Expectation-Maximisation and Variational Approaches

Mark Gales

Lent 2011

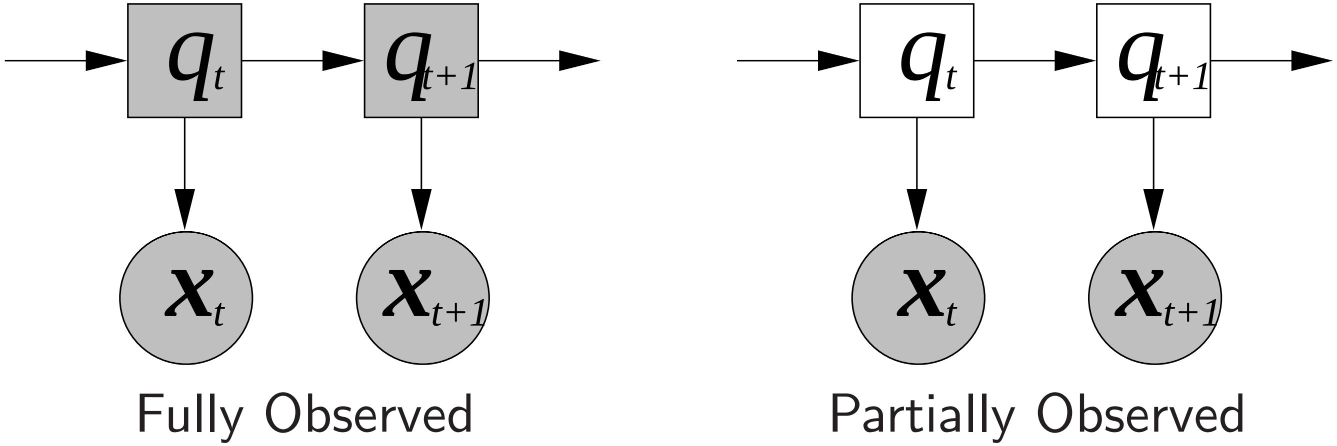
Machine Learning for Language Processing: Lecture 5

MPhil in Advanced Computer Science

Training Latent Variable Models

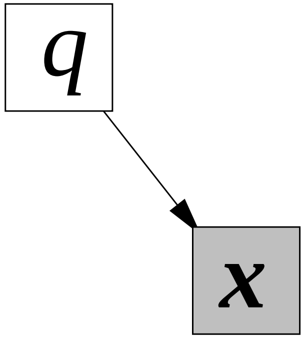
- This lecture examines the training of generative classifiers with latent variables
  - discriminative classifiers will be discussed in the next lecture
- The models are to be trained using maximum likelihood estimation
  - could use general approaches such as gradient descent
    BUT no guarantees 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

Fully and Partially Observed Training



- 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, $x_i \in \{0, 1\}$

$$P(\boldsymbol{x}) = \sum_{m=1}^{M} P(\mathsf{c}_m) P(\boldsymbol{x}|\mathsf{c}_m)$$

$$P(\boldsymbol{x}|\mathsf{c}_m) = \prod_{i=1}^{d} p_{mi}^{x_i}(1 - p_{mi})^{1-x_i}$$

- Maximum likelihood estimate of parameters: $\boldsymbol{\lambda} = \{p_{11}, \ldots, p_{1d}, \ldots, p_{M1}, \ldots, p_{Md}\}$
  - training data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ for the class of interest ω

$$\hat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \prod_{\tau=1}^{n} P(\boldsymbol{x}_\tau|\boldsymbol{\lambda}) \right\} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \sum_{\tau=1}^{n} \log \left(P(\boldsymbol{x}_\tau|\boldsymbol{\lambda}) \right) \right\}$$

- If the indicator variable, q_τ is known for each of the training example, $\boldsymbol{x}_\tau$,

$$p_{mi} = \frac{1}{n_m} \sum_{\tau : q_\tau = \mathsf{c}_m} x_{\tau i}, \quad n_m = \sum_{\tau : q_\tau = \mathsf{c}_m} 1 \quad \text{BUT } q_\tau \text{ not known}$$

Expectation Maximisation

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

$$\mathcal{L}(\boldsymbol{\lambda}) = \sum_{\tau=1}^{n} \log\left(P(\boldsymbol{x}_\tau|\boldsymbol{\lambda})\right)$$

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

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

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

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

Need to select an appropriate form for *auxiliary function* $\mathcal{Q}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{[k]})$

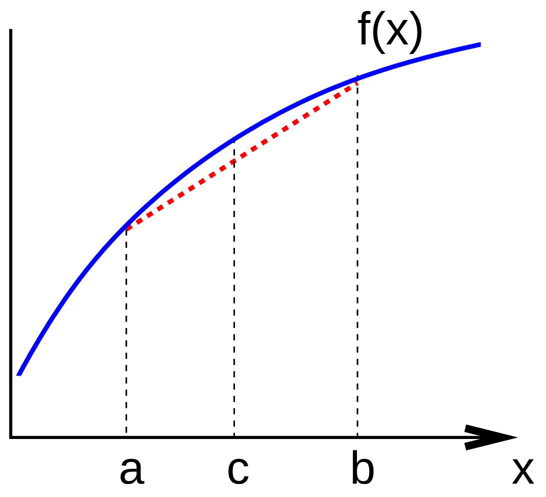
Jensen's Inequality

- 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

$$\sum_{m=1}^{M} \lambda_m = 1, \quad \lambda_m \geq 0 \;\; m = 1, \ldots, M$$



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

$$\begin{aligned} f(c) &= f((1-\lambda)a + \lambda b) \\ &\geq (1-\lambda)f(a) + \lambda f(b) \end{aligned}$$

