

Learning MN Parameters with Alternative Objective Functions

Sargur Srihari
srihari@cedar.buffalo.edu

Topics

- Max Likelihood & Contrastive Objectives
- Contrastive Objective Learning Methods
 - Pseudo-likelihood
 - Gradient descent
 - Ex: Use in Collaborative Filtering
 - Contrastive Optimization criteria
 - Contrastive Divergence
 - Margin-based Training

Recapitulate ML learning

- Task: Estimate parameters of a MN

$$P(X_1, \dots, X_n; \theta) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{i=1}^k \theta_i f_i(D_i) \right\} \quad \text{where} \quad Z(\theta) = \sum_{\xi} \exp \left\{ \sum_i \theta_i f_i(\xi) \right\}$$

- with a fixed structure given data set \mathcal{D}
- Simplest is to maximize log-likelihood objective given samples ξ_1, \dots, ξ_M , which is

$$\ell(\theta; \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(\xi[m]) \right) - M \ln Z(\theta)$$

- Although concave, no analytical form for maximum
 - Can use iterative gradient ascent in param. space

$$\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta; D) = E_D[f_i(\chi)] - E_{\theta}[f_i] \quad \text{since} \quad \frac{\partial}{\partial \theta_i} \ln Z(\theta) = \frac{1}{Z(\theta)} \sum_{\xi} f_i(\xi) \exp \left\{ \sum_j \theta_j f_j(\xi) \right\} = E_{\theta}[f_i]$$

- Good news: Likelihood is concave, Guaranteed to converge
- Bad news: Each step needs inference; estimation is intractable

Replacing ML objective

- Previously seen approx. inference methods:
 - Belief propagation, MaxEnt, Sampling +SGD
- Now we look at replacing the objective function with one that is more tractable
- The ML objective is: $\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(\xi[m]) \right) - M \ln Z(\theta)$ $Z(\theta) = \sum_{\xi} \exp \left\{ \sum_i \theta_i f_i(\xi) \right\}$
- For simplicity focus on the case of a single data instance ξ :

$$\ell(\theta : \xi) = \ln \tilde{P}(\xi | \theta) - \ln Z(\theta)$$

Log-likelihood of a single sample

- In the case of a single instance ξ

$$\ell(\theta : \xi) = \ln \tilde{P}(\xi | \theta) - \ln Z(\theta)$$

$$Z(\theta) = \sum_{\xi} \exp \left\{ \sum_i \theta_i f_i(\xi) \right\}$$

- Expanding the partition function $Z(\theta)$

$$\ell(\theta : \xi) = \ln \tilde{P}(\xi | \theta) - \ln \left(\sum_{\xi'} \tilde{P}(\xi' | \theta) \right)$$

Summing over all possible values of dummy variable ξ'

- Maximizing ℓ is to increase distance (*contrast*) between the two terms
- Consider each of the two terms separately

$$\ln \tilde{P}(\xi | \theta)$$

and

$$\ln \tilde{P}(\xi | \theta) = - \sum_{i=1}^k \theta_i f_i(D_i)$$

First Term of Objective

- First term is:

$$\ln \tilde{P}(\xi | \theta)$$

- Objective aims to increase the log measure
 - i.e., Logarithm of unnormalized probability of observed data instance ξ
 - log measure is a linear function of parameters in log-linear representation, $\ln \tilde{P}(\xi | \theta) = -\sum_{i=1}^k \theta_i f_i(D_i)$
 - Thus that goal can be accomplished by
 - Increasing all parameters associated with positive empirical expectations in ξ and decreasing all parameters associated with negative empirical expectations
 - We can increase it unboundedly using this approach

Second Term of Objective

- Second Term: $\ln \left(\sum_{\xi'} \tilde{P}(\xi' | \theta) \right)$
- It is the logarithm of the sum of unnormalized measures of all possible instances in $Val(\mathcal{X})$
 - It is the aggregate of the measures of all instances

The Contrastive Objective Approach

- Can view log-likelihood objective as
 - aiming to increase distance between log measure of ξ and aggregate of the measures of all instances
- The key difficulty with this formulation
 - second term has exponential instances in $Val(\chi)$ and requires inference in the network
- It suggests an approach to approximation:
 - We can move our parameters in the right direction if we aim to increase the difference between
 1. The log-measure of data instances and
 2. A more tractable set of other instances, one not requiring summation over an exponential space

Two Approaches to increase probability gap

1. Pseudolikelihood and its generalizations

- Easiest method circumventing intractability of network inference
- Simplifies likelihood by replacing exponential no. of summations with several summations, each more tractable

