{is} - p_i \|^2$
    end for

    $p_i \leftarrow \frac{1}{|N(i)|} \sum_{s \in N(i)} q_{is}$
    $\lambda_{is} \leftarrow \lambda_{is} - \eta (q_{is} - p_i)$
until consensus (all $q_{is} = p_i$) or maximum number of iterations reached
```

- **faster consensus**: regularize $q$-step towards average votes in $p$
- **better stopping conditions**: keeps track of primal and dual residuals
Theoretical Guarantees of AD$^3$

**Convergent** in primal and dual (Glowinski and Le Tallec, 1989)

**Iteration bound**: $O(1/\epsilon)$ (cf. $O(1/\epsilon^2)$ for projected subgradient)

**Inexact AD$^3$ subproblems**: still convergent if residuals are summable (Eckstein and Bertsekas, 1992)

**Always dual feasible**: can compute upper bounds and embed in branch-and-bound toward exact decoding (Das et al., 2012)
Theoretical Guarantees of AD$^3$

**Convergent** in primal and dual (Glowinski and Le Tallec, 1989)

**Iteration bound:** $O(1/\epsilon)$ (cf. $O(1/\epsilon^2)$ for projected subgradient)

**Inexact AD$^3$ subproblems:** still convergent if residuals are summable (Eckstein and Bertsekas, 1992)

**Always dual feasible:** can compute upper bounds and embed in branch-and-bound toward exact decoding (Das et al., 2012)

But: AD$^3$ local subproblems are *quadratic* (more involved than in projected subgradient)
Theoretical Guarantees of AD$^3$

**Convergent** in primal and dual (Glowinski and Le Tallec, 1989)

**Iteration bound:** $O(1/\epsilon)$ (cf. $O(1/\epsilon^2)$ for projected subgradient)

**Inexact AD$^3$ subproblems:** still convergent if residuals are summable (Eckstein and Bertsekas, 1992)

**Always dual feasible:** can compute upper bounds and embed in branch-and-bound toward exact decoding (Das et al., 2012)

But: AD$^3$ local subproblems are *quadratic* (more involved than in projected subgradient)

Still—very easy and efficient for logic and knapsack factors!
Martins et al. (2011a): logic factors can be solved in $O(K)$ time

Almeida and Martins (2013): same for knapsack factors!
Structured Factors

What about structured factors?

Projected subgradient handles these quite well with combinatorial machinery (Viterbi, Chu-Liu-Edmonds, Fulkerson-Ford, Floyd-Warshall,...)

We cannot solve the AD subproblems with that machinery...

Or can we?

Active set method: seek the support of the solution by adding/removing components; very suitable for warm-starting (Nocedal and Wright, 1999)

André Martins (Priberam/IT)

LP Decoders in NLP

http://tiny.cc/lpdnlp
Structured Factors

What about structured factors?

**Projected subgradient** handles these quite well

- combinatorial machinery (Viterbi, Chu-Liu-Edmonds, Fulkerson-Ford, Floyd-Warshall,...)

We cannot solve the AD$^3$ subproblems with that machinery...
Structured Factors

What about structured factors?

**Projected subgradient** handles these quite well

- combinatorial machinery (Viterbi, Chu-Liu-Edmonds, Fulkerson-Ford, Floyd-Warshall,...)

We cannot solve the AD\(^3\) subproblems with that machinery...

**Or can we?**
Structured Factors

What about structured factors?

Projected subgradient handles these quite well

- combinatorial machinery (Viterbi, Chu-Liu-Edmonds, Fulkerson-Ford, Floyd-Warshall,...)

We cannot solve the $AD^3$ subproblems with that machinery...

Or can we?

Active set method: seek the support of the solution by adding/removing components; very suitable for warm-starting (Nocedal and Wright, 1999)
An Active Set Method for the AD\textsuperscript{3} Subproblem

\[ \tilde{q}_s \leftarrow \arg \max_{\tilde{q}_s \in Q_s} \left( \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \|q_{is} - p_i\|^2 \right) \]
An Active Set Method for the AD³ Subproblem

\[
\bar{q}_s \leftarrow \arg \max_{\bar{q}_s \in Q_s} \left( \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \|q_{is} - p_i\|^2 \right)
\]

Too many possible assignments: dimension of \( q_s \) is \( O(\exp(|N(s)|)) \)
An Active Set Method for the AD$^3$ Subproblem

\[
\bar{q}_s \leftarrow \arg \max_{\bar{q}_s \in Q_s} \left( \theta_s^\top q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^\top q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \|q_{is} - p_i\|^2 \right)
\]

Too many possible assignments: dimension of $q_s$ is $O(\exp(|N(s)|))$

Key result: there’s a sparse solution (only $O(|N(s)|)$ nonzeros)
An Active Set Method for the AD$^3$ Subproblem

$$\bar{q}_s \leftarrow \arg \max_{\bar{q}_s \in \mathcal{Q}_s} \left( \theta_s^T q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^T q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \| q_{is} - p_i \|^2 \right)$$

Too many possible assignments: dimension of $q_s$ is $O(\exp(|N(s)|))$

Key result: there’s a sparse solution (only $O(|N(s)|)$ nonzeros)

Active set methods: seek the support of the solution by adding/removing components; very suitable for warm-starting (Nocedal and Wright, 1999)

Only requirement: a local-max oracle (as in projected subgradient)
An Active Set Method for the AD$^3$ Subproblem

$$\bar{q}_s \leftarrow \arg \max_{\bar{q}_s \in Q_s} \left( \theta_s^\top q_s + \sum_{i \in N(s)} (\theta_{is} + \lambda_{is})^\top q_{is} - \frac{\eta}{2} \sum_{i \in N(s)} \|q_{is} - p_i\|_2^2 \right)$$

Too many possible assignments: dimension of $q_s$ is $O(\exp(|N(s)|))$

Key result: there’s a sparse solution (only $O(|N(s)|)$ nonzeros)

Active set methods: seek the support of the solution by adding/removing components; very suitable for warm-starting (Nocedal and Wright, 1999)

Only requirement: a local-max oracle (as in projected subgradient)

More info: Martins et al. (2014)
Caching and warm-starting the subproblems reduces drastically the number of oracle calls—huge speed-ups!!

