Research Journal of Applied Sciences, Engineering and Technology 5(10): 2969-2943, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: September 16, 2012 Accepted: October 31, 2012 Published: March 25, 2013

Application of EM Algorithm in Statistics Natural Language Processing

Xuexia Gao and Yun Wang
Computer and Information Engineering College, Xinxiang University, Xinxiang 453000, China

Abstract: This study describes the basic framework of EM algorithm and gives how to apply EM algorithm to solve the problem of maximum-likelihood parameters estimation combining with the models of HMM and PCFG. In the process of statistics natural language, one kind of problem is often encountered that is how to solve the parameter's maximum-likelihood estimation when observation data is incomplete. EM algorithm is the classical method to solve this problem. Finally, the advantages and disadvantages of EM algorithm are discussed.

Keywords: Context-free grammar, EM algorithm, hidden Markov model, likelihood function, natural language, parameter estimation

INTRODUCTION

Along with the appearance of large scale machine readable corpus and the rapid increase of computer operation speed and storage capacity, empiricism in natural language processing field also obtains a rapid revival. The introduction based on statistical learning method has made computational linguistics field a great change, this study method can through the training of corpus to automatically or partly automatically process linguistics knowledge, which has great significance to solve the problem "knowledge acquisition bottleneck”.

But in the statistics of natural language processing, there are often this kind of parameters valuation issues, that is, when the observed data is incomplete how to solve the maximum likelihood estimation of parameters. EM (Expectation Maximum) algorithm is the classic algorithm to solve this kind of problem, which was brought out by Dempster, Laird and Rubin in 1997 and widely used in parameter estimation of incomplete data.

EM algorithm has two major applications (Bilmes, 1997): one is used in the parameter estimation for data loss, another application is assuming there exists other missing parameters (these parameters may not exist or be hidden), which can greatly simplify likelihood function. In natural language statistics field the latter’s application is more common. This study first gives the basic framework of general EM algorithm and then combining with Hidden Markov Model (HMM) and probabilistic context-free grammar model presents how to use EM algorithm to solve the parameter of the maximum likelihood estimation, the conclusion are given in the end.

This study describes the basic framework of EM algorithm and gives how to apply EM algorithm to solve the problem of maximum-likelihood parameters estimation combining with the models of HMM and PCFG. In the process of statistics natural language, one kind of problem is often encountered that is how to solve the parameter's maximum-likelihood estimation when observation data is incomplete. EM algorithm is the classical method to solve this problem. Finally, the advantages and disadvantages of EM algorithm are discussed.

BASIC FRAMEWORK OF EM ALGORITHM

In this study the basic framework are presented in literature (Bilmes, 1997; Zoubin and Michael, 1994, 1995). The basic idea of EM algorithm divides the problem into two steps to solve, these are E steps (to the logarithm of complete data set likelihood function to solve conditional expectation) and M steps(to maximize the solved expectations) and then constantly iterate E steps and M steps until work out the maximum points so far. The formal description of algorithm is as follows:

Assume the complete data set is Z = (X, Y), data set X is the observed data collection, Y is missing (or hiding) data set, in parameters set Θ of Z, the joint density function about X, Y is $p(z|\Theta) = p(x, y|\Theta) = p(y|x, \Theta)$, hereinto, $x \in X$, $y \in Y$. Now the likelihood function of complete data set Z is $L(\Theta|Z) = L(\Theta|X,Y) = p(X, Y| \Theta)$.

Step 1: The step 1 of EM algorithm (E step) is to find logarithm likelihood function log $p(X, Y|\Theta)$, while given observation data set X and the current parameter set $\Theta^{(i-1)}$, the expectation value about unknown data set Y is the value to calculate the next expression: $Q(\Theta, \Theta^{(i-1)}) = E[\log p(X,Y|\Theta^{(i-1)})]$, hereinto Θ is the new parameter set after optimization and makes the

Corresponding Author: Xuexia Gao, Computer and Information Engineering College, Xinxiang University, China

value of function Q increasing with the new parameter.

