Conditional Random Fields
3. Energy Minimization, Inference and Learning Tasks

Chunpai Wang

May 5, 2019

1 Energy Minimization

In computer vision, the term energy minimization is popularly used to describe approaches in which the solution to the problem is determined by minimizing a function, the "energy". The energy function is defined for all feasible solutions and measures the quality of a solution. Recall the Markov Random Fields, we define the joint probability as

$$p(\mathbf{x}) = \frac{1}{Z}\prod_C \Psi_C(\mathbf{x}_C) \tag{1}$$

which is a product of *potential function* $\Psi_C(\mathbf{x}_C)$ over the maximal cliques C of the graph. It is important to understand that for a general undirected graph the potential functions ϕ_C need not have any obvious or direct relation to marginal or conditional distribution defined over the graph cliques. This property should be contrasted with the directed factorization, where the factors correspond to conditional probabilities over the child-parent sets.

Because we are restricted to potential functions which are **strictly positive** it is convenient to express them as exponential, so that

$$\Psi_C(\mathbf{x}_C) = \exp\{-E(\mathbf{x}_C)\} \tag{2}$$

where $E(\mathbf{x}_C)$ is called an energy function, and the exponential representation is called the *Boltzmann distribution*. The joint distribution is defined as the product of potentials, and so the total energy is obtained by adding the energies of each of the maximal cliques. Finding the state $x \in \mathcal{X}$ with the highest probability can now be seen as an energy minimization problem:

$$\begin{aligned}
\arg\max_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x}) &= \arg\max_{\mathbf{x}\in\mathcal{X}} \frac{1}{Z}\prod_C \Psi_C(\mathbf{x}_C) \\
&= \arg\max_{\mathbf{x}\in\mathcal{X}} \prod_C \Psi_C(\mathbf{x}_C) \\
&= \arg\max_{\mathbf{x}\in\mathcal{X}} \prod_C \exp\{-E(\mathbf{x}_C)\} \\
&= \arg\max_{\mathbf{x}\in\mathcal{X}} \exp\{\sum_C -E(\mathbf{x}_C)\} \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} \sum_C E(\mathbf{x}_C) \\
&= \arg\min_{\mathbf{x}\in\mathcal{X}} E(\mathbf{x})
\end{aligned}$$

Hence, the probability $p(\mathbf{x})$ is completely determined by $E(\mathbf{x}) = \sum_C E(\mathbf{x}_C)$. Energy minimization can be interpreted as solving for the most likely state of some factor graph model.

2 Parameterization

Parameterization means introducing parameters on a probability such that the member of probability family can be specified or identified by the parameter values. Factor graphs define a family of distributions, so we can parameterize the factor graph.

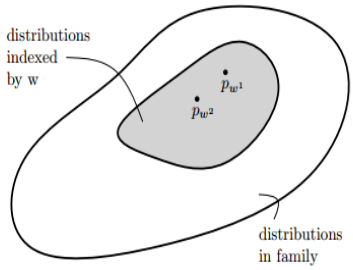

Figure 1: Increasing the number of parameters and features enlarges the realizable subset of distributions; decreasing the number of parameters by parameter sharing makes the set smaller.

Parameters are typically introduced into the energy functions. Why ?

3 Inference and Learning Tasks

Once a factor graph model has been fully specified and parameterized, there are two tasks left to do:

- to learn its parameters from training instances
- to use the model for solving inference tasks on future data instance

3.1 Inference

We need a metric to measure the prediction quality, which can be formalized in the framework of statistical decision theory. Let $d(X, Y)$ denote the probability distribution of the data for the problem we try to solve, which we factor into the conditional probability distribution of the label $d(y|x)$, and a data prior $d(x)$. That is, we want to maximize the $d(Y, X)$. Furthermore, we define the loss function

$$\triangle : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+ \tag{3}$$

and we can measure the quality of prediction $f(x) \in \mathcal{Y}$ by the expected loss of this decision:

$$\mathcal{R}_f^{\triangle}(x) = \sum_{y \in \mathcal{Y}} d(y|x) \triangle(y, f(x)) \tag{4}$$

The most common loss function for classification is 0/1 loss, $\triangle(y, y') = 0$ for $y = y'$ and $\triangle(y, y') = 1$ otherwise. Computing the expected loss we obtain

$$\mathcal{R}_f^{0/1}(x) = d(y \neq f(x)|x) = 1 - p(Y = f(x)|x, w) \tag{5}$$

we assume the parameterized $p(y = f(x)|x, w)$ reflects $d(y|x)$. We can minimize the expected loss by minimizing the posterior $p(Y = f(x)|x, w)$. Thus,we use Maximum A Posteriori (MAP) Inference to make the prediction.

Another popular loss function for structured prediction is the Hamming loss:

$$\triangle(y, y')_H = \frac{1}{|V|} \sum_{i \in V} [y_i \neq y_i'] \tag{6}$$

and expected loss is

$$\mathcal{R}_f^H(x) = 1 - \frac{1}{|V|} \sum_{i \in V} p(Y_i = f(x)_i|x, w) \tag{7}$$

which is minimized by predicting with $f(x)_i = \arg\max_{y_i \in \mathcal{Y}_i} p(Y_i = y_i|x, w)$. This should be done by the probabilistic inference. Given a factor graph, parameterization, and weight vector w, and given the observation x, probabilstic inference is to find the value of the log partition function and the marginal distributions for each factor,

$$\log Z(x, w) = \log \sum_{y \in \mathcal{Y}} \exp(-E(y; x, w)) \tag{8}$$

$$\mu_F(y_F) = p(Y_F = y_F|x, w), \quad \forall F \in \mathcal{F}, \forall y_F \in \mathcal{Y}_F \tag{9}$$

where F denotes the factor.

3.2 Learning

Learning graphical models from training data is to find the best model among a large class of possible models. In most applications, we assume that the model structure and parameterization are specified manually, and learning is to finding a vector of real-valued parameters.

Probabilistic Parameter Learning Let $d(y|x)$ be the unknown conditional distribution of labels for a problem to be solved. For a parameterized conditional distribution $p(y|x, w)$ with parameters $w \in \mathbb{R}^D$, probabilistic parameter learning is the task of finding a point estimate of the parameter w^* that makes $p(y|x, w^*)$ closest to $d(y|x)$ for every $x \in \mathcal{X}$.

Loss-Minimizing Parameter Learning Let $d(x, y)$ be the unknown distribution of data in labels, and let $\triangle : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Loss minimizing parameter learning is the task of finding a parameter value w^* such that the expected prediction risk

$$E_{(x,y)\sim d(x,y)}[\triangle(y, f_p(x))] \tag{10}$$

References

[1] Sutton, Charles, and Andrew McCallum. "An introduction to conditional random fields." Foundations and Trends® in Machine Learning 4.4 (2012): 267-373.

[2] http://pages.cs.wisc.edu/~jerryzhu/cs769/CRF.pdf

[3] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision." Foundations and Trends® in Computer Graphics and Vision 6.3–4 (2011): 185-365. http://www.nowozin.net/sebastian/papers/nowozin2011structured-tutorial.pdf