AD$^3$ faster to achieve consensus (due to the quadratic penalty)
### What Kind of Local Decoding Do We Need?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Local Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-Prod. BP (Pearl, 1988)</td>
<td>marginals</td>
</tr>
<tr>
<td>TRBP (Wainwright et al., 2005)</td>
<td>marginals</td>
</tr>
<tr>
<td>Norm-Product BP (Hazan and Shashua, 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>Max-Prod. BP (Pearl, 1988)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>TRW-S (Kolmogorov, 2006)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>MPLP (Globerson and Jaakkola, 2008)</td>
<td>max-marginals</td>
</tr>
<tr>
<td>PSDD (Komodakis et al., 2007)</td>
<td>MAP</td>
</tr>
<tr>
<td>Accelerated DD (Jojic et al., 2010)</td>
<td>marginals</td>
</tr>
<tr>
<td>AD³ (Martins et al., 2011a)</td>
<td>QP/MAP</td>
</tr>
</tbody>
</table>

<figure>
<figcaption>Y_1, Y_2, Y_3, Y_4, Y_5 nodes with interconnections showing the local operations.</figcaption>
</figure>
Example: Potts Grid (20 × 20, 8 states)


Example: Frame-Semantic Parsing

- Embedded in a branch-and-bound procedure for *exact* decoding
- D. Das, A. Martins, N. Smith.
  “An Exact DD Algorithm for Shallow Semantic Parsing with Constraints.”
  *SEM Workshop, 2012.*
Try It Yourself: AD$^3$ Toolkit

- Freely available at: http://www.ark.cs.cmu.edu/AD3
- Implemented in C++, includes a Python wrapper (thanks to Andy Mueller)
- Implements MPLP, PSDD, AD$^3$ for arbitrary factor graphs
- Many built-in factors: logic, knapsack, dense, and some structured factors
- You can implement your own factor (only need to write a local MAP decoder!)
- Toy examples included (parsing, coreference, Potts models)
Summing Up Dual Decomposition

- Dual decomposition is a general optimization technique that splits the dual into several subproblems (one per factor) that must agree on overlaps.
- This can be used to solve LP-MAP.
- We discussed three variants: subgradient (PSDD), accelerated gradient (ADD), and alternating directions (AD$^3$).
- The algorithms are convergent and retrieve the true MAP if the graph has no cycles; they also give certificates when the solution of LP-MAP equals the MAP.
- For PSDD and AD$^3$ only local maximizations are necessary; ADD requires computing marginals.
Outline

1. Structured Prediction and Factor Graphs

2. Integer Linear Programming

3. Message-Passing Algorithms
   - Sum-Product
   - Max-Product

4. Dual Decomposition

5. Applications

6. Conclusions
Applications

We’ll discuss three applications:

- Turbo Parsing
- Compressive Summarization
- Joint Coreference Resolution and Quotation Attribution
Applications

We’ll discuss three applications:

- Turbo Parsing
- Compressive Summarization
- Joint Coreference Resolution and Quotation Attribution
What is a Turbo Parser?

A parser that runs inference in factor graphs, ignoring global effects caused by loops (Martins et al., 2010). Name inspired from turbo decoders (Berrou et al., 1993).

Next: We speed up turbo parsers via AD with active set.
What is a Turbo Parser?

- A parser that runs inference in factor graphs, ignoring global effects caused by loops (Martins et al., 2010)
- name inspired from *turbo* decoders (Berrou et al., 1993)
What is a Turbo Parser?

- A parser that runs inference in factor graphs, ignoring global effects caused by loops (Martins et al., 2010)
- name inspired from *turbo* decoders (Berrou et al., 1993)
- **Next:** we speed up turbo parsers via AD$^3$ w/ active set
An Important Distinction

- A projective tree:

  * Logic plays a minimal role here

- A non-projective tree:

  * We learned a lesson in 1987 about volatility
An Important Distinction

- A projective tree:

![Diagram of a projective tree with arrows indicating the structure of the sentence: "Logic plays a minimal role here."