Step 2: The step 2 (M step) of EM algorithm is to maximize expectation value of part 1, that is next expression $\Theta^{(i)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(i-1)})$, this two steps constantly iterates, each iteration will ensure to increase the logarithm likelihood function values and ensure that likelihood function converges to a local maximum value point.

THE APPLICATION OF EM ALGORITHM IN THE STATISTICS OF NATURAL LANGUAGE PROCESS

EM algorithm has a wide application range in the statistics of natural language process, such as the forward-backward algorithm in HMM, the inside-outside algorithm in PCFG, EM clustering algorithm and no supervision semantic disambiguation algorithm, which are the specific applications of EM algorithm for parameter estimation problems. Below there are the detail parameter estimate process between forward-backward algorithm and inside-outside algorithm.

The parameter estimation problem of HMM: HMM parameter estimation problem is according to some observation sequence to estimate a group of HMM parameters (A, B, π), which makes the probability maximization of producing these observation sequences under this model's parameters. Forward-backward algorithm (also called Baum-Welch algorithm) is an often used method, the proposed algorithm is equivalent to EM algorithm. The model parameter $\lambda = (A, B, \pi)$ may be adjusted to local extremum of $P(O|\lambda)$; this is a revaluation iterative process of parameters. In order to facilitate description, forward-backward variables can be defined:

Forward variable is: $\alpha_t(i) = P(O_1, O_2, \cdots, O_t, q_t = s_i|\lambda)$ which represents in a given model λ, from moment 1 to moment t the observed sequence is (O_1, O_2, … , O_t) and at t moment the system state is the probability of s_i.

Backward variable is: $\beta_t(i) = P(O_{t+1}, O_{t+2}, \cdots, O_T|q_t = s_i, \lambda)$ which represents in the condition of a given model and at moment t the status is s_i, from moment t+1 to moment T the probability of generating the observed sequence $(O_{t+1}, O_{t+2}, \cdots, O_T)$.

The two variables can be conducted through the forward and backward process, the details can be checked in literature (Christopher, 1999), the observation data of HMM is $O = (o_1, o_2, \cdots, o_T)$, hidden (or invisible) status sequence is $q = (q_1, q_2, \cdots, q_T)$ and incomplete data set of the likelihood function is $P(O|\lambda)$ and the likelihood function of complete data set is $P(O, Q|\lambda)$. So Q function is defined as:

$$Q(\lambda, \lambda^{'}) = \sum_{q \in \gamma} \log P(O, q|\lambda) P(O, q|\lambda^{'})$$

λ is the new parameter estimated from current parameter and observing sequence O, γ is the status sequence value space with length T. A particular status sequence q is given, $P(O,q|\lambda)$ can be written as:

$$P(O, q|\lambda) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(o_t)$$

In the expression: π_{q_0} represents initial condition, $a_{qt\text{-}1qt}$ represents the probability of status $q_{t\text{-}1}$ transferring to status q_t, $b_{q_t}(o_t)$ represents the probability of launch symbol O_t in status q_t. Therefore, Q function can be rewritten as:

$$\begin{aligned} Q(\lambda, \lambda^{'}) &= \sum_{q \in \gamma} \log \pi_{q_0} P(O, q|\lambda^{'}) \\ &+ \sum_{q \in \gamma} \left(\sum_{t=1}^{T} \log a_{q_{t-1}q_t} \right) P(O, q|\lambda^{'}) \\ &+ \sum_{q \in \gamma} \left(\sum_{t+1}^{T} \log b_{q_t}(o_t) \right) P(O, q|\lambda^{'}) \end{aligned} \tag{1}$$

Because the parameters need to be optimized are distributed in above three independent expressions, so each part can be independent optimized to achieve the expression of the parameters. Lagrange multiplier method is used respectively to calculate the conditional extremum values to these three parts under the constraint conditions:

$$\sum_i \pi_i = 1, \quad \sum_{j=1}^{N} a_{ij} = 1, \quad \sum_{j=1}^{L} b_i(j) = 1$$

