Expectation-Maximisation and Variational Approaches

Mark Gales

Lent 2011

Machine Learning for Language Processing: Lecture 5

MPhil in Advanced Computer Science
This lecture examines the training of generative classifiers with latent variables
- discriminative classifiers will be discussed in the next lecture

The models are to be trained using maximum likelihood estimation
- could use general approaches such as gradient descent
  BUT no guarantees of convergence, need to tune learning rate

This lecture will describe Expectation Maximisation (EM) and Variational EM
- elegantly handles the case when there are unobserved variables
- guaranteed convergence properties, no parameters to tune
Two scenarios need to be considered when training models

- **fully observed**: all variables observed (including “hidden” state in HMM)
- **partially observed**: only the observation sequence observed

For the fully observed case ML estimation performed by counting joint events

For partially observed case more interesting
- the unobserved state-sequence means it is not possible to simply count
Mixture Model Training

- **Bernoulli** mixture model, \( x_i \in \{0, 1\} \)

\[
P(x) = \sum_{m=1}^{M} P(c_m) P(x|c_m)
\]

\[
P(x|c_m) = \prod_{i=1}^{d} p_{mi}^{x_i} (1 - p_{mi})^{1-x_i}
\]

- **Maximum likelihood** estimate of parameters: \( \lambda = \{p_{11}, \ldots, p_{1d}, \ldots, p_{M1}, \ldots, p_{Md}\} \)
  - training data \( x_1, \ldots, x_n \) for the class of interest \( \omega \)

\[
\hat{\lambda} = \arg\max_{\lambda} \left\{ \prod_{\tau=1}^{n} P(x_{\tau}|\lambda) \right\} = \arg\max_{\lambda} \left\{ \sum_{\tau=1}^{n} \log \left( P(x_{\tau}|\lambda) \right) \right\}
\]

- If the indicator variable, \( q_{\tau} \) is **known** for each of the training example, \( x_{\tau} \),

\[
p_{mi} = \frac{1}{n_m} \sum_{\tau:q_{\tau}=c_m} x_{\tau i}, \quad n_m = \sum_{\tau:q_{\tau}=c_m} 1 \quad \text{BUT } q_{\tau} \text{ not known}
\]
Expectation Maximisation

• Rather than directly optimising the log-likelihood $\mathcal{L}(\lambda)$ where

$$\mathcal{L}(\lambda) = \sum_{\tau=1}^{n} \log (P(x_\tau | \lambda))$$

use an iterative approach and to ensure that for each iteration $k$

$$\mathcal{L}(\lambda^{[k+1]}) - \mathcal{L}(\lambda^{[k]}) \geq Q(\lambda^{[k+1]}; \lambda^{[k]}) - Q(\lambda^{[k]}; \lambda^{[k]}) \geq 0$$

where $Q(\lambda^{[k+1]}; \lambda^{[k]}) - Q(\lambda^{[k]}; \lambda^{[k]})$ is a lower-bound on $\mathcal{L}(\lambda^{[k+1]}) - \mathcal{L}(\lambda^{[k]})$

• If $Q(\lambda; \lambda^{[k]})$ can be simply optimised wrt $\lambda$, then iterate until convergence

Need to select an appropriate form for auxiliary function $Q(\lambda; \lambda^{[k]})$
Jensen’s Inequality

- A useful lower-bound is Jensen’s inequality.

\[ f \left( \sum_{m=1}^{M} \lambda_m x_m \right) \geq \sum_{m=1}^{M} \lambda_m f(x_m) \]

where \( f() \) is any concave function and

\[ \sum_{m=1}^{M} \lambda_m = 1, \quad \lambda_m \geq 0 \quad m = 1, \ldots, M \]

Take simple example to left:
Here \( c = (1 - \lambda)a + \lambda b \) and \( 0 \leq \lambda \leq 1 \)

\[ f(c) = f((1 - \lambda)a + \lambda b) \geq (1 - \lambda)f(a) + \lambda f(b) \]
Lower-Bound for Mixture Models

- Consider the change in the log likelihood:

\[
\mathcal{L}(\lambda^{[k+1]}) - \mathcal{L}(\lambda^{[k]}) = \sum_{i=1}^{n} \log \left( \frac{P(x_i | \lambda^{[k+1]})}{P(x_i | \lambda^{[k]})} \right)
\]

Expand mixture model and multiply numerator/denominator by \(P(c_m | x_i, \lambda^{[k]})\)

\[
\mathcal{L}(\lambda^{[k+1]}) - \mathcal{L}(\lambda^{[k]}) = \sum_{i=1}^{n} \log \left( \frac{1}{P(x_i | \lambda^{[k]})} \sum_{m=1}^{M} \left( \frac{P(c_m | x_i, \lambda^{[k]}) P(x_i, c_m | \lambda^{[k+1]})}{P(c_m | x_i, \lambda^{[k]})} \right) \right)
\]

