Expectation-Maximization Algorithm and Applications

Eugene Weinstein

Courant Institute of Mathematical Sciences

Nov 14th, 2006

List of Concepts

- Maximum-Likelihood Estimation (MLE)
- Expectation-Maximization (EM)
- Conditional Probability
- Mixture Modeling
- Gaussian Mixture Models (GMMs)
- String edit-distance
- Forward-backward algorithms

Overview

- Expectation-Maximization
- Mixture Model Training
- Learning String Edit-Distance

One-Slide MLE Review

- Say I give you a coin with

$$P(heads) = \theta, P(tails) = 1 - \theta$$

- But I don't tell you the value of θ
- Now say I let you flip the coin n times
  - You get h heads and n-h tails
- What is the natural estimate of θ?
  - This is $\hat{\theta} = h/n$
- More formally, the likelihood of θ is governed by a binomial distribution: $P(\theta) = \binom{h}{n}\theta^h (1-\theta)^{n-h}$
  - Can prove $\hat{\theta}$ is the maximum-likelihood estimate of θ
  - Differentiate with respect to θ, set equal to 0

EM Motivation

- So, to solve any ML-type problem, we analytically maximize the likelihood function?
  - Seems to work for 1D Bernoulli (coin toss)
  - Also works for 1D Gaussian (find μ, σ^2)
- Not quite
  - Distribution may not be well-behaved, or have too many parameters
  - Say your likelihood function is a mixture of 1000 1000-dimensional Gaussians (1M parameters)
  - Direct maximization is not feasible
- Solution: introduce hidden variables to
  - Simplify the likelihood function (more common)
  - Account for actual missing data

Hidden and Observed Variables

- Observed variables: directly measurable from the data, e.g.
  - The waveform values of a speech recording
  - Is it raining today?
  - Did the smoke alarm go off?
- Hidden variables: influence the data, but not trivial to measure
  - The phonemes that produce a given speech recording
  - $P(\text{rain today} \mid \text{rain yesterday})$
  - Is the smoke alarm malfunctioning?

Expectation-Maximization

- Model dependent random variables:
  - Observed variable x
  - Unobserved (hidden) variable y that generates x
- Assume probability distributions: $P_\theta(x), P_\theta(y)$
  - θ represents set of all parameters of distribution
- Repeat until convergence
  - E-step: Compute expectation of $\log P_\theta(y, x)$

    $$Q(\theta, \theta') = \sum_y P_{\theta'}(y|x) \log P_\theta(y, x)$$

    (θ',θ: old, new distribution parameters)
  - M-step: Find θ that maximizes Q

Conditional Expectation Review

- Let X, Y be r.v.'s drawn from the distributions $P(x)$ and $P(y)$
- Conditional distribution given by: $P(y|x) = \frac{P(y,x)}{P(x)}$
- Then $\mathbb{E}[Y] = \sum_y P(y)y$
- For function $h(Y)$: $\mathbb{E}[h(Y)] = \sum_y P(y)h(y)$
- Given a particular value of X (X=x):

$$\mathbb{E}[h(Y)|x] = \sum_y P(y|x)h(y)$$

Maximum Likelihood Problem

- Want to pick θ that maximizes the log-likelihood of the observed (x) and unobserved (y) variables given
  - Observed variable x
  - Previous parameters θ'
- Conditional expectation of $\log P_\theta(y, x)$ given x and θ' is

$$\begin{aligned}\mathbb{E}[\log P(y, x|\theta)|x, \theta'] &= \sum_y P(y|x, \theta') \log P(y, x|\theta) \\ &= \sum_y P_{\theta'}(y|x) \log P_\theta(y, x)\end{aligned}$$

EM Derivation

- Lemma (Special case of Jensen's Inequality): Let $p(x)$, $q(x)$ be probability distributions. Then

$$\sum_x p(x) \log p(x) \geq \sum_x p(x) \log q(x)$$

- Proof: rewrite as: $\sum_x p(x) \log \frac{q(x)}{p(x)} \leq 0$

$$\log x \leq x - 1 \quad \Rightarrow$$

$$\sum_x p(x) \log \frac{q(x)}{p(x)} \leq \sum_x p(x) \left(\frac{q(x)}{p(x)} - 1 \right) = \sum_x q(x) - \sum_x p(x) = 0$$