- A non-projective tree:

![Diagram of a non-projective tree with arrows indicating the structure of the sentence: "We learned a lesson in 1987 about volatility."

This talk: we allow non-projective trees.

Suitable for languages with flexible word order (Dutch, German, Czech,...)
First-Order Scores for Arcs

We learned a lesson in 1987 about volatility.
We learned a lesson in 1987 about volatility.
Second-Order Scores for Grandparents

* We learned a lesson in 1987 about volatility
Scores for Arbitrary Siblings

* We learned a lesson in 1987 about volatility
Scores for Head Bigrams

* We learned a lesson in 1987 about volatility
We learned a lesson in 1987 about volatility

Used by Koo and Collins (2010) for projective parsing.
Third-Order Scores for Tri-siblings

* We learned a lesson in 1987 about volatility

Used by Koo and Collins (2010) for projective parsing.
Decoding

How to deal with all these parts?

- Dynamic programming only available for the *projective* case...
Decoding

How to deal with all these parts?

- Dynamic programming only available for the *projective* case...
- Beyond arc-factored models, non-projective parsing is **NP-hard** (McDonald and Satta, 2007)
- **Need to embrace approximations!**
Approximate Dependency Parsers

<table>
<thead>
<tr>
<th></th>
<th>AF</th>
<th>CS</th>
<th>G</th>
<th>AS</th>
<th>DP</th>
<th>HB</th>
<th>NPA</th>
<th>GS</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>McDonald et al. (2006)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smith et al. (2008)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Martins et al. (2010)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Koo et al. (2010)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Martins et al. (2011)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Martins et al. (2013)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Factor Graph Representation

- Variables nodes for **dependency arcs**, linked to a tree constraint
- Head automata for **consecutive siblings and grandparents** (as in Smith and Eisner (2008); Koo et al. (2010))
- Pairwise factors for **arbitrary siblings** (as Martins et al. (2011b))
- Third-order head automata for **grand-siblings and tri-siblings**
- Sequence model for **head bigrams**
Factor Graph Representation

- Variables nodes for dependency arcs, linked to a tree constraint
- Head automata for consecutive siblings and grandparents (as in Smith and Eisner (2008); Koo et al. (2010))
- Pairwise factors for arbitrary siblings (as Martins et al. (2011b))
- Third-order head automata for grand-siblings and tri-siblings
- Sequence model for head bigrams

We solve the LP-MAP relaxation with AD$^3$. 
### Parsing Accuracies/Runtimes

SOTA accuracies for the largest non-projective datasets (CoNLL-2006 and CoNLL-2008):

<table>
<thead>
<tr>
<th>Language</th>
<th>Year</th>
<th>Authors</th>
<th>Accuracy</th>
<th>Tokens/s</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td></td>
<td>Koo et al. (2011)</td>
<td>92.57</td>
<td>131</td>
<td>92.68</td>
</tr>
<tr>
<td>EN</td>
<td></td>
<td>Martins et al. (2011b)</td>
<td>-</td>
<td>-</td>
<td>785 tiks/sec</td>
</tr>
<tr>
<td>DE</td>
<td></td>
<td>Martins et al. (2011b)</td>
<td>91.89</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DE</td>
<td></td>
<td>Rush &amp; Petrov (2012)</td>
<td>90.8</td>
<td>2,880</td>
<td>-</td>
</tr>
<tr>
<td>DE</td>
<td></td>
<td>Zhang &amp; McDonald (2012)</td>
<td>91.35</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>This work</td>
<td></td>
<td>785 tiks/sec</td>
<td>93.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This work</td>
<td></td>
<td>965 tiks/sec</td>
<td>92.41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NL</td>
<td></td>
<td>Koo et al. (2011)</td>
<td>85.81</td>
<td>121</td>
<td>85.53</td>
</tr>
<tr>
<td>NL</td>
<td></td>
<td>Martins et al. (2011b)</td>
<td>-</td>
<td>-</td>
<td>599 tiks/sec</td>
</tr>
<tr>
<td>CZ</td>
<td></td>
<td>Martins et al. (2010)</td>
<td>88.78</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CZ</td>
<td></td>
<td>Martins et al. (2011b)</td>
<td>89.46</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>This work</td>
<td></td>
<td>599 tiks/sec</td>
<td>86.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This work</td>
<td></td>
<td>501 tiks/sec</td>
<td>90.32</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Extension: Broad-Coverage Semantic Parsing

Same idea applied to **semantic role labeling**.

![Diagram](https://example.com/diagram.png)

Mr. Percival declined to comment.

Best results in the SemEval 2014 shared task:

Applications

We’ll discuss three applications:

- Turbo Parsing
- Compressive Summarization
- Joint Coreference Resolution and Quotation Attribution
Applications

We’ll discuss three applications:

■ Turbo Parsing
■ Compressive Summarization
■ Joint Coreference Resolution and Quotation Attribution
Miguel B. Almeida and André F. T. Martins.
