Some Studies of Expectation Maximization Clustering Algorithm to Enhance Performance

D. J. Nagendra Kumar[1], J. V. R. Murthy
[1]nagendrakumardj@yahoo.co.in

Abstract : *Expectation Maximization (EM) is an efficient mixture-model based clustering method. In this paper, authors made an attempt to scale-up the algorithm, by reducing the computation time required for computing quadratic term, without sacrificing the accuracy. Probability density function (pdf) is to be calculated in EM, which involves evaluating quadratic term calculation. Three recursive approaches are introduced for quadratic term computation. As per our observation, the standard EM needs* $O(d^2)$ *computations for quadratic term computation, where* d *is number of dimensions. The proposed recursive EM approaches are with time complexity of* $O(d^2/2)$ *for the quadratic term computation.*

Keywords: *Expectation Maximization — Quadratic Term — Speed-up — Lower Triangular Canonical Form - Forward Substitution*

1. Introduction

Clustering is the division of data into groups of similar objects and plays an important role in broader range of applications ranging from information retrieval to CRM [2].

Expectation maximization (EM) is a well-known algorithm, proposed by Demster et al., [7] used for clustering in the context of mixture models. Due to its wide adaptability and applicability, in literature, we come across several variants and its applications. For example, starting from feature selection[16], adjusting the parameters

of the neural network architecture in the robot dynamic domain[13], for Text Mining applications like text classifications[19][23][27] and for Machine Translation [17][38] etc.

Hierarchical variant of EM[35] is used simultaneously to learn the main directions of the planar structures, correct the position, and orientation of planes using, for image segmentation[24], for various motion estimation frameworks and variants of it have been used in multi frame super resolution restoration methods which combine motion estimation[4], tracking a complex articulated object like a human hand[9], to perform principal component analysis on given data[34] [29] [32], for joint channel estimation and data detection in communication system[8]. For Segmentation of overlapped Nuclei in Biomedical engineering[14], user modeling in web usage mining[21] and parametric spectrogram modeling of speech in noisy environment[28].

Deterministic Annealing EM (DAEM) operates on Generalized Dirichlet Multinomial Distributions has various applications like - spatial color image database indexing, hand written digit recognition and text document clustering[5]. EM in Bayesian Knowledge Tracing (KT) models is employed by the cognitive tutors in order to determine student knowledge based on four parameters: learn rate, prior, guess and slip[25].

In [7] Dempster et. al. laid a solid foundation to the theory of EM clustering algorithm for computing maximum likelihood estimates for the incomplete data. EM results in closer approximations to local optimum, rather than global optimum.

As per [18], EM has favorable convergence properties with an automatic satisfaction of constraints. It performs soft assignments, adopts soft pruning mechanism, and accommodates model-based clustering formulation. Seam[4] mentioned that EM has become a popular tool in statistical estimation problems, that involves incomplete data or problems that can be posed in a similar form e.g. mixture estimation. In general the EM affects/suffers with the following:

1. Sensitivity to the selection of initial parameters,
2. Effect of singular covariance matrix,
3. Possibility of convergence to a local optimum, and
4. Slow convergence rate[40]

As the EM algorithm is more adaptable, in literature, several authors made an attempt to speed up the algorithm[22] [12].

[26] proposed TRUST-TECH-Based EM algorithm for learning mixture models from multivariate data to reduce the sensitivity of initial point selection. [11] demonstrates a Scalable clustering algorithm for large databases to summarize data into subclusters and then generate Gaussian mixtures from their data summaries. In [38] Jason Wolfe et. al. proposed a modified E-step, speeding-up and scaling-up Distributed EM, suitable to support for very large datasets that would be too large to fit in the memory of a single machine. In[15] Wojtek Kowalczyk et. al. demonstrates the usage of Gossip-based Distributed EM algorithm for large databases. Bilmes [3] proposed an EM based parameter estimation procedure, aimed to find the parameters of a mixture of Gaussian densities and parameters of a Hidden Markov Model (HMM) for both discrete and Gaussian mixture models. Convergence properties of EM are studied in [39] [1].