Lower-Bound for Mixture Models

- Consider the change in the log likelihood:

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

Expand mixture model and multiply numerator/denominator by $P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]})$

$$\mathcal{L}(\boldsymbol{\lambda}^{[k+1]}) - \mathcal{L}(\boldsymbol{\lambda}^{[k]}) = \sum_{i=1}^{n} \log\left(\frac{1}{P(\boldsymbol{x}_i|\boldsymbol{\lambda}^{[k]})} \sum_{m=1}^{M}\left(\frac{P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]})P(\boldsymbol{x}_i, \mathsf{c}_m|\boldsymbol{\lambda}^{[k+1]})}{P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]})}\right)\right)$$

Treating $P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]})$ as λ_m for Jensen's inequality ($\log()$ concave)

$$\mathcal{L}(\boldsymbol{\lambda}^{[k+1]}) - \mathcal{L}(\boldsymbol{\lambda}^{[k]}) \geq \sum_{i=1}^{n} \sum_{m=1}^{M} P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]}) \log\left(\frac{P(\boldsymbol{x}_i, \mathsf{c}_m|\boldsymbol{\lambda}^{[k+1]})}{P(\boldsymbol{x}_i|\boldsymbol{\lambda}^{[k]})P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]})}\right)$$

Definition of Auxiliary Function

- Recalling the desired change

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

  Comparing with the derivation from Jensen's inequality

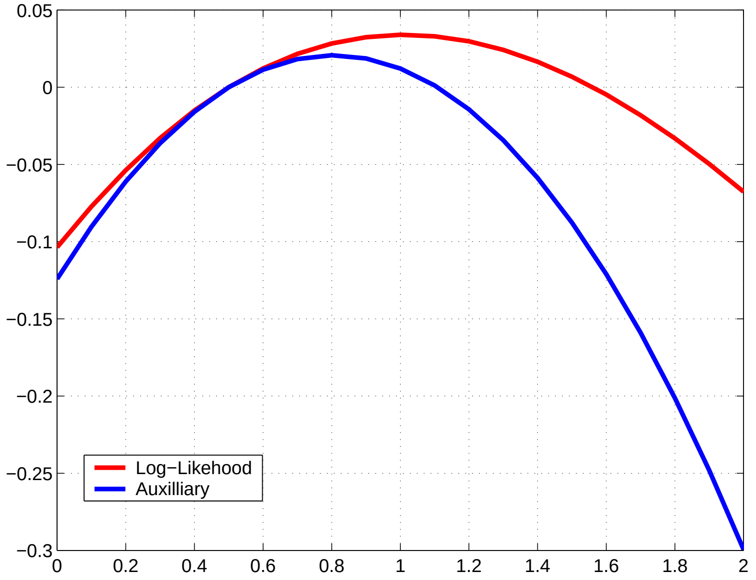
$$\mathcal{Q}(\boldsymbol{\lambda}^{[k+1]}; \boldsymbol{\lambda}^{[k]}) = \sum_{i=1}^{n} \sum_{m=1}^{M} P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]}) \log \left(P(\boldsymbol{x}_i, \mathsf{c}_m|\boldsymbol{\lambda}^{[k+1]}) \right)$$

$$= \sum_{i=1}^{n} \sum_{m=1}^{M} P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]}) \left(\log \left(P(\mathsf{c}_m|\boldsymbol{\lambda}^{[k+1]}) \right) + \log \left(P(\boldsymbol{x}_i|\mathsf{c}_m, \boldsymbol{\lambda}^{[k+1]}) \right) \right)$$

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

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

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 $\boldsymbol{\lambda}^{[0]}$, $k = 0$
2. compute component posteriors given parameters $\boldsymbol{\lambda}^{[k]}$ and observation $\boldsymbol{x}_i$

$$P(\mathsf{c}_m|\boldsymbol{x}_i, \boldsymbol{\lambda}^{[k]}) = \frac{P(\mathsf{c}_m|\boldsymbol{\lambda}^{[k]})P(\boldsymbol{x}_i|\mathsf{c}_m, \boldsymbol{\lambda}^{[k]})}{\sum_{j=1}^{M} P(\mathsf{c}_j|\boldsymbol{\lambda}^{[k]})P(\boldsymbol{x}_i|\mathsf{c}_j, \boldsymbol{\lambda}^{[k]})})$$

   These are then used to accumulate the sufficient statistics for $\mathcal{Q}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{[k]})$
3. given the posterior derived sufficient statistics find

$$\boldsymbol{\lambda}^{[k+1]} = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \left\{ \mathcal{Q}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^{[k]}) \right\}$$