“Fast and Robust Compressive Summarization with Dual Decomposition and Multi-Task Learning.”
ACL 2013.
Multi-Document Summarization

Map a set of related **documents** to a brief **summary**.

**CNN**

** Obama hopes for 'continued progress' in Myanmar **

- Barack Obama met with Nobel Peace Prize winner Aung San Suu Kyi at her home in Myanmar on Monday, praising her "courage and determination" during a historic visit to the once repressive and secretive country.

- The first sitting U.S. president to visit Myanmar, also known as Burma.

- Obama encouraged the country to continue a "remarkable journey".

- He also visits Cambodia to meet the prime minister and attend the East Asia Summit.

**The New York Times**

YANGON, Myanmar — **President Obama** journeyed to this storied tropical outpost of pagodas and jungles on Monday to "extend the hand of friendship" as a land long tormented by repression and poverty begins to throw off military rule and emerge from decades of isolation.

The visit was intended to show support for the reforms put in place by Thein Sein's government since the end of military rule in November 2010. Activists have warned that the visit may be too hasty - political prisoners remain behind bars and ethnic conflicts in border areas are unresolved.

**BBC News**

The country, which is also known as Burma, was ruled by military leaders until early 2011 and for decades was politically and economically cut off from the rest of the world.

Suu Kyi acknowledged that Myanmar’s opening up would be difficult.
Multi-Document Summarization

Map a set of related **documents** to a brief **summary**.

*CNN*

**Obama hopes for 'continued progress' in Myanmar**

(CNN) -- Barack Obama met with Nobel Peace Prize winner Aung San Suu Kyi at her home in Myanmar on Monday, praising her "courage and determination" during a historic visit to the once repressive and secretive country.

The first sitting U.S. president to visit Myanmar, Obama urged its leaders, who have embarked on a series of far-reaching political and economic reforms since 2011, not to extinguish the "flickers of progress that we have seen."

Obama said that his visit to the lakeside villa where the pro-democracy icon spent years under house arrest marked a new chapter between the two countries.

"Here, through so many difficult years, is where she has displayed such unbreakable courage and determination," Obama told reporters, standing next to his fellow Nobel peace laureate. "It is here where she showed that human freedom and human dignity cannot be denied."

The country, which is also known as Burma, was ruled by military leaders until early 2011 and for decades was politically and economically cut off from the rest of the world.

Suu Kyi acknowledged that Myanmar's opening up would be difficult.

*The New York Times*

**YANGON, Myanmar** -- President Obama journeyed to this storied tropical outpost of pagodas and jungles on Monday to "extend the hand of friendship" as a land long tormented by repression and poverty begins to throw off military rule and emerge from decades of isolation.

The visit was intended to show support for the reforms put in place by Thein Sein's government since the end of military rule in November 2010.

Activists have warned that the visit may be too hasty - political prisoners remain behind bars and ethnic conflicts in border areas are unresolved.
What Makes a Good Summary?

**STORY HIGHLIGHTS**

- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar's president
- He's the first sitting U.S. president to visit Myanmar, also known as Burma
- Obama encourages the country to continue a "remarkable journey"
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit
What Makes a Good Summary?

STORY HIGHLIGHTS

- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar's president
- He's the first sitting U.S. president to visit Myanmar, also known as Burma
- Obama encourages the country to continue a "remarkable journey"
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit

1 conciseness
What Makes a Good Summary?

STORY HIGHLIGHTS

- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar's president
- He's the first sitting U.S. president to visit Myanmar, also known as Burma
- Obama encourages the country to continue a "remarkable journey"
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit

1. conciseness
2. informativeness
What Makes a Good Summary?

1. **Conciseness**
2. **Informativeness**
3. **Grammaticality**

**Story Highlights**

- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar's president.
- He's the first sitting U.S. president to visit Myanmar, also known as Burma.
- Obama encourages the country to continue a "remarkable journey".
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit.
Extractive Summarization

Just *extract* the most salient sentences.
Extractive Summarization

Just *extract* the most salient sentences.

**CNN**

**STORY HIGHLIGHTS**

- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar’s president
- He’s the first sitting U.S. president to visit Myanmar, also known as Burma
- Obama encourages the country to continue a “remarkable journey”
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit

(CNN) -- Barack Obama met with Nobel Peace Prize winner Aung San Suu Kyi at her home in Myanmar on Monday, praising her “courage and determination” during a historic visit to the once repressive and secretive country.

The first sitting U.S. president to visit Myanmar, Obama urged its leaders, who have embarked on a series of far-reaching political and economic reforms since 2011, not to extinguish the “flickers of progress that we have seen.”

Obama said that his visit to the lakeside villa where the pro-democracy icon spent years under house arrest marked a new chapter between the two countries.

“Here, through so many difficult years, is where she has displayed such unbreakable courage and determination,” Obama told reporters, standing next to his fellow Nobel peace laureate. “It is here where she showed that human freedom and human dignity cannot be denied.”