Treating \(P(c_m | x_i, \lambda^{[k]})\) as \(\lambda_m\) for Jensen’s inequality (log() concave)

\[
\mathcal{L}(\lambda^{[k+1]}) - \mathcal{L}(\lambda^{[k]}) \geq \sum_{i=1}^{n} \sum_{m=1}^{M} P(c_m | x_i, \lambda^{[k]}) \log \left( \frac{P(x_i, c_m | \lambda^{[k+1]})}{P(x_i | \lambda^{[k]}) P(c_m | x_i, \lambda^{[k]})} \right)
\]
Definition of Auxiliary Function

- Recalling the desired change

\[ \mathcal{L}(\lambda^{[k+1]}) - \mathcal{L}(\lambda^{[k]}) \geq Q(\lambda^{[k+1]}, \lambda^{[k]}) - Q(\lambda^{[k]}, \lambda^{[k]}) \geq 0 \]

Comparing with the derivation from Jensen’s inequality

\[ Q(\lambda^{[k+1]}, \lambda^{[k]}) = \sum_{i=1}^{n} \sum_{m=1}^{M} P(c_m|x_i, \lambda^{[k]}) \log \left( P(x_i, c_m|\lambda^{[k+1]}) \right) \]

\[ = \sum_{i=1}^{n} \sum_{m=1}^{M} P(c_m|x_i, \lambda^{[k]}) \left( \log \left( P(c_m|\lambda^{[k+1]}) \right) + \log \left( P(x_i|c_m, \lambda^{[k+1]}) \right) \right) \]

- So to ensure that the log-likelihood doesn’t decrease at each iteration

\[ Q(\lambda^{[k+1]}, \lambda^{[k]}) \geq Q(\lambda^{[k]}, \lambda^{[k]}) \]
GMM Auxiliary Function Example

- Data generated from the following GMM:

\[ x \sim 0.4 \times \mathcal{N}(1, 1) + 0.6 \times \mathcal{N}(-1, 1) \]

Initial estimate of the model parameters is

\[ x^{(0)} \sim 0.4 \times \mathcal{N}(0.5, 1) + 0.6 \times \mathcal{N}(-1, 1) \]

- Plot shows the variation of the log-likelihood difference and auxiliary function difference as the estimate of the mean of component 1
  - auxiliary function difference always a lower-bound
  - peak of auxiliary function about 0.8
  - peak of log-likelihood function 1.0
  - gradient at current value (0.5) same for both
Mixture Model Training Procedure

- The overall procedure for training a mixture model is:
  1. initialise model parameters $\lambda^{[0]}$, $k = 0$
  2. compute component posteriors given parameters $\lambda^{[k]}$ and observation $x_i$

$$P(c_m|x_i, \lambda^{[k]}) = \frac{P(c_m|\lambda^{[k]})P(x_i|c_m, \lambda^{[k]})}{\sum_{j=1}^{M} P(c_j|\lambda^{[k]})P(x_i|c_j, \lambda^{[k]})}$$

These are then used to accumulate the sufficient statistics for $Q(\lambda; \lambda^{[k]})$

3. given the posterior derived sufficient statistics find

$$\lambda^{[k+1]} = \arg\max_{\lambda} \left\{ Q(\lambda; \lambda^{[k]}) \right\}$$

4. unless converged, let $k = k + 1$ goto (2)
Bernoulli Mixture Model Updates

- Now consider the training of the mixture of Bernoulli distribution
  - substituting the form into the auxiliary function (ignoring component prior)

\[
Q(\lambda; \lambda^{[k]}) = \sum_{m=1}^{M} \sum_{i=1}^{n} P(c_m | x_i, \lambda^{[k]}) \sum_{j=1}^{d} [x_{ij} \log(\lambda_{mj}) + (1 - x_{ij}) \log(1 - \lambda_{mj})]
\]

Differentiate this with respect to \( \lambda_{qr} \) gives

\[
\frac{\partial Q(\lambda, \lambda^{[k]})}{\partial \lambda_{qr}} = \sum_{i=1}^{n} P(c_q | x_i, \lambda^{[k]}) \left[ \frac{x_{ir}}{\lambda_{qr}} - \frac{(1 - x_{ir})}{(1 - \lambda_{qr})} \right]
\]

Equating this expression to zero to find new estimates \( \lambda^{[k+1]} \)

\[
(1 - \lambda_{qr}^{[k+1]}) \sum_{i=1}^{n} P(c_q | x_i, \lambda^{[k]}) x_{ir} = \lambda_{qr}^{[k+1]} \sum_{i=1}^{n} P(c_q | x_i, \lambda^{[k]}) (1 - x_{ir})
\]