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)

$$\mathcal{Q}(\boldsymbol{\lambda};\boldsymbol{\lambda}^{[k]}) = \sum_{m=1}^{M}\sum_{i=1}^{n} P(\mathsf{c}_m|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]}) \sum_{j=1}^{d}\left[x_{ij}\log(\lambda_{mj}) + (1-x_{ij})\log(1-\lambda_{mj})\right]$$

Differentiate this with respect to λ_{qr} gives

$$\frac{\partial\mathcal{Q}(\boldsymbol{\lambda},\boldsymbol{\lambda}^{[k]})}{\partial\lambda_{qr}} = \sum_{i=1}^{n} P(\mathsf{c}_q|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]})\left[\frac{x_{ir}}{\lambda_{qr}} - \frac{(1-x_{ir})}{(1-\lambda_{qr})}\right]$$

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

$$(1-\lambda_{qr}^{[k+1]})\sum_{i=1}^{n} P(\mathsf{c}_q|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]})x_{ir} = \lambda_{qr}^{[k+1]}\sum_{i=1}^{n} P(\mathsf{c}_q|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]})(1-x_{ir})$$

Rearranging yields: $\lambda_{mj}^{[k+1]} = \frac{\sum_{i=1}^{n} P(\mathsf{c}_m|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]})x_{ij}}{\sum_{i=1}^{n} P(\mathsf{c}_m|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]})}$

Update for Component Prior

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

$$\mathcal{Q}(\boldsymbol{\lambda};\boldsymbol{\lambda}^{[k]}) = \sum_{i=1}^{n}\sum_{m=1}^{M} P(\mathsf{c}_m|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]})\log\left(P(\mathsf{c}_m|\boldsymbol{\lambda})\right)$$

subject to the constraints: $\sum_{m=1}^{M} P(\mathsf{c}_m|\boldsymbol{\lambda}) = 1, \quad P(\mathsf{c}_m|\boldsymbol{\lambda}) \geq 0$

- Use Lagrange optimisation for this constrained optimisation problem

$$P(\mathsf{c}_m|\boldsymbol{\lambda}^{[k+1]}) = \frac{1}{n}\sum_{i=1}^{n} P(\mathsf{c}_m|\boldsymbol{x}_i,\boldsymbol{\lambda}^{[k]})$$

General Form for EM

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

$$\begin{aligned}\mathcal{L}(\boldsymbol{\lambda}) = \mathcal{F}\left(q(\boldsymbol{Z},\boldsymbol{\lambda}),\boldsymbol{\lambda}\right) &= \int q(\boldsymbol{Z},\boldsymbol{\lambda}) \log\left(\frac{p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\lambda})}{q(\boldsymbol{Z},\boldsymbol{\lambda})}\right) d\boldsymbol{Z} \\ &= \left\langle \log\left(\frac{p(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{\lambda})}{q(\boldsymbol{Z},\boldsymbol{\lambda})}\right)\right\rangle_{q(\boldsymbol{Z},\boldsymbol{\lambda})}\end{aligned}$$

  where $q(\boldsymbol{Z},\boldsymbol{\lambda}) = p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\lambda})$
- For any parameter values, e.g. $\tilde{\boldsymbol{\lambda}}$, and associated posterior distribution $q(\boldsymbol{Z},\tilde{\boldsymbol{\lambda}})$,

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

  - uses Jensen's inequality to yield a lower-bound
  - equality only when $\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}$

General Form for EM (cont)

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

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

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

  - E-step: $\mathcal{F}\left(q(\boldsymbol{Z},\boldsymbol{\lambda}^{[k]}),\boldsymbol{\lambda}^{[k]}\right) = \mathcal{L}(\boldsymbol{\lambda}^{[k]})$ find $p(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{\lambda}^{[k]})$
  - M-step: $\mathcal{F}\left(q(\boldsymbol{Z},\boldsymbol{\lambda}^{[k]}),\boldsymbol{\lambda}^{[k+1]}\right) \geq \mathcal{F}\left(q(\boldsymbol{Z},\boldsymbol{\lambda}^{[k]}),\boldsymbol{\lambda}^{[k]}\right)$ 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 $\boldsymbol{\lambda}^{[0]}$

Variational EM

- Not always tractable to compute posterior distribution $p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{\lambda}^{[k]})$
  - introduce a tractable approximation to this $q(\boldsymbol{Z})$, using Jensen's inequality

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

- Iterations for Variational EM consists of:
  - E-step (approximate): $q^{[k]}(\boldsymbol{Z}) = \mathrm{argmax}_{q(\boldsymbol{Z})}\left\{\mathcal{F}(q(\boldsymbol{Z}), \boldsymbol{\lambda}^{[k]})\right\}$
  - M-step: $\boldsymbol{\lambda}^{[k+1]} = \mathrm{argmax}_{\boldsymbol{\lambda}}\left\{\mathcal{F}(q^{[k]}(\boldsymbol{Z}), \boldsymbol{\lambda})\right\}$
- Though this makes the training tractable, not guaranteed to increase likelihood

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

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