The country, which is also known as Burma, was ruled by military leaders until early 2011 and for decades was politically and economically cut off from the rest of the world.

Suu Kyi acknowledged that Myanmar’s opening up would be difficult.

**The New York Times**

YANGON, Myanmar — President Obama journeyed to this storied tropical outpost of pagodas and jungles on Monday to “extend the hand of friendship” as a land long tormented by repression and poverty begins to throw off military rule and emerge from decades of isolation.

The visit was intended to show support for the reforms put in place by Thein Sein's government since the end of military rule in November 2010.

Activists have warned that the visit may be too hasty - political prisoners remain behind bars and ethnic conflicts in border areas are unresolved.

**BBC News**
Just **extract** the most salient sentences.

**CNN**

*Obama hopes for 'continued progress' in Myanmar*

- Obama meets with pro-democracy icon Aung San Suu Kyi and Myanmar’s president.
- He’s the first sitting U.S. president to visit Myanmar, also known as Burma.
- Obama encourages the country to continue a “remarkable journey.”
- He also visits Cambodia to meet the prime minister and attend the East Asia Summit.

*(CNN) — Barack Obama met with Nobel Peace Prize winner Aung San Suu Kyi at her home in Myanmar on Monday, praising her "courage and determination" during a historic visit to the once repressive and secretive country.*

The first sitting U.S. president to visit Myanmar, Obama urged its leaders, who have embarked on a series of far-reaching political and economic reforms since 2011, not to extinguish the “flickers of progress that we have seen.”

Obama said that his visit to the lakeside villa where the pro-democracy icon spent years under house arrest marked a new chapter between the two countries.

"Here, through so many difficult years, is where she has displayed such unbreakable courage and determination," Obama told reporters, standing next to his fellow Nobel peace laureate. "It is here where she showed that human freedom and human dignity cannot be denied."

**The New York Times**

*YANGON, Myanmar — President Obama journeyed to this storied tropical outpost of pagodas and jungles on Monday to "extend the hand of friendship" as a land long tormented by repression and poverty begins to throw off military rule and emerge from decades of isolation.*

The visit was intended to show support for the reforms put in place by Thein Sein’s government since the end of military rule in November 2010.

Activists have warned that the visit may be too hasty - political prisoners remain behind bars and ethnic conflicts in border areas are unresolved.

**BBC News**

*Myanmar Obama visit*
What We Do: Compressive Summarization

Jointly extract and compress sentences.
What We Do: Compressive Summarization

Jointly **extract** and **compress** sentences.

**The New York Times**

YANGON, Myanmar — President Obama journeyed to this storied tropical outpost of pagodas and jungles on Monday to “extend the hand of friendship” as a land long tormented by repression and poverty begins to throw off military rule and emerge from decades of isolation.

The visit was intended to show support for the reforms put in place by Thein Sein’s government since the end of military rule in November 2010.

Activists have warned that the visit may be too hasty — political prisoners remain behind bars and ethnic conflicts in border areas are unresolved.

---

**CNN**

**STORY HIGHLIGHTS**

- Obama meets with pro-democracy icon Aung San Suu Kyi at her home in Myanmar on Monday, praising her “courage and determination” during a historic visit to the once repressive and secretive country.

The first sitting U.S. president to visit Myanmar, Obama urged its leaders, who have embarked on a series of far-reaching political and economic reforms since 2011, not to extinguish the “flickers of progress that we have seen.”

Obama said that his visit to the lakeside villa where the pro-democracy icon spent years under house arrest marked a new chapter between the two countries.

“The country, which is also known as Burma, was ruled by military leaders until early 2011 and for decades was politically and economically cut off from the rest of the world.

Suu Kyi acknowledged that Myanmar’s opening up would be difficult.

---

**BBC NEWS**

Myanmar Obama visit
What We Do: Compressive Summarization

Jointly extract and compress sentences.

For given summary size, easier to be informative, but harder to be grammatical.
Compressive Summarization as Global Optimization

- Indicator variables for every word of the $n$th sentence, $z_n := \langle z_n, \ell \rangle_{\ell=1}^{L_n}$

Summary length must not exceed the budget ($B$ words).

Quality function rewards global informativeness (through $g(z)$)...

...but also local grammaticality (through $h_n(z_n)$):

$$\max g(z) + \sum_{n=1}^{N} h_n(z_n)$$

s.t.

$$\sum_{n=1}^{N} L_n \sum_{\ell=1}^{1} z_{n,\ell} \leq B.$$
Compressive Summarization as Global Optimization

- Indicator variables for every word of the $n$th sentence, $z_n := \langle z_n, \ell \rangle^L_{\ell=1}$
- Summary length must not exceed the budget ($B$ words)
- Quality function rewards global informativeness (through $g(z)$)...
