Structure Learning in UGMs

Sharan Vaswani

28th August, 2015

Conditional Random Fields - Review

Likelihood function:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_i \phi_i(y_i, \mathbf{x}) \prod_{ij} \phi_{ij}(y_i, y_j, \mathbf{x}) \tag{1}$$

Log Linear Assumption:

$$\phi_i(y_i, \mathbf{x}) = \big(\exp(\mathbf{v}_{i,1}^T \mathbf{x}_i), \exp(\mathbf{v}_{i,2}^T \mathbf{x}_i) \big) \tag{2}$$

$$\phi_{ij}(y_i, y_j, \mathbf{x}) = \begin{bmatrix} \exp(\mathbf{w}_{ij,11}^T \mathbf{x}_{ij}) & \exp(\mathbf{w}_{ij,12}^T \mathbf{x}_{ij}) \\ \exp(\mathbf{w}_{ij,21}^T \mathbf{x}_{ij}) & \exp(\mathbf{w}_{ij,22}^T \mathbf{x}_{ij}) \end{bmatrix} \tag{3}$$

Special Cases:

- If $x_{ij} = 1$, we recover an MRF.
- If $\mathbf{w}_{ij,11} = \mathbf{w}_{ij,22} = w$ and $\mathbf{w}_{ij,12} = \mathbf{w}_{ij,21} = -w$, we recover an Ising model.

Conditional Random Fields - Review

Let $\theta = [v, w]$. Negative Log-Likelihood is given by:

$$NLL(\theta) = \sum_{n=1}^{N} -\theta^T F(\mathbf{x}_n, \mathbf{y}_n) + \sum_{n=1}^{N} logZ(\theta, \mathbf{x}_n) \tag{4}$$

NLL is convex with the gradient given by:

$$\nabla_\theta NLL = -\sum_n [F(\mathbf{x}_n, \mathbf{y}_n) - \mathbb{E}_{\mathbf{y}'} F(\mathbf{x}_n, \mathbf{y}')] \tag{5}$$

Structure Learning

Assumes that the structure of the CRF is known or decided manually.
Can we learn the structure as well ?
Methods for structure learning:

- Iterative edge addition / removal
- Restrict to chordal graphs. Ensure efficient parameter estimation [Whi90].
- Search in the space of possible graph structures with bounded treewidth [BJ01].
- Use submodular optimization to discover conditional independences and learn a bounded tree-width network [NB04].

[Whi90]: Graphical models in applied multivariate analysis
[BJ01]:Thin junction trees
[NB04]:PAC-learning bounded tree-width graphical models

Methods for structure learning:

- Graph cuts - recursively partition the nodes to learn multiple bounded treewidth networks [SG09].
- Restrict to learning networks of bounded degree. [KF09].
- Use L1 regularization and use approximate inference. [LGK06].

[SG09]:Learning thin junction trees via graph cuts
[KF09]:Probabilistic graphical models: principles and techniques
[LGK06]:Efficient Structure Learning of Markov Networks using L1-Regularization

Learning through Block-L1 regularization

Use block sparsity on all edge parameters [SMFR]

$$J(\theta) = NLL(\theta) + \lambda_1 ||\mathbf{v}||_2^2 + \lambda_2 R(\mathbf{w}) \tag{6}$$

$$R(\mathbf{w}) = \sum_b ||\mathbf{w}_b||_\alpha \tag{7}$$

Group Lasso: Use $\alpha = 2$ to enforce all parameters in the block (one for each edge) to go to zero.
Can also use $\alpha = \infty$ to enforce block sparsity.

[SMFR]:Structure learning in random fields for heart motion abnormality detection

Learning through Block-L1 regularization

For minimizing equation 6 by an iterative method, need to calculate NLL(θ) each time.
Each gradient computation depends on the graph structure (which is what we are learning). Time complexity $= \mathcal{O}(k^w)$ where k is size of the state space and w is the tree width. $w \leq d$ (number of nodes)
Possible Solutions:

- Use approximate inference or Gibbs sampling.
- Change the objective function to pseudo-likelihood.

Pseudo Likelihood

Let n_i be the neighbours of i (Markov Blanket) in the graph. Pseudo likelihood [Bes77] is defined as:

$$PL(\mathbf{y}_n|\mathbf{x}_n) = \prod_i p(y_i^n|\mathbf{y}_{n_i}, \mathbf{x}^n) \tag{8}$$

$$p(y_i^n|\mathbf{y}_{n_i}, \mathbf{x}^n) = \exp(\theta_i^T \mathbf{F}_i)(\mathbf{x}, \mathbf{y})/Z_i \tag{9}$$

PL is a consistent estimator and convex !
Can be calculated in $\mathcal{O}(d)$.

[Bes77]:Efficiency of pseudolikelihood estimation for simple Gaussian fields

Homework

Read Chapter 3 of Graphical Models, Exponential Families, and Variational Inference by Wainwright, Jordan.

Questions ?

References

- Julian Besag, *Efficiency of pseudolikelihood estimation for simple gaussian fields*, Biometrika (1977), 616–618.
- Francis R Bach and Michael I Jordan, *Thin junction trees*, Advances in Neural Information Processing Systems, 2001, pp. 569–576.
- Daphne Koller and Nir Friedman, *Probabilistic graphical models: principles and techniques*, MIT press, 2009.
- Su-In Lee, Varun Ganapathi, and Daphne Koller, *Efficient structure learning of markov networks using l_1-regularization*, Advances in neural Information processing systems, 2006, pp. 817–824.
- Mukund Narasimhan and Jeff Bilmes, *Pac-learning bounded tree-width graphical models*, Proceedings of the 20th conference on Uncertainty in artificial intelligence, AUAI Press, 2004, pp. 410–417.
- Dafna Shahaf and Carlos Guestrin, *Learning thin junction trees via graph cuts*, International Conference on Artificial Intelligence and Statistics, 2009, pp. 113–120.