CS 6347

Lecture 18

Alternatives to MLE

Alternatives to MLE

- Exact MLE estimation is intractable
  - To compute the gradient of the log-likelihood, we need to compute the marginal of the model
- Alternatives include
  - Pseudolikelihood approximation to the MLE problem that relies on computing only local probabilities
  - For structured prediction problems, we could avoid likelihoods entirely by minimizing a loss function that measures our prediction error

Pseudolikelihood

- Consider a log-linear MRF $p(x|\theta) = \frac{1}{Z(\theta)} \prod_C \exp\langle\theta, f_C(x_C)\rangle$

- By the chain rule, the joint distribution factorizes as

$$p(x|\theta) = \prod_i p(x_i | x_1, \ldots, x_{i-1}, \theta)$$

- This quantity can be approximated by conditioning on all of the other variables

$$p(x|\theta) \approx \prod_i p(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, \theta)$$

Pseudolikelihood

- Using the independence relations from the MRF

$$p(x|\theta) \approx \prod_i p(x_i|x_{N(i)}, \theta)$$

- Only requires computing local probability distributions (typically much easier)
  - Does not require knowing $Z(\theta)$

UTD

Pseudolikelihood

- For samples $x^1, \ldots, x^M$

$$\log \ell_{PL}(\theta) = \sum_m \sum_i \log p(x_i^m | x_{N(i)}^m, \theta)$$

- This approximation is called the pseudolikelihood
  - If the data is generated from a model of this form, then in the limit of infinite data, maximizing the pseudolikelihood recovers the true model parameters
  - Can be much more efficient to compute than the log likelihood

Pseudolikelihood

$$
\begin{aligned}
\log \ell_{PL}(\theta) &= \sum_m \sum_i \log p(x_i^m | x_{N(i)}^m, \theta) \\
&= \sum_m \sum_i \log \frac{p\left(x_i^m, x_{N(i)}^m \middle| \theta\right)}{\sum_{x_i'} p\left(x_i', x_{N(i)}^m \middle| \theta\right)} \\
&= \sum_m \sum_i \left[\log p\left(x_i^m, x_{N(i)}^m \middle| \theta\right) - \log \sum_{x_i'} p\left(x_i', x_{N(i)}^m \middle| \theta\right) \right] \\
&= \sum_m \sum_i \left[\left\langle \theta, \sum_{C \supset i} f_C(x_C^m) \right\rangle - \log \sum_{x_i'} \exp \left\langle \theta, \sum_{C \supset i} f_C(x_i', x_{C \setminus i}^m) \right\rangle \right]
\end{aligned}
$$

Pseudolikelihood

$$\log \ell_{PL}(\theta) = \sum_m \sum_i \log p(x_i^m | x_{N(i)}^m, \theta)$$

$$= \sum_m \sum_i \log \frac{p\left(x_i^m, x_{N(i)}^m \middle| \theta\right)}{\sum_{x_i'} p\left(x_i', x_{N(i)}^m \middle| \theta\right)}$$

$$= \sum_m \sum_i \left[\log p\left(x_i^m, x_{N(i)}^m \middle| \theta\right) - \log \sum_{x_i'} p\left(x_i', x_{N(i)}^m \middle| \theta\right)\right]$$

$$= \sum_m \sum_i \left[\left\langle \theta, \sum_{C \supset i} f_C(x_C^m) \right\rangle - \log \sum_{x_i'} \exp \left\langle \theta, \sum_{C \supset i} f_C(x_i', x_{C \setminus i}^m) \right\rangle\right]$$

Only involves summing over x_i!

Pseudolikelihood

$$\log \ell_{PL}(\theta) = \sum_m \sum_i \log p(x_i^m | x_{N(i)}^m, \theta)$$

$$= \sum_m \sum_i \log \frac{p\left(x_i^m, x_{N(i)}^m \middle| \theta\right)}{\sum_{x_i'} p\left(x_i', x_{N(i)}^m \middle| \theta\right)}$$

$$= \sum_m \sum_i \left[\log p\left(x_i^m, x_{N(i)}^m \middle| \theta\right) - \log \sum_{x_i'} p\left(x_i', x_{N(i)}^m \middle| \theta\right)\right]$$

$$= \sum_m \sum_i \left[\left\langle \theta, \sum_{C \supset i} f_C(x_C^m) \right\rangle - \log \sum_{x_i'} \exp \left\langle \theta, \sum_{C \supset i} f_C(x_i', x_{C \setminus i}^m) \right\rangle\right]$$

Concave in θ!