- ... but also local grammaticality (through $h_n(z_n)$):

$$\text{maximize } \sum_{n=1}^{N} g(z) + \sum_{n=1}^{N} h_n(z_n)$$

$$\text{s.t. } \sum_{n=1}^{N} \sum_{\ell=1}^{L_n} z_{n,\ell} \leq B.$$
Grammaticality: Sentence Compression Model

Inspired by Knight and Marcu (2000)’s word deletion model
Grammaticality: Sentence Compression Model

Inspired by Knight and Marcu (2000)’s word deletion model
Our model factors over dependency arcs:

$\text{The leader of moderate Kashmiri separatists warned Thursday that ...}$
Inspired by Knight and Marcu (2000)’s word deletion model

Our model factors over dependency arcs:

Goal: maximize sum of arc scores, allowing only deletion of subtrees.
Grammaticality: Sentence Compression Model

Inspired by Knight and Marcu (2000)’s word deletion model

Our model factors over dependency arcs:

$\text{The leader of moderate Kashmiri separatists warned Thursday that ...}$

**Goal:** maximize sum of arc scores, allowing only *deletion of subtrees.*

*A structured factor, locally decodable with dynamic programming.*
Inspired by *extractive* max-coverage models (Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Gillick et al., 2008; Lin and Bilmes, 2010)
Informativeness: Coverage Model

Inspired by extractive max-coverage models (Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Gillick et al., 2008; Lin and Bilmes, 2010)

- Extract a list of concepts from the original documents
- Define relevance scores for each concept (linear feature-based model)
Informativeness: Coverage Model

Inspired by *extractive* max-coverage models (Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Gillick et al., 2008; Lin and Bilmes, 2010)

- Extract a list of **concepts** from the original documents
- Define **relevance scores** for each concept (linear feature-based model)
- Define $g(z)$ as sum of scores for each **concept** in the summary
Graphical Model for Our Compressive Summarizer

We use dual decomposition (AD3) for solving a linear relaxation. We apply a fast rounding procedure to obtain a valid summary.

Multi-task learning: user-generated data (Simple English Wikipedia) along with manual abstracts and compressive summaries.

Budget
The leader of moderate Kashmiri separatists warned Thursday that... 

Talks with Kashmiri separatists began last year... 

We use dual decomposition (AD3) for solving a linear relaxation.

We apply a fast rounding procedure to obtain a valid summary.

Multi-task learning: user-generated data (Simple English Wikipedia) along with manual abstracts and compressive summaries.

André Martins (Priberam/IT)

LP Decoders in NLP

http://tiny.cc/lpdnlp
The leader of moderate Kashmiri separatists warned Thursday that...

Talks with Kashmiri separatists began last year...

André Martins (Priberam/IT)

LP Decoders in NLP

http://tiny.cc/lpdnlp
The leader of moderate Kashmiri separatists warned Thursday that...

Talks with Kashmiri separatists began last year...

We use dual decomposition (AD^3) for solving a linear relaxation.

We apply a fast rounding procedure to obtain a valid summary.

Multi-task learning: user-generated data (Simple English Wikipedia) along with manual abstracts and compressive summaries.

Budget

Concept tokens

Sentences

André Martins (Priberam/IT)

LP Decoders in NLP

http://tiny.cc/lpdnlp
The leader of moderate Kashmiri separatists warned Thursday that...

Talks with Kashmiri separatists began last year...

"Kashmiri separatists"

Concept tokens

Concept type

Budget

Sentences

We use dual decomposition (AD3) for solving a linear relaxation.

We apply a fast rounding procedure to obtain a valid summary.

Multi-task learning: user-generated data (Simple English Wikipedia) along with manual abstracts and compressive summaries.

André Martins (Priberam/IT)
1. We use dual decomposition (AD$^3$) for solving a linear relaxation
2. We apply a fast rounding procedure to obtain a valid summary
We use dual decomposition (AD³) for solving a linear relaxation.

We apply a fast rounding procedure to obtain a valid summary.

**Multi-task learning:** user-generated data (Simple English Wikipedia) along with manual abstracts and compressive summaries.
Results on TAC-2008 Dataset

- Better informativeness (without sacrificing grammaticality):

<table>
<thead>
<tr>
<th>Method</th>
<th>ROUGE-2 Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gillick et al. (2008)</td>
<td>11.03</td>
</tr>
<tr>
<td>Berg-Kirkpatrick et al. (2011)</td>
<td>11.71</td>
</tr>
<tr>
<td>Woodsend and Lapata (2012)</td>
<td>11.37</td>
</tr>
<tr>
<td>Single-task AD(^3)</td>
<td>11.88</td>
</tr>
<tr>
<td><strong>Multi-task AD(^3)</strong></td>
<td><strong>12.30</strong></td>
</tr>
</tbody>
</table>

- Averaged runtimes per summarization problem (10 documents):

<table>
<thead>
<tr>
<th>Solver</th>
<th>Runtime (sec.)</th>
<th>ROUGE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ILP Exact, GLPK</td>
<td>10.394</td>
<td>12.40</td>
</tr>
<tr>
<td>LP-Relax., GLPK</td>
<td>2.265</td>
<td>12.38</td>
</tr>
<tr>
<td><strong>AD(^3) (1,000 its.)</strong></td>
<td><strong>0.406</strong></td>
<td><strong>12.30</strong></td>
</tr>
<tr>
<td>Extractive (ILP)</td>
<td>0.265</td>
<td>11.16</td>
</tr>
</tbody>
</table>
Applications

We’ll discuss three applications:

- Turbo Parsing
- Compressive Summarization
- Joint Coreference Resolution and Quotation Attribution
Applications

We’ll discuss three applications:

- Turbo Parsing
- Compressive Summarization
- Joint Coreference Resolution and Quotation Attribution
Why Jointly?

