Implementation of an Alternative Scaling for Baum-Welch Algorithm for Hidden Markov Model (HMM) in Apache Mahout

Manogna Vemulapati

Introduction

During each iteration of Baum-Welch algorithm, it computes forward and backward variables which are then used to estimate the model parameters for the next iteration. The computation of forward variables involves computing the sum of products of terms which are significantly less than 1. So the computation exponentially goes to zero as the length of the training sequence becomes large. Apache Mahout's implementation of HMM has a scaling mechanism based on logarithms to address this issue. The scaling implementation discussed here is described in [1] and is numerically more stable and also not based on the usage of logarithms.

Terminology

The HMM is specified as follows.

1. The set S is the set of hidden states $\{S_0, S_1, ..., S_{N-1}\}$ where N is the number of hidden states. The hidden state at time t is denoted by q_t .
2. V is the set of output symbols $\{V_0, V_1, ..., V_{M-1}\}$ where M is the number of output symbols. The output symbol at time t is denoted by O_t . An observation sequence of length T is denoted by $O_0 O_1 ... O_{T-1}$.
3. A is the state transition probability matrix where an element a_{ij} of the matrix is the probability of transitioning from hidden state S_i to hidden state S_j .
4. B is the emission probability matrix where an element $b_j(k)$ is the probability of outputting symbol V_k from hidden state S_j .
5. The initial probability matrix π is the $1 \times N$ matrix where element π_i is the probability of being in state S_i at time $t = 0$.

A HMM model is represented by $\lambda = (A, B, \pi)$.

Forward Variables

A forward variable $\alpha_t(i)$ is the probability of being in state S_i at time t and is defined as:

$$\alpha_t(i) = P(O_0 O_1 ... O_t, q_t = S_i | \lambda)$$

Forward variables are computed inductively as follows.

- Initialization:
  $\alpha_0(i) = \pi_i b_i(O_0), \ 0 \leqslant i \leqslant N-1$
- Induction:
  $$\alpha_{t+1}(j) = \left(\sum_{i=0}^{N-1} \alpha_t(i) a_{ij} \right) b_j(O_{t+1}), \ 0 \leqslant t \leqslant T-2, \ 0 \leqslant j N-1$$

Backward Variables

A backward variable $\beta_t(i)$ is the probability of the partial observation sequence $t+1$ to the end given state S_i at time t and the model λ and is defined as follows:

$$\beta_t(i)=P(O_{t+1}...O_{T-1}|q_t=S_i,\lambda)$$

Backward variables are computed inductively as follows.

- Initialization:

$$\beta_{T-1}(i)=1,0\leqslant i\leqslant N-1$$

- Induction:

$$\beta_t(i)=\sum_{j=0}^{N-1} a_{ij}b_j(O_{t+1})\beta_{t+1}(j),0\leqslant t\leqslant T-2,0\leqslant j\leqslant N-1.$$

Scaled Forward Variables

For an observation sequence $O_0O_1...O_{T-1}$ of length T there are T scaling factors where c_t is scaling factor at time step t . In other words, the scaling factors are independent of the hidden state but only depend on time step. The scaled forward variable is denoted by $\hat{\alpha}_t(i)$ and computed inductively as follows:

- Initialization:

$$\ddot{\alpha}_0(i)=\alpha_0(i)$$

$$c_0=\frac{1}{\sum_0^{N-1}\ddot{\alpha}_0(i)}$$

$$\hat{\alpha}_0(i)=c_0\ddot{\alpha}_0(i)$$

- Induction:

$$\ddot{\alpha}_t(i)=\sum_{j=0}^{N-1}\ddot{\alpha}_{t-1}(j)a_{ji}b_i(O_t)$$

$$c_t=\frac{1}{\sum_{i=0}^{N-1}\ddot{\alpha}_t(i)}$$

$$\hat{\alpha}_t(i)=c_t\ddot{\alpha}_t(i)$$

Scaled Backward Variables

For scaling backward variables, the same scaling factors that were computed for forward variables are used.

- Initialization:

$$\ddot{\beta}_{T-1}(i)=1$$

$$\hat{\beta}_{T-1}(i)=c_{T-1}\ddot{\beta}_{T-1}(i)$$

- Induction:

$$\ddot{\beta}_t(i) = \sum_{j=0}^{N-1} a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)$$

$$\hat{\beta}_t(i) = c_t \ddot{\beta}_t(i)$$

Baum-Welch Algorithm with Scaling

- The element a_{ij} of the state transition probability matrix is updated to $\bar{a}_{ij}$ as follows.

$$\bar{a}_{ij} = \frac{\sum_{t=0}^{T-2} \hat{\alpha}_t(i).a_{ij}.b_j(O_{t+1}).\hat{\beta}_{t+1}(j)}{\sum_{t=0}^{T-2} \hat{\alpha}_t(i).\hat{\beta}_t(i)/c_t}$$

- The element $b_j(k)$ of the emission probability matrix is updated to $\bar{b}_j(k)$ as follows.

$$\bar{b}_j(k) = \frac{\sum_{t=0, O_t=v_k}^{T-1} \hat{\alpha}_t(j).\hat{\beta}_t(j)/c_t}{\sum_{t=0}^{T-1} \hat{\alpha}_t(j).\hat{\beta}_t(j)/c_t}$$

- The element π_i of the initial probability matrix is updated to $\bar{\pi}_i$ as follows.

$$\bar{\pi}_i = \hat{\alpha}_0(i).\hat{\beta}_0(i)/c_0$$

- The log-likelihood of an observation sequence O given model λ is computed as follows.

$$\log\left[P(O|\lambda)\right] = -\sum_{t=0}^{T-1} \log\ c_t$$

Implementation in Apache Mahout

The source code is hosted in GitHub [2]. The new scaling method is called RESCALING.

References

1. Dawei Shen, “Some Mathematics for HMM”, http://courses.media.mit.edu/2010fall/mas622j/ProblemSets/ps4/tutorial.pdf.
2. Manogna, Vemulapati, "An alternative scaling method for Baum Welch for HMM", GitHub repository, https://github.com/manognavemulapati/mahout.git.
3. Stamp, Mark, “A Revealing Introduction to Hidden Markov Models”,https://www.cs.sjsu.edu/~stamp/RUA/HMM.pdf.