2. Contrastive Optimization

- Drive probability of observed samples higher
- Contrast data with a randomly perturbed set of neighbors

Pseudolikelihood for Tractability

- Consider likelihood of single instance ξ :

$$P(\xi) = \prod_{j=1}^n P(x_j | x_1, \dots, x_{j-1})$$

From chain rule $P(x_1, x_2) = P(x_1) P(x_2 | x_1)$ and $P(x_1, x_2, x_3) = P(x_1) P(x_2 | x_1) P(x_3 | x_1, x_2)$

- Approximate by replacing each product term by conditional probability of x_j given all other variables

$$P(\xi) = \prod_{j=1}^n P(x_j | x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n)$$

- Gives the pseudolikelihood (PL) objective

$$\ell_{PL}(\theta : \mathcal{D}) = \frac{1}{M} \sum_m \sum_j \ln P(x_j[m] | \mathbf{x}_{-j}[m], \theta)$$

where \mathbf{x}_{-j} stands for $x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n$

- This objective measures ability to predict each variable given full observation of all other variables

Pseudolikelihood and Multinomial

- The predictive model takes a form that generalizes the multinomial logistic CPD

$P(Y|X_1, \dots, X_k)$ such that

$$P(y^j | X_1, \dots, X_k) = \frac{\exp(\ell_j(X_1, \dots, X_k))}{\sum_{j'=1}^m \exp(\ell_{j'}(X_1, \dots, X_k))}$$

$$\ell_j(X_1, \dots, X_k) = w_{j,0} + \sum_{i=1}^k w_{j,i} X_i$$

- Identical to it when the network consists of only pairwise features
 - Factors over edges in the network
- We can use conditional independence properties in the network to simplify the expression,

$$\ell_{PL}(\theta : \mathcal{D}) = \frac{1}{M} \sum_m \sum_j \ln P(x_j[m] | \mathbf{x}_{-j}[m], \theta)$$

- removing from the rhs of $P(X_j | \mathbf{X}_{-j})$ any variable that is not a neighbor of X_j

Simplification in Pseudolikelihood

- The pseudolikelihood (PL) objective is

$$\ell_{PL}(\theta : \mathcal{D}) = \frac{1}{M} \sum_m \sum_j \ln P(x_j[m] | \mathbf{x}_{-j}[m], \theta)$$

- Whereas likelihood is

$$\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(\xi[m]) \right) - M \ln Z(\theta)$$

$$\ln Z(\theta) = \ln \sum_{\xi} \exp \left\{ \sum_i \theta_i f_i(\xi) \right\}$$

- Pseudolikelihood eliminates exponential summation over instances with several summations, each of which is more tractable

- In particular

$$P(x_j | \mathbf{x}_{-j}) = \frac{P(x_j, \mathbf{x}_{-j})}{P(\mathbf{x}_{-j})} = \frac{\tilde{P}(x_j, \mathbf{x}_{-j})}{\sum_{x_j} \tilde{P}(x_j, \mathbf{x}_{-j})}$$

- Global partition function has disappeared
- Requires only summation over X_j
- But there is a contrastive perspective
 - Described next

Contrastive perspective of PL

- Pseudolikelihood objective of single data ξ :

$$\begin{aligned}\sum_j \ln P(x_j | \mathbf{x}_{-j}) &= \sum_j \left(\ln \tilde{P}(x_j, \mathbf{x}_{-j}) - \ln \sum_{x'_j} \tilde{P}(x'_j, \mathbf{x}_{-j}) \right) \\ &= \sum_j \left(\ln \tilde{P}(\xi) - \ln \sum_{x'_j} \tilde{P}(x'_j, \mathbf{x}_{-j}) \right)\end{aligned}$$

- Each term of final sum is a contrastive term
 - Where we aim to increase difference between log-measure of training instance ξ and an aggregate of log-measures of instances that differ from ξ in the assignment to precisely one variable
 - In other words we are increasing the contrast between our training instance ξ and the instances in a local neighborhood around it

Pseudolikelihood is concave

- Further simplification of summands in the expression $\sum_j \ln P(x_j | \mathbf{x}_{-j}) = \sum_j \left(\ln \tilde{P}(\xi) - \ln \sum_{x'_j} \tilde{P}(x'_j, \mathbf{x}_{-j}) \right)$ obtains

$$\ln P(x_j | \mathbf{x}_{-j}) = \left(\sum_{i: \text{Scope}[f_i] \ni X_j} \theta_i f_i(x_j, \mathbf{x}_{-j}) \right) - \ln \left(\sum_{x'_j} \exp \left\{ \sum_{i: \text{Scope}[f_i] \ni X_j} \theta_i f_i(x'_j, \mathbf{x}_{-j}) \right\} \right)$$