Consistency of Pseudolikelihood

- Pseudolikelihood is a consistent estimator
  - That is, in the limit of large data, it is exact if the true model belongs to the family of distributions being modeled

$$\nabla_\theta \ell_{PL} = \sum_m \sum_i \left[\sum_{C \supset i} f_C(x_C^m) - \frac{\sum_{x_i'} \exp\langle \theta, \sum_{C \supset i} f_C\left(x_i', x_{C\setminus i}^m\right) \rangle \sum_{C \supset i} f_C\left(x_i', x_{C\setminus i}^m\right)}{\sum_{x_i'} \exp\langle \theta, \sum_{C \supset i} f_C\left(x_i', x_{C\setminus i}^m\right) \rangle} \right]$$

$$= \sum_m \sum_i \left[\sum_{C \supset i} f_C(x_C^m) - \sum_{x_i'} p(x_i' | x_{N(i)}^m, \theta) \sum_{C \supset i} f_C\left(x_i', x_{C\setminus i}^m\right) \right]$$

Can check that the gradient is zero in the limit of large data if $\theta = \theta^*$

Structured Prediction

- Suppose we have a CRF, $p(x|y,\theta) = \frac{1}{Z(\theta,y)} \prod_C \exp(\langle \theta, f_C(x_C, y) \rangle)$
- If goal is to compute $\operatorname{argmax}_x p(x|y)$, then MLE may be overkill
  - We only care about classification error, not about learning the correct marginal distributions as well
- Recall that the classification error is simply the expected number of incorrect predictions made by the learned model on samples from the true distribution
- Instead of maximizing the likelihood, we can minimize the classification error over the training set

Structured Prediction

- For samples $(x^1, y^1), \dots, (x^M, y^M)$, the (unnormalized) classification error is

$$\sum_m 1_{\{x^m \in \operatorname{argmax}_x p(x|y^m,\theta)\}}$$

- The classification error is zero when $p(x^m|y^m, \theta) \geq p(x|y^m, \theta)$ for all x and m or equivalently

$$\left\langle \theta, \sum_C f_C(x_C^m, y^m) \right\rangle \geq \left\langle \theta, \sum_C f_C(x_C, y^m) \right\rangle$$

Structured Prediction

- In the exact case, this can be thought of as having a linear constraint for each possible x and each $y^1, \ldots, y^M$

$$\left\langle \theta, \sum_C [f_C(x_C^m, y^m) - f_C(x_C, y^m)] \right\rangle \geq 0$$

- Any θ that simultaneously satisfies each of these constraints will guarantee that the classification error is zero
  - As there are exponentially many constraints, finding such a θ (if one even exists) is still a challenging problem
  - If such a θ exists, we say that the problem is separable

Structured Perceptron Algorithm

- In the separable case, a straightforward algorithm can be designed to for this task
- Choose an initial θ
- Iterate until convergence
  - For each m
    - Choose $x' \in \operatorname{argmax}_x p(x|y^m, \theta)$
    - Set $\theta = \theta + \sum_C [f_C(x_C^m, y^m) - f_C(x'_C, y^m)]$

Other Alternatives

- Piecewise likelihood uses the observation that $Z(\theta)$ is a convex function of θ

$$Z\left(\sum_T \alpha_T \theta_T\right) \leq \sum_T \alpha_T Z(\theta_T)$$

  – If $Z(\theta_T)$ corresponds to a tree-structured distribution, then the upper bound can be computed in polynomial time

  – To do learning, we minimize the upper bound over $\theta_1, \ldots, \theta_T$

  – Instead of using arbitrary T, the piecewise likelihood constructs an upper bound on the objective function by summing over $\theta|_C$ obtained by zeroing out all components of θ except for those over the clique C (not always possible)