Parameter learning in CRF's

Varun Gulshan

June 01, 2009

Structured output learning

We wish to learn a discriminant (or compatability) function:

$$F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R} \tag{1}$$

where $\mathcal{X}$ is the space of inputs and $\mathcal{Y}$ is the space of outputs. The space of functions F is of the form:

$$F(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) \tag{2}$$

The parameter **w** is the parameter to be learned. It is learned by optimizing a certain Regularized Risk functional over the training data.

Notation

Setting the notation:

- $\mathbf{x}$ denotes an element from the input space $\mathcal{X}$.
- $\mathbf{y}$ denotes an element from the output space $\mathcal{Y}$.
- $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \cdots \times \mathcal{Y}_L$, where each $\mathcal{Y}_l = \{1, 2, ..., q\}$. So $|\mathcal{Y}| = q^L$.
- $\phi(\mathbf{x}, \mathbf{y})$ is a feature mapping for the joint space of inputs and outputs
- Training data $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{N}$. n is used to index over training data.

Probablistic model

The compatibility function $F(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$ can be written in a probabilistic interpretation as:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \frac{\exp\big(\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})\big)}{\sum_{\mathbf{y}} \exp\big(\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})\big)} \tag{3}$$

where

$$Z(\mathbf{x}, \mathbf{w}) = \sum_{\mathbf{y}} \exp\big(\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})\big) \tag{4}$$

is the partition function.

ML learning

Maximum likelihood learning involves maximizing:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \prod_{n=1}^{N} p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{w}) \tag{5}$$

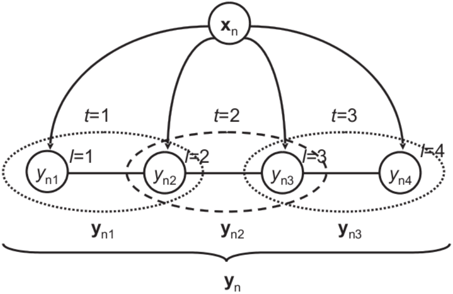
equivalently written as:

$$\begin{aligned} \mathbf{w}^* &= \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \log\big(p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{w})\big) && (6) \\ &= \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \mathbf{w}^T \phi(\mathbf{x}_n, \mathbf{y}_n) - \log\big(Z(\mathbf{x}_n, \mathbf{w})\big) && (7) \end{aligned}$$

ML learning is intractable in general because of the partition function evaluation.

CRF's

CRF's induce a factorization of the probability $p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{w})$, which can make the partition function computation tractable.



$$p(\mathbf{y}_n|\mathbf{x}_n, \mathbf{w}) \propto \exp\left(\sum_{t=1}^{T} \mathbf{w}_t^T \phi_t(\mathbf{x}_n, \mathbf{y}_{nt})\right) \tag{8}$$

The index t is used to index over the cliques in the above equation.

ML Learning in CRF's

For certain factorizations (like chains or tree's), the partition function can be computed efficiently and exactly (using the sum-product algorithm). Thus ML learning is feasible for such CRF's. Recalling the ML maximization criteria:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{n=1}^{N} \Big[\mathbf{w}^T \phi(\mathbf{x}_n, \mathbf{y}_n) - \log\big(Z(\mathbf{x}_n, \mathbf{w})\big)\Big] \quad (9)$$

$$= \arg\min_{\mathbf{w}} \sum_{n=1}^{N} \Big[\log\big(Z(\mathbf{x}_n, \mathbf{w})\big) - \mathbf{w}^T \phi(\mathbf{x}_n, \mathbf{y}_n)\Big] \quad (10)$$

$$\mathbf{w}^T \phi(\mathbf{x}_n, \mathbf{y}_n) = \sum_{t=1}^{T} \mathbf{w}_t^T \phi_t(\mathbf{x}_n, \mathbf{y}_{nt}) \quad (11)$$

$$Z(\mathbf{x}_n, \mathbf{w}) = \sum_{y_{n1}} \sum_{y_{n2}} .. \sum_{y_{nL}} \prod_{t=1}^{T} \exp\big(\mathbf{w}_t^T \phi_t(\mathbf{x}_n, \mathbf{y}_{nt})\big) \quad (12)$$

MAP learning in CRF's

MAP learning just adds a prior to the ML learning and is written as:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\{ \|\mathbf{w}\|^2 + C \sum_{n=1}^{N} \left[\log\big(Z(\mathbf{x}_n, \mathbf{w})\big) - \mathbf{w}^T \phi(\mathbf{x}_n, \mathbf{y}_n) \right] \right\} \tag{13}$$

MAP learning can be interpreted as Regularized Risk Minimization:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \left\{ \|\mathbf{w}\|^2 + C \sum_{n=1}^{N} L(\mathbf{x}_n, \mathbf{y}_n, \mathbf{w}) \right\} \tag{14}$$

$$L(\mathbf{x}_n, \mathbf{y}_n, \mathbf{w}) = \log\big(Z(\mathbf{x}_n, \mathbf{w})\big) - \mathbf{w}^T \phi(\mathbf{x}_n, \mathbf{y}_n) \tag{15}$$

Kernel CRF's

The above formulation of ML/MAP learning can be kernelized using the representer theorem. Refer (12.9) and (12.10) of Chapter 12 of the book (Predicting Structured Data). The CRF factorization helps in keeping the kernelized formulation tractable.

Learning CRF's: SVMs

Learning using ML estimation is one way of learning CRF's. ML estimation can be though of as a specific loss function over which the Empirical Risk is being minimized. Other loss functions give different optimization algorithms. Recalling the rescaled margin SVM formulation:

$$\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{n=1}^{N} \left[\max_{\mathbf{y} \neq \mathbf{y}_n} \left(\Delta(\mathbf{y}_n, \mathbf{y}) + F(\mathbf{x}_n, \mathbf{y}) - F(\mathbf{x}_n, \mathbf{y}_n) \right) \right]_+ \right\} \tag{16}$$

where:

$$F(\mathbf{x}_n, \mathbf{y}_n) = \sum_{t=1}^{T} \mathbf{w}_t^T \phi_t(\mathbf{x}_n, \mathbf{y}_{nt}) \tag{17}$$

Learning CRF's: SVMs ...

The rescaled margin SVM formulation can also be written as:

$$\min_{\mathbf{w},\xi_n}\left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n\right\} \tag{18}$$

subject to:

$$F(\mathbf{x}_n,\mathbf{y}_n) - F(\mathbf{x}_n,\mathbf{y}) \geq \Delta(\mathbf{y}_n,\mathbf{y}) - \xi_n \ \ \forall\mathbf{y},\forall n \tag{19}$$

This formulation has exponentially many constraints, but can be solved in polynomial space using the cutting plane algorithm discussed last time. This algorithm doesnt take any advantage of the factorization provided by the CRF.
(12.13)-(12.16) of the book provide an alternative way of optimizing (18) which uses the CRF factorization to achieve tractability.

Learning CRF's: CGM's

Conditional Graphical Models is an alternative way of training the CRF's. It makes use of the CRF factorization to upper bound the Empirical Risk as a sum of functions which decomposes per clique.