- Each term is a log-conditional-likelihood term for a MN for a single variable X_j conditioned on rest
- Thus it follow that the function is concave in the parameters θ
- Since a sum of concave functions is concave the pseudolikelihood objective
 - $\ell_{PL}(\theta : \mathcal{D}) = \frac{1}{M} \sum_m \sum_j \ln P(x_j[m] | \mathbf{x}_{-j}[m], \theta)$ is also concave

Gradient of pseudolikelihood

- To compute gradient we use
 - to obtain

$$\ln P(x_j | \mathbf{x}_{-j}) = \left(\sum_{i: \text{Scope}[f_i] \supset X_j} \theta_i f_i(x_j, \mathbf{x}_{-j}) \right) - \ln \left(\sum_{x_j'} \exp \left\{ \sum_{i: \text{Scope}[f_i] \supset X_j} \theta_i f_i(x_j', \mathbf{x}_{-j}) \right\} \right)$$

$$\frac{\partial}{\partial \theta_i} \ln(x_j | \mathbf{x}_{-j}) = f_i(x_j, \mathbf{x}_{-j}) - E_{x_j' \sim P_\theta(X_j | \mathbf{x}_{-j})} [f_i(x_j', \mathbf{x}_{-j})]$$

- If X_j is not in the scope of f_i then $f_i(x_j, \mathbf{x}_{-j}) = f_i(x_j', \mathbf{x}_{-j})$ for any x_j' and the two terms are identical making the derivative 0. Inserting this into ℓ_{PL} we get

$$\frac{\partial}{\partial \theta_i} \ell_{\text{PL}}(\theta : \mathcal{D}) = \sum_{j: X_j \in \text{Scope}[f_i]} \left(\frac{1}{M} \sum_m f_i(\xi[m]) - E_{x_j' \sim P_\theta(X_j | \mathbf{x}_{-j}[m])} [f_i(x_j', \mathbf{x}_{-j}[m])] \right)$$

- It is much easier to compute this than $\frac{\partial}{\partial \theta_i} \frac{1}{M} \ell(\theta : D) = E_D[f_i(\chi)] - E_\theta[f_i]$
- Each expectation term requires a summation over a single random variable X_j , conditioned on all of its neighbors
- A computation that can be performed very efficiently

Relationship between likelihood and pseudolikelihood

- *Theorem:* Assume that our data are generated by a log-linear model P_{θ^*} , i.e.,
$$P(X_1, \dots, X_n; \theta) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{i=1}^k \theta_i f_i(D_i) \right\}$$
 - Then as M goes to infinity, with probability approaching 1, θ^* is a global optimum of the pseudolikelihood objective
$$\ell_{PL}(\theta : \mathcal{D}) = \frac{1}{M} \sum_m \sum_j \ln P(x_j[m] | \mathbf{x}_{-j}[m], \theta)$$
- Result is an important, but there are limitations
 - Model being learnt must be expressive
 - But model never perfectly represents true distribution
 - Data distribution must be near generating distribution
 - Often not enough data to approach large sample limit¹⁶

Type of queries determine method

- How good is the pseudolikelihood objective depends on the type of queries
 - If we condition on most of variables and condition on few, pseudolikelihood is a very close match to the type of predictions we would like to make
 - So pseudolikelihood may well be better than likelihood
 - Ex: in learning a MN for collaborative filtering, we take user's preference for all items as observed except the query item
 - Conversely, if a typical query involves most or all variables, the likelihood objective is better
 - In learning a CRF model for image segmentation