[30] demonstrates the usage of Tridiagonalization and diagonalization approaches to speed up EM. Incremental EM, Lazy EM [33] and kd-tree based EM [37] are proposed to accelerate EM clustering for large databases. [12] proposed CLEM to speed-up EM algorithm for clustering categorical data. As per [31], in certain cases, Expectation-Conjugate-Gradient(ECG) and Hybrid EM-ECG substantially outperform standard EM in terms of speed of convergence.

As per [20] observation, with respect to quality of models, Soft assignment EM outperforms the "winner take all" version of the EM (also called *Classification EM (CEM)*) and model-based hierarchical agglomerative clustering (HAC). Moreover, CEM is the fastest of all the three algorithms.

As EM is useful for various applications, here, we attempted to speed-up Expectation step by reducing the time taken for computing quadratic term.

2. EM Algorithm

In [10], EM procedure begins with an initial estimate of the parameter vector and iteratively rescores the patterns against the mixture density produced by the parameter vector. The rescored patterns are then used to update the parameter estimates. The scores of the patterns can be viewed as hints at the cluster of the pattern. Those patterns, placed in a particular component, would therefore be viewed as belonging to the same cluster.

The EM algorithm adopted here is from [6]. In this work we apply EM clustering to mixtures of gaussian distributions, since any distribution can be effectively

approximated by a mixture of Gaussians. Each cluster (population) is modeled by a d-dimensional Gaussian probability distribution. The multi-dimensional Gaussian distribution for cluster $l, l = 1,...,k,$ is parameterized by the d -dimensional mean column vector M_l and $d \times d$ covariance matrix Σ_l:

$$p(X \mid l) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma_l \mid}} e^{-\frac{1}{2}(X-M_l)^T (\Sigma_l)^{-1}(X-M_l)} \quad (1)$$

where X is sample column vector, the superscript ^{T} indicates transpose to a column vector, $|\Sigma_l|$ is the determinant of Σ_l and Σ_l^{-1} is its matrix inverse. The mixture model probability density function is:

$$p(X) = \sum_{l=1}^{k} w_l p(X \mid l). \quad (2)$$

The coefficient w_l is the weight corresponding the fraction of the database represented by the cluster l. The given data record X is a member of each of the k clusters with different probabilitis of membership. Probability in population l is given by:

$$p(l \mid X) = \frac{w_l p(X \mid l)}{p(X)}.$$

The EM algorithm starts with initializing mixture model parameters. Every EM iteration provides a non-decreasing log-likelihood of the model. The process is repeated until the log-likelihood of the mixture model at the previous iteration is sufficiently close to the log-likelihood of the current model or a fixed number of iterations are exhausted.

The algorithm proceeds as follows for the Gaussian mixture model:

1. Initialize the mixture model parameters, set current iteration $j = 0$: w_l, M_l and $\Sigma_l, l = 1,...,k$.
2. Having mixture model parameters at iteration j, update them as follows:

   For each database record $X_t, t = 1,...,m,$ compute the membership probability of X_t in each cluster:

$$p(l \mid X_t) = \frac{w_l p(X_t \mid l)}{p(X_t)}, l = 1,..., k. \quad (3)$$

3. Update mixture model parameters:

$$w_l = \frac{1}{m}\sum_{t=1}^{m} p(l \mid X_t), \tag{4}$$

$$M_l = \frac{\sum_{t=1}^{m} X_t\, p(l \mid X_t)}{\sum_{t=1}^{m} p(l \mid X_t)}, \tag{5}$$

$$\Sigma_l = \frac{\sum_{t=1}^{m} p(l \mid X_t)(X_t - M_l)(X_t - M_l)^T}{\sum_{t=1}^{m} p(l \mid X_t)}, l = 1,..., k. \tag{6}$$

4. Calculate the log-likelihood value,

$$llh_j = \sum_{t=1}^{m} log(p(X_t)) = \sum_{t=1}^{m} log(\sum_{l=1}^{k} w_l . p(X_t \mid l)) \tag{7}$$

5. If either the difference between log-likelihood values of present and last iterations is small, $| llh_j - llh_{j+1} | \leq \varepsilon$ or the maximum number of iterations are exhausted, $j = 100$, then stop. Otherwise set $j = j + 1$ and goto step 2.