Coreference resolution and quotation attribution may benefit from being treated as a joint task.

Rivals carp at “the principle of Pilson,” as NBC’s Arthur Watson once put it – “he’s always expounding that rights are too high, then he’s going crazy.” But the 49-year-old Mr. Pilson is hardly a man to ignore the numbers.

Two consecutive quotes are often from co-referent speakers:

English novelist Dorothy L. Sayers described ringing as a “passion that finds its satisfaction in mathematical completeness and mechanical perfection.”

Ringers, she added, are “filled with the solemn intoxication that comes of intricate ritual faultlessly performed.”
Why Jointly?

Coreference resolution and quotation attribution may benefit from being treated as a joint task.

A speaker doesn’t refer to himself as *he*:

*Rivals carp at “the principle of Pilson,” as NBC’s Arthur Watson once put it — “he’s always expounding that rights are too high, then he’s going crazy.” But the 49-year-old Mr. Pilson is hardly a man to ignore the numbers.*
Why Jointly?

Coreference resolution and quotation attribution may benefit from being treated as a joint task.

A speaker doesn’t refer to himself as he:

*Rivals carp at “the principle of Pilson,” as NBC’s Arthur Watson once put it – “he’s always expounding that rights are too high, then he’s going crazy.” But the 49-year-old Mr. Pilson is hardly a man to ignore the numbers.*
Why Jointly?

Coreference resolution and quotation attribution may benefit from being treated as a joint task.

A speaker doesn’t refer to himself as he:

Rivals carp at “the principle of Pilson,” as NBC’s Arthur Watson once put it – “he’s always expounding that rights are too high, then he’s going crazy.” But the 49-year-old Mr. Pilson is hardly a man to ignore the numbers.

Two consecutive quotes are often from co-referent speakers:

English novelist Dorothy L. Sayers described ringing as a “passion that finds its satisfaction in mathematical completeness and mechanical perfection.”

Ringers, she added, are “filled with the solemn intoxication that comes of intricate ritual faultlessly performed.”
Why Jointly?

Coreference resolution and quotation attribution may benefit from being treated as a joint task.

A speaker doesn’t refer to himself as *he*:

> Rivals carp at “the principle of Pilson,” as NBC’s Arthur Watson once put it – “he’s always expounding that rights are too high, then he’s going crazy.” But the 49-year-old Mr. Pilson is hardly a man to ignore the numbers.

Two consecutive quotes are often from co-referent speakers:

> English novelist Dorothy L. Sayers described ringing as a “passion that finds its satisfaction in mathematical completeness and mechanical perfection.”

> Ringers, she added, are “filled with the solemn intoxication that comes of intricate ritual faultlessly performed.”
Clusters of co-referent mentions (entities) correspond to subtrees coming out from the root node.
Clustering of co-referent mentions (entities) correspond to subtrees coming out from the root node.
From Coreference to Quotation-Coreference Trees (Almeida et al., 2014)

- Include mention nodes and quotation nodes
- Quotation nodes have to be leaves
- Subtrees coming out from the root induce entity clusters along with their quotes: entity-based quotation attribution
From Coreference to Quotation-Coreference Trees (Almeida et al., 2014)

English novelist Dorothy L. Sayers

ML

she

$M_1$

$M_2$

ringing

$M_5$

Ringers

$M_3$

mathematical completeness

$M_4$

mechanical perfection
From Coreference to Quotation-Coreference Trees (Almeida et al., 2014)

[English novelist Dorothy L. Sayers described ringing as a] “passion that finds its satisfaction in mathematical completeness and mechanical perfection.”

[Ringers, she added, are] “filled with the solemn intoxication that comes of intricate ritual faultlessly performed.”
Beyond Arc Scores

The simplest coreference models (e.g., the `SURFACE` model of Durrett and Klein (2013)) are **arc-factored**

- Exact decoding can be performed in a greedy manner
Beyond Arc Scores

The simplest coreference models (e.g., the \texttt{SURFACE} model of Durrett and Klein (2013)) are \textit{arc-factored}

- Exact decoding can be performed in a greedy manner

\textbf{However:} in our approach, an arc factored model would be equivalent to do coreference resolution and quotation attribution \textit{independently}...
Beyond Arc Scores

The simplest coreference models (e.g., the surface model of Durrett and Klein (2013)) are arc-factored

- Exact decoding can be performed in a greedy manner

However: in our approach, an arc factored model would be equivalent to do coreference resolution and quotation attribution independently...