- Interpretation: relative entropy non-negative

EM Derivation

- EM Theorem:
  - If $\sum_y P_{\theta'}(y|x) \log P_\theta(y,x) > \sum_y P_{\theta'}(y|x) \log P_{\theta'}(y,x)$
  - then $P_\theta(x) > P_{\theta'}(x)$
- Proof:

$$\log P_\theta(x) - \log P_{\theta'}(x)$$
$$= \sum_y P_{\theta'}(y|x) \log P_\theta(x) - \sum_y P_{\theta'}(y|x) \log P_{\theta'}(x) = \ldots$$

- By some algebra and lemma,

$$\ldots \geq \sum_y P_{\theta'}(y|x) \log P_\theta(y,x) - \sum_y P_{\theta'}(y|x) \log P_{\theta'}(y,x)$$

- So, if this quantity is positive, so is $\log P_\theta(x) - \log P_{\theta'}(x)$
- $\log P_\theta(x) > \log P_{\theta'}(x) \quad \Rightarrow \quad P_\theta(x) > P_{\theta'}(x)$

EM Summary

- Repeat until convergence
  - E-step: Compute expectation of $\log P_\theta(y, x)$

    $$Q(\theta, \theta') = \sum_y P_{\theta'}(y|x) \log P_\theta(y, x)$$

    (θ',θ : old, new distribution parameters)
  - M-step: Find θ that maximizes Q
- EM Theorem:
  - If $\sum_y \log P_\theta(y, x) P_{\theta'}(y|x) > \sum_y \log P_{\theta'}(y, x) P_{\theta'}(y|x)$
  - then $P_\theta(x) > P_{\theta'}(x)$
- Interpretation
  - As long as we can improve the expectation of the log-likelihood, EM improves our model of observed variable x
  - Actually, it's not necessary to maximize the expectation, just need to make sure that it increases – this is called "Generalized EM"

EM Comments

- In practice, the x is series of data points $x_1, \ldots, x_n$
  - To calculate expectation, can assume i.i.d and sum over all points: $Q(\theta, \theta') = \sum_{i=1}^{n} \sum_{y} P_{\theta'}(y|x_i) \log P_{\theta}(y, x_i)$
- Problems with EM?
  - Local maxima
  - Need to bootstrap training process (pick a θ)
- When is EM most useful?
  - When model distributions easy to maximize (e.g., Gaussian mixture models)
- EM is a meta-algorithm, needs to be adapted to particular application

Overview

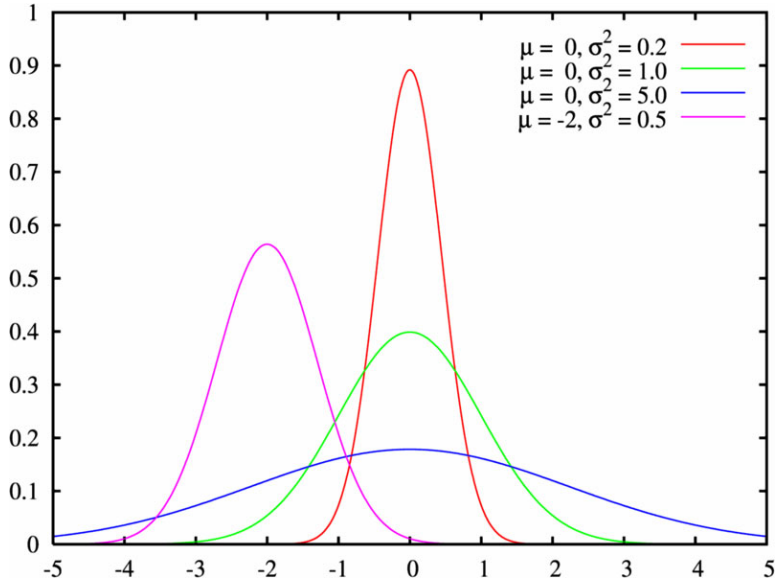
- Expectation-Maximization
- Mixture Model Training
- Learning String Distance

EM Applications: Mixture Models

- Gaussian/normal distribution
  - Parameters: mean μ and variance σ^2

$$G_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

  - In the multi-dimensional case, assume isotropic Gaussian: same variance in all dimensions
  - We can model arbitrary distributions with density mixtures