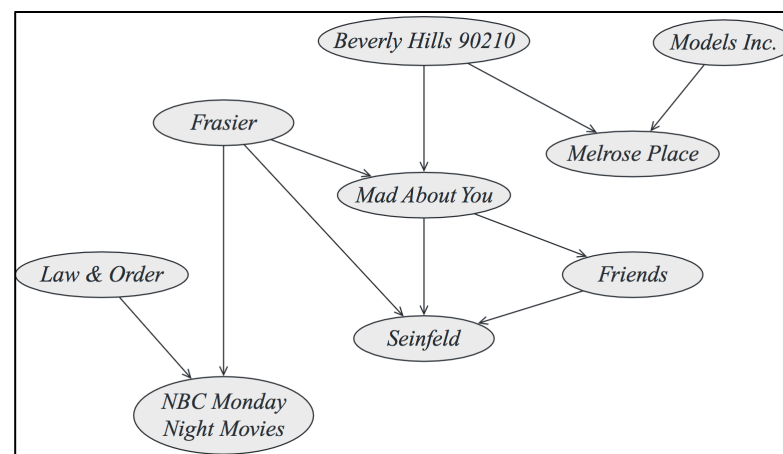
Ex: Collaborative Filtering

- We want to recommend to a user an item based on previous items he bought/liked
- Because we don't have enough data for any single user to determine his/her preference, we use *collaborative filtering*
 - i.e., use observed preferences of others to determine preferences for any other user
 - One approach: learn dependency structure between different purchases as observed in the population
 - Item i is a variable X_i in a joint distribution and each user is an instance
 - View purchase/non-purchase as values of a variable

A Collaborative Filtering Task

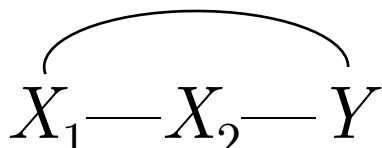
- Given a set of purchases S we can compute the probability that user would like new item i
- This is a probabilistic inference task where all purchases other than i are set to *False*; thus all variables other than X_i are observed

If we condition on most of variables and condition on few, Pseudolikelihood (PL) is a very close match to the type of predictions we would like to make



Network learned from Nielsen TV rating data capturing viewing record of sample viewers. Variables denote whether a TV program was watched

Limitation of pseudolikelihood (PL)

- Ex: MN with 3 variables: $X_1 \text{---} X_2 \text{---} Y$

 - X_1 and X_2 highly correlated, Both not as correlated to Y
 - Best predictor for X_1 is X_2 and vice versa
 - Pseudolikelihood likely to overestimate $X_1 \text{---} X_2$ parameters and almost entirely dismiss $X_1 \text{---} Y$ and $X_2 \text{---} Y$
 - Excellent predictor for X_2 when X_1 is observed, but useless when only Y , not X_1 , is observed
- PL assumes local neighborhood is observed
 - cannot exploit weaker or long-range dependencies
- Solution: generalized pseudolikelihood (GPL)

We define a set of subsets of variables $\{X_s : s \in S\}$ and define an objective:

$$\ell_{GPL}(\theta : \mathcal{D}) = \frac{1}{M} \sum_m \sum_s \ln P(\mathbf{x}_s[m] | \mathbf{x}_{-s}[m], \theta)$$

where $\mathbf{X}_{-s} = \chi - \mathbf{X}_s$

Contrastive Optimization Criteria

- Both Likelihood and Pseudolikelihood
 - attempt to increase log-probability gap between probability of observed set of m instances \mathcal{D} and logarithm of the aggregate probability of a set of instances:

$$\ell(\theta : \mathcal{D}) = \sum_i \theta_i \left(\sum_m f_i(\xi[m]) \right) - M \ln Z(\theta)$$

$$\ell(\theta : \xi) = \ln \tilde{P}(\xi | \theta) - \ln \left(\sum_{\xi'} \tilde{P}(\xi' | \theta) \right)$$

$$\ell_{PL}(\theta : \mathcal{D}) = \frac{1}{M} \sum_m \sum_j \ln P(x_j[m] | \mathbf{x}_{-j}[m], \theta)$$

$$\ln P(x_j | \mathbf{x}_{-j}) = \left(\sum_{i: \text{Scope}[f_i] \ni X_j} \theta_i f_i(x_j, \mathbf{x}_{-j}) \right) - \ln \left(\sum_{x_j'} \exp \left\{ \sum_{i: \text{Scope}[f_i] \ni X_j} \theta_i f_i(x_j', \mathbf{x}_{-j}) \right\} \right)$$

- Based on this intuition, a range of methods developed to increase log-probability gap
 - By driving probability of observed data higher relative to other instances,
 - we are tuning the parameters to predict the data better

Contrastive Objective Definition

- Consider a single training instance ξ
 - We aim to maximize the *log-probability gap*
- Importantly, the expression takes a simple form

$$\ln \tilde{P}(\xi | \theta) - \ln \tilde{P}(\xi' | \theta)$$

- where ξ' is some other instance, whose selection we discuss shortly

$$\ln \tilde{P}(\xi | \theta) - \ln \tilde{P}(\xi' | \theta) = \theta^T [f(\xi) - f(\xi')]$$

