Discriminative Unsupervised Learning of Structured Predictors

Linli Xu, Dana Wilkinson, Finnegan Southey, Dale Schuurmans

Presented by Kent Sutherland & Mark Verheggen

March 2, 2010

Outline

- Unsupervised Hidden Markov Models
- Unsupervised max-margin training
- Unsupervised M3N training
- Approximations

Hidden Markov Models

- Set of states, initial state, and transitions
- Generative model
- Models joint probability
- Easy to train given complete training data

Unsupervised Training

- Typically use EM when there are no labels
- But:
  - Not guaranteed to find a global solution
  - Can't be used in a discriminative approach

Unsupervised SVM

- Optimize the standard SVM objective over all class labelings
- For two classes:
  - $\min_{\mathbf{w},\mathbf{y}} \frac{1}{2}\|\mathbf{w}\|^2 + \sum_i \left[1 - y_i\phi(x_i)^T\mathbf{w}\right]_+$
- This approach has (at least) three issues.

Issue 1: Degenerate Solutions

- All points might be assigned to a single class
- Correction: add a class-balance constraint
- Forces a roughly equal proportion of labels
- For two classes:
  - $-\epsilon \leq \mathbf{y}^T\mathbf{e} \leq \epsilon$

Issue 2: NP-Hard Problem

- There are exponentially many possible **y**.
- But, look at the dual SVM objective:
  - $\max_{0 \le \lambda \le 1} \lambda^T \mathbf{e} - \frac{1}{2\beta} \left\langle K \circ \lambda\lambda^T, \mathbf{y}\mathbf{y}^T \right\rangle$
- **y** only occurs in the term $\mathbf{y}\mathbf{y}^T$.

NP-Hard
(continued)

- Let $M := \mathbf{y}\mathbf{y}^T$. Then $M_{ij} = y_i y_j \in \{-1, 1\}$.
- That is, M_{ij} indicates whether $y_i = y_j$.
- Iff M is an equivalence relation, the following hold:
  - $\mathrm{diag}(M) = \mathbf{e} \quad (y_i = y_i)$
  - $M = M^T \quad (y_i = y_j \Longleftrightarrow y_j = y_i)$
  - $M \succeq 0 \quad (y_i = y_j, y_j = y_k \Longrightarrow y_i = y_k)$

NP-Hard
(continued)

- Optimize over M instead of **y**
- Relax the integer constraints on M so that $M_{ij} \in [-1, 1]$
- Add the constraints $M \succeq 0$, $\mathrm{diag}(M) = \mathbf{e}$
- Result:

$$\min_{M \succeq 0, \mathrm{diag(M)} = \mathbf{e}} \left(\max_{0 \le \lambda \le 1} \lambda^T \mathbf{e} - \frac{1}{2\beta} \left\langle K \circ \lambda\lambda^T, M \right\rangle \right)$$

NP-Hard
(continued)

- Re-written as a semidefinte program:

$$\min_{M, \delta, \mu \ge 0, \nu \ge 0} \delta \quad \text{subject to}$$
$$\begin{bmatrix} M \circ K & \mathbf{e} + \mu - \nu \\ (\mathbf{e} + \mu - \nu)^T & \frac{2}{\beta}(\delta - \nu^T \mathbf{e}) \end{bmatrix} \succeq 0$$
$$\mathrm{diag(M)} = \mathbf{e}, \ \ M \succeq 0, \ -\epsilon\mathbf{e} \le M\mathbf{e} \le \epsilon\mathbf{e}$$

Formulation for Max-Margin Markov Networks

- The same idea, but applied to M3N. Messier.
- Key differences:
  - Class labels **y** replaced with indicator matrices.
  - Two sets of labels (states, transitions)

Initial Experiment

- Proof of concept
- 4 toy datasets, 2 simplified datasets
- New model significantly outperforms EM

Data set	CDHMM	EM
synth. data1 (95%)	3.38 ±0.75	15.09 ±1.92
synth. data2 (90%)	8.12 ±1.57	17.49 ±1.81
synth. data3 (80%)	22.12 ±1.40	30.06 ±1.24
synth. data4 (70%)	31.50 ±1.46	39.90 ±0.86
protein data1	51.75 ±1.80	58.11 ±0.47
protein data2	50.38 ±2.04	57.23 ±0.39

Approximations

- Semidefinite programming is too slow
- Reformulate problem
- Alternate between optimizing M and λ, ξ
- Still uses a semidefinite program to find M:

$$\min_{M} \min_{0 \le \lambda \le 1, \xi \ge 0} \omega(M; \lambda, \xi) = \lambda^T (K \circ M) \lambda / 2\beta + \xi^T \mathbf{e}$$

$$\text{subject to convex constraints}$$

Approximation

(continued)

- Iteratively retrain using the SVM:
  - Initialize labeling
  - Traing SVM
  - Label data with new discriminant
  - Retrain SVM using relabeled data

Approximation Results

- Intuitively similar approach to EM
- Approximation scales to larger datasets
- Still outperforms EM

Table 3. Prediction error for larger data sets.

DATA SET	ACDHMM	EM
20×2-SEQ	43.12 ±2.20	46.27 ±1.51
10×5-SEQ	44.33 ±2.30	48.67 ±1.51
5×10-SEQ	46.44 ±2.12	48.67 ±1.82

Questions?