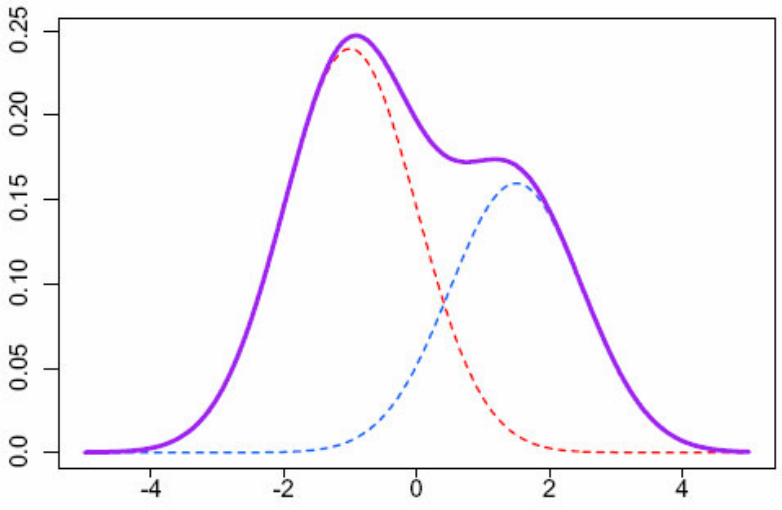


Density Mixtures

- Combine m elementary densities to model a complex data distribution

$$P_\Theta(x) = \sum_{i=1}^{m} \alpha_i P_{\theta_i}(x) \qquad \Theta = (\alpha_1, \ldots, \alpha_m, \theta_1, \ldots, \theta_m)$$

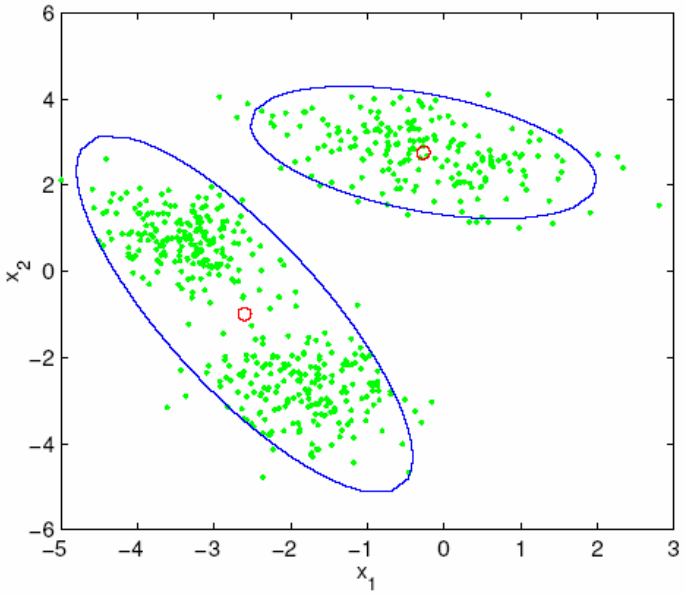
- kth Gaussian parametrized by $\theta_k = \{\mu_k, \sigma_k\}$

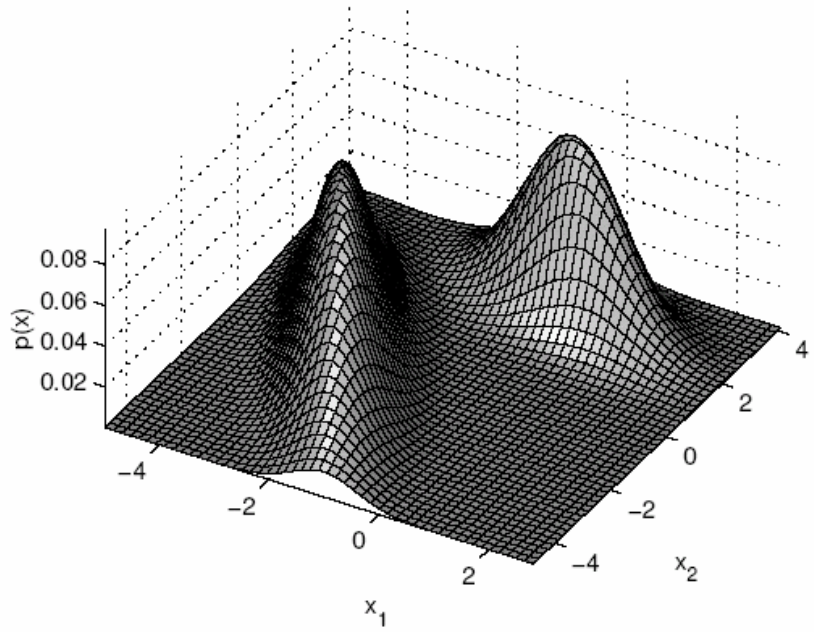


Density Mixtures

- Combine m elementary densities to model a complex data distribution

$$P_\Theta(x) = \sum_{i=1}^{m} \alpha_i P_{\theta_i}(x) \qquad \Theta = (\alpha_1, \ldots, \alpha_m, \theta_1, \ldots, \theta_m)$$