- For a fixed instantiation of ξ' , this expression is a linear function of θ and hence is unbounded
- For a coherent optimization objective, choice of ξ' has to change throughout the optimization
 - Even then, take care to prevent unbounded parameters²²

Two Contrastive Optimization Methods

- One can construct many variants of this method
 - i.e., methods to increase log-probability gap
- Two methods for choosing ξ that have been useful in practice:
 1. Contrastive divergence
 - Popularity of method has grown
 - Used in *deep learning*– for training layers of RBMs
 2. Margin-based training

Contrastive Divergence (CD)

- In this method we contrast our data instances \mathcal{D} with set of randomly perturbed *neighbors* \mathcal{D}^-
 - We aim to maximize
$$\ell_{CD}(\theta : \mathcal{D} || \mathcal{D}^-) = E_{\xi \sim \tilde{P}_{\mathcal{D}}} [\ln \tilde{P}_{\theta}(\xi)] - E_{\xi \sim \tilde{P}_{\mathcal{D}^-}} [\ln \tilde{P}_{\theta}(\xi)]$$
 - where $P_{\mathcal{D}}$ and $P_{\mathcal{D}^-}$ are the empirical distributions relative to \mathcal{D} and \mathcal{D}^-
- The set of contrasted instances \mathcal{D}^- will necessarily differ at different stages in the search
 - Choosing instances to contrast is next

Choice of Data instances

- Given current θ , what instances to which we want to contrast our data instances \mathcal{D} ?
- One intuition: move θ in a direction that increases probability of instances in \mathcal{D} relative to “typical” instances in current distribution
 - i.e., increase probability gap between instances $\xi \in \mathcal{D}$ and instances ξ sampled randomly from P_θ
- Thus, we can generate a contrastive set \mathcal{D}^- by sampling from P_θ and then maximizing the objective in

$$\ell_{CD}(\theta : \mathcal{D} \parallel \mathcal{D}^-) = E_{\xi \sim \tilde{P}_{\mathcal{D}}} [\ln \tilde{P}_\theta(\xi)] - E_{\xi \sim \tilde{P}_{\mathcal{D}^-}} [\ln \tilde{P}_\theta(\xi)]$$

How to sample from P_θ ?

- We can run a Markov chain defined by the MN P_θ using Gibbs sampling and initializing from the instances in \mathcal{D}
 - Once the chain mixes we can collect samples from the distribution
 - Unfortunately, sampling from the chain for long enough to achieve mixing takes far too long for the inner loop of a learning algorithm
 - So we initialize from instances in \mathcal{D} and run the chain only for a few steps
 - Instances from these short sampling runs define \mathcal{D} -

Updating parameters in CD

- We want model to give high probability to instances in \mathcal{D} relative to the perturbed instances in \mathcal{D}^-
 - Thus we want to move our parameters in a direction that increases the probability of instances in \mathcal{D} relative to the perturbed instances in \mathcal{D}^-
- Gradient of objective is easy to compute
$$\frac{\partial}{\partial \theta_i} \ell_{CD}(\theta : \mathcal{D} || \mathcal{D}^-) = E_{\hat{P}_D} [f_i(\chi)] - E_{\hat{P}_{D^-}} [f_i(\chi)]$$
 - In practice approximation we get by taking only a few steps in the Markov chain provides a good direction for the search

Margin-Based Training

- Very different intuition in settings where our goal is to use the network for predicting a MAP assignment
- Ex: in image segmentation we want the learned network to predict a single high probability assignment to the pixels that will encode our final segmentation output
- Occurs only when queries are conditional
- So we describe objective for a CRF

Margin-Based Training

- Training set consists of pairs $D = \left\{ \left(\mathbf{y}[m], \mathbf{x}[m] \right) \right\}_{m=1}^M$
- Given observation $\mathbf{x}[m]$ we would like learned model to give highest probability to $\mathbf{y}[m]$
 - i.e., we would like $P_{\theta}(\mathbf{y}[m] | \mathbf{x}[m])$ to be higher than any other probability $P_{\theta}(\mathbf{y} | \mathbf{x}[m])$ for $\mathbf{y} \neq \mathbf{y}[m]$
 - i.e., Maximize the margin

$$\ln P_{\theta}(\mathbf{y}[m] | \mathbf{x}[m]) - \left[\max_{\mathbf{y} \neq \mathbf{y}[m]} \ln P_{\theta}(\mathbf{y} | \mathbf{x}[m]) \right]$$

- the difference between the log-probability of the target assignment $\mathbf{y}[m]$ and “next best” assignment