At last the three parameters can be obtained:

$$\overline{\pi_i} = \frac{P(O, q_0 = i|\lambda^{'})}{P(O|\lambda)} \tag{2}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T} P(O, q_{t-1} = i, q_t = j|\lambda^{'})}{\sum_{t=1}^{T} P(O, q_{t-1} = i|\lambda^{'})} \tag{3}$$

$$\overline{b_i(k)} = \frac{\sum_{t=1}^{T} P(O, q_t = i|\lambda^{'}) \delta_{ot,vk}}{\sum_{t=1}^{T} P(O, q_t = i|\lambda^{'})} \tag{4}$$

The probability in above three expressions can obtained through the forward variable $\alpha_t(i)$ and backward variable $\beta_t(i)$.

EM algorithm starts from an initial model λ = (A, B, π), using above set of parameter revaluation formula to get a new model $\overline{\lambda} = \left(\overline{A}, \overline{B}, \overline{\pi}\right)$ to replace the original model. With the constant iteration, Baum proved that the new model $P(O|\lambda)$ will continue to change until reach local maximum value point. The final obtained Hidden Markov Model is called the maximum likelihood model; this model makes the probability of an observed sequence O maximization.

After the successful application in speech recognition areas of Hidden Markov Model (Rabiner, 1989), in part-of-speech tagging field which also had achieved great success and made the part-of-speech tagger to the practical. In literature (Cutting *et al*., 1992) part-of-speech tagger based on HMM used half part of Brown corpus (500 000 words) to do training, after eight times of iteration training, the other half part of Brown corpus is done mark, accuracy achieves 96%, the accuracy is also better at present. Of course, Hidden Markov Model own problems also cannot be ignored, such as data sparse problem, which needs lots of training linguistic data and the parameters number of the model is too big, linguistic knowledge gained from corpus is inconvenient for artificial reading and can't get long distance linguistic information, etc. In order to overcome these shortcomings, some deformations of Hidden Markov Model are brought out, such as: variable memory Hidden Markov Model, gradation Hidden Markov Model, etc, these improvements in some degree overcome these shortcomings.

The parameter estimation problem in probabilistic context-free grammar (PCFG): Probabilistic Context-Free Grammar (PCFG), is a simple Context-Free Grammar (CFG) after adding probability for rule, probability indicates the possibility size of different rewrite rules. PCFG through three hypotheses (that is, location-free hypothesis, context-free assumption, ancestors-free assumption), not only inherits the context-free of CFG program, also make probability value to be context-free used, so that it can compute the probability of each analysis tree, the analysis tree with the biggest probability is the most probable analysis tree.

The introduction of rules probability is beneficial to the solution of syntactic disambiguation problem, also increases the flexible process capability of sentence structure analysis.

PCFG model also has a parameter estimation problem, that is, a grammar G and training sentence W_{lm} are given, how to select probabilities for grammar rules, making the probability of sentence training $\arg\max_{g} P\,(W_{lm}|G)$ maximum. In the standard form of Chomsky, a PCFG parameter is:

The probability of grammar rules:

$$P\left(N^j \rightarrow N^r N^s|G\right)$$

The probability of vocabulary rules:

$$P\left(N^j \rightarrow w^k|G\right)$$

The constraint conditions of parameters:

$$\forall j \in (1,2,\cdots,n)\sum_{r,s} P\left(N^j \rightarrow N^r N^s\right) + \sum_k P\left(N^j \rightarrow w^k\right) = 1$$

In order to estimate the value of parameters, at first outside variables $\alpha_j(p,q)$ and inside variables $\beta_j(p, q)$ need to be defined.