Density Mixtures

- Combine m elementary densities to model a complex data distribution

$$P_\Theta(x) = \sum_{i=1}^{m} \alpha_i P_{\theta_i}(x) \qquad \Theta = (\alpha_1, \ldots, \alpha_m, \theta_1, \ldots, \theta_m)$$

- Log-likelihood function of the data x given Θ :

$$\log P_\Theta(x) = \log \left(\sum_{i=1}^{m} \alpha_i P_{\theta_i}(x) \right)$$

- Log of sum – hard to optimize analytically!
- Instead, introduce hidden variable y
  - $y = k : x$ generated by Gaussian k

    $\log P_\Theta(x, y) = \log \left(\alpha_y P_{\theta_y}(x) \right)$
  - EM formulation: maximize

    $$Q(\Theta, \Theta') = \sum_y P_{\theta'_y}(y|x) \log \left(\alpha_y P_{\theta_y}(x) \right)$$

Gaussian Mixture Model EM

- Goal: maximize $Q(\Theta, \Theta') = \sum_y P_{\theta'_y}(y|x) \log\left(\alpha_y P_{\theta_y}(x)\right)$
- n (observed) data points: $x_1, \ldots, x_n$
- n (hidden) labels: $y_1, \ldots, y_n$
  - $y_i = k$: x_i generated by Gaussian k
- Several pages of math later, we get:
- E step: compute likelihood of $y_i = k$

$$\Lambda_{i,k} = P_{\Theta'}(y_i = k|x_i) = \frac{\alpha'_k P_{\theta'_k}(x_i)}{P_{\Theta'}(x_i)} = \frac{\alpha'_k P_{\theta'_k}(x_i)}{\sum_{j=1}^m \alpha_j P_{\theta'_j}(x_i)}$$

- M step: update α_k, μ_k, σ_k for each Gaussian k=1..m

$$\alpha_k = \frac{1}{n}\sum_{i=1}^n \Lambda_{i,k} \qquad \mu_k = \frac{\sum_{i=1}^n x_i \Lambda_{i,k}}{\sum_{i=1}^n \Lambda_{i,k}} \qquad \sigma_k^2 = \frac{\sum_{i=1}^n \Lambda_{i,k} ||x_i - \mu_k||^2}{\sum_{i=1}^n \Lambda_{i,k}}$$

GMM-EM Discussion

- Summary: EM naturally applicable to training probabilistic models
- EM is a generic formulation, need to do some hairy math to get to implementation
- Problems with GMM-EM?
  - Local maxima
  - Need to bootstrap training process (pick a θ)
- GMM-EM applicable to enormous number of pattern recognition tasks: speech, vision, etc.
- Hours of fun with GMM-EM

Overview

- Expectation-Maximization
- Mixture Model Training
- Learning String Distance

String Edit-Distance

- Notation: $x_i^j = x[i..j], x_i = x[i], x^i = x[1..i]$
- Operate on two strings: $x^T \in A^T, y^V \in B^V$
- Edit-distance: transform one string into another using
  - Substitution: **k**itten → **b**itten, cost $c(x_i, y_j)$
  - Insertion: cop → **cr**op, cost $c(\epsilon, y_j)$
  - Deletion: **l**earn → earn, cost $c(x_i, \epsilon)$
- Can compute efficiently recursively

$$d_c(x^t, y^v) = \min \left\{ \begin{array}{l} c(x_y, y_v) + d_c(x^{t-1}, y^{v-1}), \\ c(x_y, \epsilon) + d_c(x^{t-1}, y^v), \\ c(\epsilon, y_v) + d_c(x^t, y^{v-1}) \end{array} \right\}$$