Rearranging yields:

\[
\lambda_{mj}^{[k+1]} = \frac{\sum_{i=1}^{n} P(c_m | x_i, \lambda^{[k]}) x_{ij}}{\sum_{i=1}^{n} P(c_m | x_i, \lambda^{[k]})}
\]
Update for Component Prior

- Also need to find component prior $P(c_m|\lambda^{[k+1]})$ so maximise wrt $\lambda$

$$Q(\lambda; \lambda^{[k]}) = \sum_{i=1}^{n} \sum_{m=1}^{M} P(c_m|x_i, \lambda^{[k]}) \log (P(c_m|\lambda))$$

subject to the constraints: $\sum_{m=1}^{M} P(c_m|\lambda) = 1$, $P(c_m|\lambda) \geq 0$

- Use Lagrange optimisation for this constrained optimisation problem

$$P(c_m|\lambda^{[k+1]}) = \frac{1}{n} \sum_{i=1}^{n} P(c_m|x_i, \lambda^{[k]})$$
**General Form for EM**

- EM can be applied to a range of tasks (and latent variables)
  - consider a set of continuous latent variables, $Z$
  - introduce posterior distribution over latent variables, $Z$, $p(Z|X, \lambda)$

$$
\mathcal{L}(\lambda) = \mathcal{F}(q(Z, \lambda), \lambda) = \int q(Z, \lambda) \log \left( \frac{p(X, Z|\lambda)}{q(Z, \lambda)} \right) dZ
$$

$$
= \left\langle \log \left( \frac{p(X, Z|\lambda)}{q(Z, \lambda)} \right) \right\rangle_{q(Z, \lambda)}
$$

where $q(Z, \lambda) = p(Z|X, \lambda)$

- For any parameter values, e.g. $\tilde{\lambda}$, and associated posterior distribution $q(Z, \tilde{\lambda})$,

$$
\mathcal{L}(\lambda) \geq \mathcal{F}(q(Z, \tilde{\lambda}), \lambda) = \left\langle \log \left( \frac{p(X, Z|\lambda)}{q(Z, \tilde{\lambda})} \right) \right\rangle_{q(Z, \tilde{\lambda})}
$$

  - uses Jensen’s inequality to yield a lower-bound
  - equality only when $\tilde{\lambda} = \lambda$
General Form for EM (cont)

• Using the previous two expressions at iteration $k + 1$, find parameters $\lambda^{[k+1]}

$$\mathcal{L}(\lambda^{[k]}) = \mathcal{F} \left( q(Z, \lambda^{[k]}), \lambda^{[k]} \right) \leq \mathcal{F} \left( q(Z, \lambda^{[k]}), \lambda^{[k+1]} \right) \leq \mathcal{L}(\lambda^{[k+1]})$$

where $q(Z, \lambda^{[k]}) = p(Z|X, \lambda^{[k]})$

- **E-step:** $\mathcal{F} \left( q(Z, \lambda^{[k]}), \lambda^{[k]} \right) = \mathcal{L}(\lambda^{[k]})$ find $p(Z|X, \lambda^{[k]})$

- **M-step:** $\mathcal{F} \left( q(Z, \lambda^{[k]}), \lambda^{[k+1]} \right) \geq \mathcal{F} \left( q(Z, \lambda^{[k]}), \lambda^{[k]} \right)$ find parameters

• Iterate until convergence:

  – each iteration guaranteed not to decrease the likelihood
  – finds a **local** maximum of the likelihood
  – final solution depends on initial parameters $\lambda^{[0]}$
Variational EM

- Not always tractable to compute posterior distribution $p(Z|X, \lambda^{[k]})$
  - introduce a tractable approximation to this $q(Z)$, using Jensen’s inequality
    \[
    \mathcal{L}(\lambda) \geq \mathcal{F}(q(Z), \lambda) = \left\langle \log \left( \frac{p(X, Z|\lambda)}{q(Z)} \right) \right\rangle_{q(Z)}
    \]

- Iterations for Variational EM consists of:
  - E-step (approximate): $q^{[k]}(Z) = \text{argmax}_{q(Z)} \{ \mathcal{F}(q(Z), \lambda^{[k]}) \}$
  - M-step: $\lambda^{[k+1]} = \text{argmax}_{\lambda} \{ \mathcal{F}(q^{[k]}(Z), \lambda) \}$

- Though this makes the training tractable, not guaranteed to increase likelihood
  \[
  \mathcal{L}(\lambda^{[k]}) \geq \mathcal{F}(q^{[k]}(Z), \lambda^{[k]}) \leq \mathcal{F}(q^{[k]}(Z), \lambda^{[k+1]}) \leq \mathcal{L}(\lambda^{[k+1]})
  \]

- One standard form is the mean-field approximation where $q(Z) = \prod_{i=1}^{n} q_i(z_i)$