To outside variables:

$$a_j(p,q) = P\left(w_{l(p=1)}, N^j_{pq}, w_{(q+1)m}|G\right) \tag{5}$$

To inside variables:

$$\beta_j(p,q) = P\left(w_{pq}|N^j_{pq}, G\right) \tag{6}$$

The inside variable $\beta_j(p, q)$ is the probability of word string w_{pq} = wp w_{p+1} … w_q deduced by non-terminal symbol N^j; outside variable $\alpha_j(p, q)$ is the probability of sentence $w_1 \cdots w_{p-1} N^j_{pq} w_{q+1} \cdots w_m$ derived by the grammar beginning symbol N^s. The two variable values can be obtained by PCFG context-free hypothesis.

In EM algorithm (inward-outward algorithm), no-observation (or hidden) data is rules $N^j \rightarrow \xi$ (including $N^j \rightarrow N^s N^r$ and $N^j \rightarrow w^k$) being used to create a particular word sequence w_{pq}. E step is the expected use times to calculate the rule; M step is the maximum likelihood estimation about probability of calculation rules. It is allowed to train PCFG on the corpus without syntax ingredient label.

Here introduces inward-outward algorithm within two steps.

First of all, expected using frequency of rules needs to be calculated in order to determine the probability of rules, what needs to calculate is:

$$\hat{P}\left(N^j \rightarrow \xi\right) = \frac{C\left(N^j \rightarrow \zeta\right)}{\sum_{Y} C\left(N^j \rightarrow \zeta\right)}$$

In expression, C(·) is a frequency counter of specific rule, if a corpus after syntactic analysis can be obtained, the probability value can be directly calculated. But usually it is difficult to get an analyzed corpus; a certain rule is unknown to be used for the formation of a particular word sequence. Thus rule is given an initial probability estimate (which can be randomly chosen), then iterative algorithm is used to improve estimate.

To a single training sentence w_{lm}, inside-outside variables are introduced, the probability of sentence w_{lm}, i.e., $P(w_{lm}|G)=\beta_j(l,m)$, or $P(w_{lm}|G)=\sum_j a_j(k,k)p(N_j \to w_k)$. In the deduction, the estimate value of use frequency of non-terminal signal N^j is:

E (the using times of N^j in deduction) = $\sum_{p=l}^{m}\sum_{q=p}^{m} P(N^j \Rightarrow w_{pq}|N^l \Rightarrow w_{lm}, G)$

So in the deduction the estimated value of use frequency of P $N^j \to N^s N^r$ and $N^j \to w^k$ are (E $N^j \to N^r N^s, N^j$ used times) = $\sum_{p=1}^{m-1}\sum_{q=p+1}^{m} P(N^j \to N^r N^s|N^l \Rightarrow w_{lm}, G)$, E($N^j \to w^k, N^j$ used times) = $P(N^j \to w^k|N^l \Rightarrow w_{lm}, G)$.

Therefore the maximum likelihood estimation value of sum of $P(N^j \to N^r N^s|G)$ and $P(N^j \to w^k|G)$ are:

$$\hat{P}(N^j \to N^r N^s|G) = \frac{E(N^j \to N^r N^s, N^j)}{E(N^j)} \quad (7)$$

$$\hat{P}(N^j \to w^k|G) = \frac{E(N^j \to w^k, N^j)}{E(N^j)} \quad (8)$$

Training corpus can not be only one sentence in deduction, suppose there is a training sentences set W = $(W_1, W_2 \ldots W_\omega)$, hereinto $W_i = (w_{i,1}\, w_{i,2} \ldots w_{i,m}$. Suppose f_i, g_i and h_i represent the probability of branch nodes, pre-terminal nodes and other non-terminal signal nodes in analysis tree of sentence W_i, the expression can be calculated:

$$f_i(p,q,j,r,s) = P(N^j \to N^r N^s \Rightarrow w_{pq}|N^l \Rightarrow W_i, G) \quad (9)$$

$$g_i(j,k) = P(N^j \to w^k|N^l \Rightarrow W_i, G) \quad (10)$$

$$h_i(p,q,j) = P(N^j \Rightarrow w_{pq}|N^l \Rightarrow W_i, G) \quad (11)$$

Suppose that sentences in training corpus are independent, then in revaluation process through the contribution sum of more sentences the revaluation formula is given as follows:

$$\hat{P}(N^j \to N^r N^s) = \frac{\sum_{i=1}^{\omega}\sum_{p=1}^{m_i-1}\sum_{q=p+1}^{m_i} f_i(p,q,j,r,s)}{\sum_{i=1}^{\omega}\sum_{p=1}^{m_i}\sum_{q=p}^{m_i} h_i(p,q,j)} \quad (12)$$

$$\hat{P}(N^j \to w^k) = \frac{\sum_{i=1}^{\omega} g_i(j,k)}{\sum_{i=1}^{\omega}\sum_{p=1}^{m_i}\sum_{q=p}^{m_i} h_i(p,q,j)} \quad (13)$$

The process of inside-outside algorithm repeats the parameters estimation, until the estimate probability of training corpus changes very little. If G_i is the grammar of the No. i step iteration in training (including rules probability), so the probability of corpus corresponding to model is guaranteed to increase without reduction, that is:

$$P(W|G_{i+1}) \geq P(W|G_i)$$

PCFG gives a new thinking way to build robust syntactic analysis, but PCFG has the following two reasons which restrict its application: at first the learning algorithm's convergence speed is very slow, for each sentence, each iteration time complexity in training is $O(m^3\, n^3)$, hereinto m is the length of sentence, n is the number of non-terminal signals in grammar. Secondly the algorithm convergence properties will become worsen sharply with the increase of the non-terminal signal number, local extremum problem is very serious. Therefore, many people improve the algorithm, the experiment in literature presents that PCFG grammar analyzer adopts ATIS corpus training, when the scale of corpus reaches 700 sentences, the iteration times is 75, the accuracy rate is 37.35% when training corpus only tags information, the accuracy of training corpus is 90.36% when doing superficial layer grammar analysis. The training of corpus in literature is WSJ after a superficial syntax analysis, when training sentences scale is 1095 sentences, the accuracy with 80 iterative times is: nodes accuracy rate is 90.22%, the accuracy rate of sentence analysis is 57.14%.

Last word: EM algorithm in the process of statistics natural language has wide range of applications, it is not directly maximizing or doing analog to the complicated posterior distribution, but based on the observation data adding some "potential data" to simplify calculation and complete a serial of simple maximizing or simulation. The characteristics of EM algorithm is simple and stable, especially each iteration can guarantee the logarithm likelihood function of observation data is monotonous without decrease, which can guarantee the likelihood function converge to a local maximum value point. But this algorithm has some pitfalls: first of all, EM algorithm is very sensitive to the setting of initial value, bad parameter initial values are easy to make the algorithm convergence value to reach some local optimization points; second, the convergence speed of EM algorithm is slow. Therefore, the training of model generally adopts the "offline" method, that is, after training model being qualified then doing application; this is against the real time process.

REFERENCES

Bilmes, J.A., 1997. A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and Hidden Markov models. Berkeley, CA: Technical Report ICSI-TR-97-021, International Computer Science Institute.

Christopher, D., 1999. Manning, Hinrich Schütze. Foundations of Statistical Natural Language Processing. The MIT Press, London, pp: 381-403.

Cutting, D., J. Kupiec, J. Pedersen and P. Sibun, 1992. A practical part-of-speech tagger. Proceedings of the Third Conference on Applied Natural Language Processing, ACL, Trento, Italy, pp: 133-140.

Rabiner, L.R., 1989. A tutorial on hidden Markov model and selected applications in speech recognition. Proc. IEEE, 77(2): 257-285.

Zoubin, G. and I.J. Michael, 1994. Supervised learning from incomplete data using an EM approach. Advances in Neural Information Processing Systems 6, (Proceedings of NIPS-93), pp: 120-127.

Zoubin, G. and I.J. Michael, 1995. Learning from Incomplete Data. AI Memo 1509, CBCL Paper 108, MIT.