Stochastic String Edit-Distance

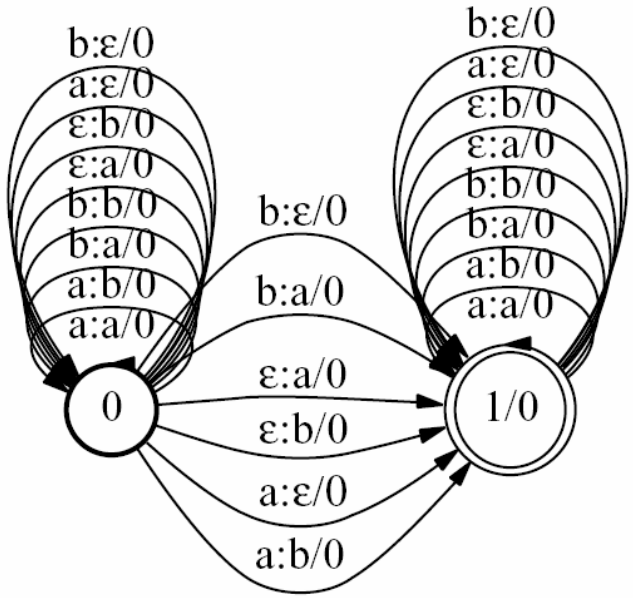
- Instead of setting costs, model edit operation sequence as a random process
- Edit operations selected according to a probability distribution

$$E = \{E_s, E_d, E_i\} \qquad \delta : E \cup \{\#\} \rightarrow [0, 1]$$

- For edit operation sequence $z_1, \ldots, z_n$
- View string edit-distance as
  - memoryless (Markov): $\delta(z_i|z_{i-1}) = \delta(z_i)$
  - stochastic: random process according to $\delta(\cdot)$ is governed by a true probability distribution
  - transducer: $\phi = \langle A, B, \delta \rangle$

Edit-Distance Transducer

- Arc label a:b/0 means input a, output b and weight 0
- Assume $A = B = \{a, b\}$



Two Distances

- Define yield of an edit sequence $\nu(z^n\#)$ as the set of all strings $\langle x,y\rangle$ such that $z^n\#$ turns x into y
- Viterbi edit-distance: negative log-likelihood of most likely edit sequence

$$d^v_\phi(x^T, y^V) = -\log \operatorname{argmax}_{\{z^n:\nu(z^n)=\langle x^T,y^V\rangle\}} P_\phi(z^n)$$

- Stochastic edit-distance: negative log-likelihood of all edit sequences from x to y

$$d^s_\phi(x^T, y^V) = -\log P(x^T, y^V|\phi) = -\log \sum_{\{z^n:\nu(z^n)=\langle x^T,y^V\rangle\}} P_\phi(z^n)$$

Evaluating Likelihood

- Viterbi: $d^v_\phi(x^T, y^V) = -\log \operatorname{argmax}_{\{z^n : \nu(z^n) = \langle x^T, y^V \rangle\}} P_\phi(z^n)$
- Stochastic:

$$d^s_\phi(x^T, y^V) = -\log P(x^T, y^V | \phi) = -\log \sum_{\{z^n : \nu(z^n) = \langle x^T, y^V \rangle\}} P_\phi(z^n)$$

- Both require calculation of $P_\phi(z^n)$ over all possible edit sequences
  - 3^n possibilities (three edit operations)
- However, memoryless assumption allows us to compute likelihood efficiently
- Use the forward-backward method!

Forward

- Evaluation of forward probabilities $\alpha_{t,v}$: likelihood of picking an edit sequence that generates the prefix pair $\langle x_t, y_v \rangle$
- Memoryless assumption allows efficient recursive computation: $O(T \cdot V)$

FORWARD-EVALUATE(x^T,y^V,ϕ)

1. $\alpha_{0,0} := 1;$
2. For $t = 0 \ldots T$
3. For $v = 0 \ldots V$
4. if $(v > 1 \vee t > 1)$ [$\alpha_{t,v} := 0;$]
5. if $(v > 1)$ [$\alpha_{t,v}$ += $\delta(\epsilon, y_v)\alpha_{t,v-1};$]
6. if $(t > 1)$ [$\alpha_{t,v}$ += $\delta(x_t, \epsilon)\alpha_{t-1,v};$]
7. if $(v > 1 \wedge t > 1)$ [$\alpha_{t,v}$ += $\delta(x_t, y_v)\alpha_{t-1,v-1};$]
8. $\alpha_{T,V}$ *= $\delta(\#);$
9. return(α);

