Inference and Representation

Rachel Hodos

New York University

Lab 12, December 9, 2015

Outline

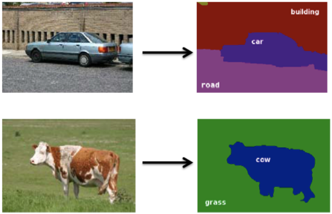
Structured prediction

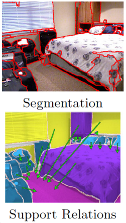
- Problem: given input $\mathbf{x}$, predict *structured* output $\mathbf{y}$, i.e.

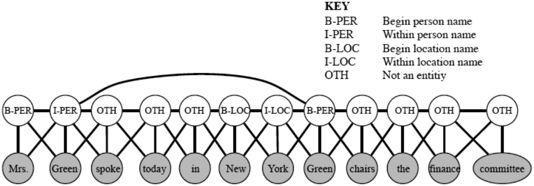
$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{Y}} p(\mathbf{y}|\mathbf{x})$$

- For example, $\mathbf{y}$ could be:
  - a *sequence* of labels on a sentence
  - a 2D *grid* of values estimating the depth of the image
  - a *tree* of dependencies over words in a sentence
- Naive approach: treat as multi-class classification (for $\mathbf{y}$ discrete) or multivariate regression ($\mathbf{y}$ cts.)
  - Impractical: $|\mathcal{Y}|$ could be huge!
- However, we can succeed by **modeling the relationships between the y variables.**
- How? CRF's.

Examples of CRFs







Parametrizing CRFs

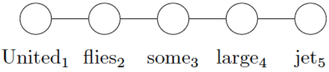
- We've talked before about how we might manually *define* CRF parameters to encode some heuristic, e.g.:
  - *"Neighboring pixels usually have the same label."*
  - *"Nouns often come before verbs."*
- Now, we are addressing how to *learn* the parameters

Example: parameterizing POS tagging model

given:

- a sentence of length n and a tag set $\mathcal{T}$
- one variable for each word, takes values in $\mathcal{T}$
- edge potentials $\theta(i-1, i, t', t)$ for all $i \in n$, $t, t' \in \mathcal{T}$

example:



$$\mathcal{T} = \{A, D, N, V\}$$

Details on chalkboard...

How to learn the parameter vector **w**?

We saw yesterday that MLE is impractical due to Z:

$$\begin{aligned}
\mathbf{w}^{ML} &= \arg\min_{\mathbf{w}} \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} -\log p(\mathbf{y} \mid \mathbf{x}; \mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \left(\sum_{c} \log \phi_c(\mathbf{x}, \mathbf{y}_c; \mathbf{w}) - \log Z(\mathbf{x}; \mathbf{w}) \right) \\
&= \arg\max_{\mathbf{w}} \mathbf{w} \cdot \left(\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \sum_{c} \mathbf{f}_c(\mathbf{x}, \mathbf{y}_c) \right) - \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \log Z(\mathbf{x}; \mathbf{w})
\end{aligned}$$

From MLE to minimizing classification error

From yesterday:

- Consider the empirical risk for 0-1 loss (classification error):
$$\frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \mathbb{1}\{\ \exists \mathbf{y}' \neq \mathbf{y} \text{ s.t. } \hat{p}(\mathbf{y}'|\mathbf{x}) \geq \hat{p}(\mathbf{y}|\mathbf{x})\ \}$$
- Each constraint $\hat{p}(\mathbf{y}'|\mathbf{x}) \geq \hat{p}(\mathbf{y}|\mathbf{x})$ is equivalent to
$$\mathbf{w}\cdot\sum_c \mathbf{f}_c(\mathbf{x},\mathbf{y}'_c) - \log Z(\mathbf{x};\mathbf{w}) \geq \mathbf{w}\cdot\sum_c \mathbf{f}_c(\mathbf{x},\mathbf{y}_c) - \log Z(\mathbf{x};\mathbf{w})$$
- The log-partition function cancels out on both sides. Re-arranging, we have:
$$\mathbf{w}\cdot\left(\sum_c \mathbf{f}_c(\mathbf{x},\mathbf{y}'_c) - \sum_c \mathbf{f}_c(\mathbf{x},\mathbf{y}_c)\right) \geq 0$$
- Said differently, the empirical risk is **zero** when $\forall(\mathbf{x},\mathbf{y})\in\mathcal{D}$ and $\mathbf{y}'\neq\mathbf{y}$,
$$\mathbf{w}\cdot\left(\sum_c \mathbf{f}_c(\mathbf{x},\mathbf{y}_c) - \sum_c \mathbf{f}_c(\mathbf{x},\mathbf{y}'_c)\right) > 0.$$

From MLE to minimizing classification error

- **Gain**: partition function cancels out...
- **New challenge**: large # of constraints!

Algorithms for structured prediction

- Structured perceptron
- Structured SVM:
  - All constraints satisfied
  - With slack
  - With margin scaling