To do things jointly, we add extra scores for:

- A speaker being mentioned inside a quotation
- Consecutive quotes having the same speakers
Beyond Arc Scores

The simplest coreference models (e.g., the \texttt{SURFACE} model of Durrett and Klein (2013)) are \textit{arc-factored}

- Exact decoding can be performed in a greedy manner

\textbf{However:} in our approach, an arc factored model would be equivalent to do coreference resolution and quotation attribution \textit{independently}...

To do things \textit{jointly}, we add extra scores for:

- \textbf{A speaker being mentioned inside a quotation}
- \textbf{Consecutive quotes having the same speakers}

\textit{These scores require knowing if pairs of nodes are in the same subtree.}
Logic Program

- **Arc variables:** each node (except the root) has exactly one parent
  \[
  \sum_{i=0}^{j-1} a_{i\rightarrow j} = 1, \quad \forall j \neq 0
  \]

- **Path variables:** paths propagate through arcs
  \[
  \pi_{i\rightarrow *i} = 1, \quad \forall i, \quad \pi_{i\rightarrow *k} = \bigvee_{i<j\leq k} (a_{i\rightarrow j} \land \pi_{j\rightarrow *k}), \quad \forall i, k
  \]

- **Pair variables:** nodes $k$ and $\ell$ are in the same subtree if they have a common ancestor which is not the root
  \[
  p_{k,\ell} = \bigvee_{i \neq 0} (\pi_{i\rightarrow *k} \land \pi_{i\rightarrow *\ell}), \quad \forall k, \ell.
  \]
Experiments

Datasets:

- WSJ portion of the Ontonotes (597 documents); same splits as CoNLL 2011 shared task
- Quotation annotations of the PARC dataset (Pareti, 2012; O’Keefe et al., 2012)

Coreference evaluation metrics: average between MUC, B³, CEAFₑ

Quotation evaluation metrics:

- **Representative speaker match (RSM):** # matches to representative (non-pronominal) mention of the gold speaker’s entity
- **Entity cluster \( F_1 (ECF_1) \):** \( F_1 \) score between the predicted and gold speaker entity (sets of mentions)
Results

Coreference Resolution:

<table>
<thead>
<tr>
<th></th>
<th>MUC $F_1$</th>
<th>BCUB $F_1$</th>
<th>CEAFE $F_1$</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Durrett and Klein (2013) (SURFACE)</td>
<td><strong>58.87</strong></td>
<td>62.74</td>
<td>45.46</td>
<td>55.7</td>
</tr>
<tr>
<td>QUOTE/Coref independent</td>
<td>57.89</td>
<td>62.50</td>
<td>45.48</td>
<td>55.3</td>
</tr>
<tr>
<td><strong>Joint System</strong></td>
<td><strong>58.78</strong></td>
<td><strong>63.79</strong></td>
<td><strong>45.50</strong></td>
<td><strong>56.0</strong></td>
</tr>
</tbody>
</table>

Quotation attribution:

<table>
<thead>
<tr>
<th></th>
<th>RSM</th>
<th>ECF$_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUOTEOnly</td>
<td>49.4%</td>
<td>41.2%</td>
</tr>
<tr>
<td>QUOTECOREF</td>
<td>64.6%</td>
<td>70.0%</td>
</tr>
<tr>
<td>QUOTE/Coref independent</td>
<td>74.7%</td>
<td>73.7%</td>
</tr>
<tr>
<td><strong>Joint System</strong></td>
<td><strong>76.6%</strong></td>
<td><strong>74.1%</strong></td>
</tr>
</tbody>
</table>
Outline

1. Structured Prediction and Factor Graphs
2. Integer Linear Programming
3. Message-Passing Algorithms
   - Sum-Product
   - Max-Product
4. Dual Decomposition
5. Applications
6. Conclusions
Conclusions

- Many structured problems in NLP are NP-hard or expensive (constrained models, diversity, combination of structured models)
- Often they can be approximately decoded via Linear Programming (e.g., by relaxing an ILP)
- The structure inherent to these problems can be represented with a factor graph
- Message-passing and dual decomposition algorithms can solve these LPs efficiently, exploiting the structure of the graph
- Conceptually: approximate global decoding by invoking only local decoders (local maximizations, marginals, max-marginals, QPs, ...)
- AD$^3$ is faster than the subgradient algorithm both in theory and in practice, and requires the same local decoders
- SOTA results in several applications (turbo parsing, summarization, joint coref and quotation attribution)
Thank you!

The syntactic/semantic parser and AD³ are freely available at:

http://www.ark.cs.cmu.edu/TurboParser
http://www.ark.cs.cmu.edu/AD3
Acknowledgments

- Fundação para a Ciência e Tecnologia and Information and Communication Technologies Institute (Portugal/USA), through the CMU-Portugal Program.
- Priberam: QREN/POR Lisboa (Portugal), EU/FEDER programme, Discooperio project, contract 2011/18501.
- Priberam: QREN/POR Lisboa (Portugal), EU/FEDER programme, Intelligo project, contract 2012/24803.


References IV


References V