Backward

- Evaluation of backward probabilities $\beta_{t,v}$: likelihood of picking an edit sequence that generates the suffix pair $\langle x_{t+1}^{T}, y_{v+1}^{V} \rangle$
- Memoryless assumption allows efficient recursive computation: $O(T \cdot V)$

BACKWARD-EVALUATE(x^T,y^V,ϕ)

1. $\beta_{T,V} := \delta(\#)$;
2. for $t = T \ldots 0$
3. for $v = V \ldots 0$
4. if $(v < V \vee t < T)$ [$\beta_{t,v} := 0$;]
5. if $(v < V)$ [$\beta_{t,v}$ += $\delta(\epsilon, y_{v+1})\beta_{t,v+1}$;]
6. if $(t < T)$ [$\beta_{t,v}$ += $\delta(x_{t+1}, \epsilon)\beta_{t+1,v}$;]
7. if $(v < V \wedge t < T)$ [$\beta_{t,v}$ += $\delta(x_{t+1}, y_{v+1})\beta_{t+1,v+1}$;]
8. return(β);

EM Formulation

- Edit operations selected according to a probability distribution

  $E = \{E_s, E_d, E_i\} \quad \delta : E \cup \{\#\} \rightarrow [0, 1]$

- So, EM has to update δ based on occurrence counts of each operation (similar to coin-tossing example)
- Idea: accumulate expected counts from forward, backward variables
- $\gamma(z)$: expected count of edit operation z

EM Details

- $\gamma(z)$: expected count of edit operation z
- e.g, P(after reading $\langle x_{t-1}, y_v \rangle$, we emit a deletion operation $\langle x_t, \epsilon \rangle$) $= \alpha_{t-1,v}\delta(x_t, \epsilon)\beta_{t,v}/\alpha_{T,V}$

EXPECTATION-STEP(x^T,y^V,ϕ,γ,λ)

1. $\alpha :=$ FORWARD-EVALUATE(x^T,y^V,ϕ);
2. $\beta :=$ BACKWARD-EVALUATE(x^T,y^V,ϕ);
3. if ($\alpha_{T,V} = 0$) [return;]
4. $\gamma(\#)$ += λ;
5. for $t = 0 \ldots T$
6. for $v = 0 \ldots V$
7. if ($t > 0$) [$\gamma(x_t, \epsilon)$ += $\lambda\alpha_{t-1,v}\delta(x_t, \epsilon)\beta_{t,v}/\alpha_{T,V}$;]
8. if ($v > 0$) [$\gamma(\epsilon, y_v)$ += $\lambda\alpha_{t,v-1}\delta(\epsilon, y_v)\beta_{t,v}/\alpha_{T,V}$;]
9. if ($t > 0 \wedge v > 0$) [$\gamma(x_t, y_v)$ += $\lambda\alpha_{t-1,v-1}\delta(x_t, y_v)\beta_{t,v}/\alpha_{T,V}$;]

MAXIMIZATION-STEP(ϕ,γ)

1. $N := \gamma(\#)$;
2. forall z in E [N += $\gamma(z)$;]
3. forall z in E [$\delta(z) := \gamma(z)/N$;]
4. $\delta(\#) := \gamma(\#)/N$;

References

- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society B, 39(1), 1977 pp. 1-38.
- C. F. J. Wu, On the Convergence Properties of the EM Algorithm, The Annals of Statistics, 11(1), Mar 1983, pp. 95-103.
- F. Jelinek, *Statistical Methods for Speech Recognition*, 1997
- M. Collins, The EM Algorithm, 1997
- J. A. Bilmes, A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, Technical Report, University of Berkeley, TR-97-021, 1998
- E. S. Ristad and P. N. Yianilos, Learning string edit distance, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(2), 1998, pp. 522-532.
- L.R. Rabiner. A tutorial on HMM and selected applications in speech recognition, In Proc. IEEE, 77(2), 1989, pp. 257-286.
- A. D'Souza, Using EM To Estimate A Probablity [sic] Density With A Mixture Of Gaussians
- M. Mohri. Edit-Distance of Weighted Automata, in Proc. Implementation and Application of Automata, (CIAA) 2002, pp. 1-23
- J. Glass, Lecture Notes, MIT class 6.345: Automatic Speech Recognition, 2003
- Carlo Tomasi, Estimating Gaussian Mixture Densities with EM – A Tutorial, 2004
- Wikipedia