3. Quadratic Term and Techniques to speedup its calculation

The equation $(X - M_l)^T \Sigma_l^{-1} (X - M_l)$ is known as quadratic term. For every data sample, the probability density function (pdf) is to be calculated, which in turn requires to calculate the quadratic term. N. B. Venkateswarlu et. al. introduces some methods useful in Machine Learning Classification to reduce the number of computations while classifying a sample by using fewer computations in calculating the pdf of each group [36]. The quadratic term in the pdf is presented to be a montonically increasing sum of squares by representing the inverse covariance matrix in Lower Triangular Canonical Form (LTCF).

The computational complexity of LTCF in calculating quadratic terms recursively is identified to be half of the standard quadratic term computation.

In this paper one recursive approach and two approaches based on LTCF are used for quadratic term computation in EM clustering algorithm. The quadratic term values of the first n features can be used to calculate the quadratic term values of the first $n+1(n+1 \leq d)$ features iteratively. Computational time required for the proposed EM algorithms are compared with the standard EM algorithm. Performance of these algorithms are studied by applying EM algorithm on Synthetic data varying dimensionality and number of gaussian distributions.

3.1 Recursive Approach

We partition Σ_{n+1} in the following manner:

$$\Sigma_{n+1} = \begin{bmatrix} \Sigma_n & U_{n+1} \\ U_{n+1}^T & \sigma_{n+1,n+1} \end{bmatrix}$$

where

$$\Sigma_n = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} = [\sigma_{ij}]$$

and

$$U_{n+1} = \begin{bmatrix} \sigma_{n+1,1} & \sigma_{n+1,2} & \cdots & \sigma_{n+1,n} \end{bmatrix}$$

Mathematically, we can show that

$$\Sigma_{n+1}^{-1} = \begin{bmatrix} \Sigma_n^{-1} + \alpha_n \Sigma_n^{-1} U_{n+1} U_{n+1}^T \Sigma_n^{-1} & -\alpha_n \Sigma_n^{-1} U_{n+1} \\ -\alpha_n U_{n+1}^T \Sigma_n^{-1} & \alpha_n \end{bmatrix}$$

where

$$\alpha_n = \frac{1}{\sigma_{n+1,n+1} - U_{n+1}^T \Sigma_n^{-1} U_{n+1}}.$$

In the above equations, the subscript n is used to denote the first n features. By setting $Z_n = X_n - M_n$, a recursive form of the quadratic term can be represented as:

$$Q_{n+1}(x) = (X_{n+1} - M_{n+1})^T \Sigma_{n+1}^{-1} (X_{n+1} - M_{n+1})$$

$$= \begin{bmatrix} Z_n^T & z_{n+1} \end{bmatrix} \begin{bmatrix} \Sigma_n^{-1} + \alpha_n \Sigma_n^{-1} U_{n+1} U_{n+1}^T \Sigma_n^{-1} & -\alpha_n \Sigma_n^{-1} U_{n+1} \\ -\alpha_n U_{n+1}^T \Sigma_n^{-1} & \alpha_n \end{bmatrix} \begin{bmatrix} Z_n^T \\ z_{n+1} \end{bmatrix}$$

$$= Z_n^T \Sigma_n^{-1} Z_n + \alpha_n ((U_{n+1}^T \Sigma_n^{-1} Z_n)((U_{n+1}^T \Sigma_n^{-1} Z_n) - 2z_{n+1}) + z_{n+1}^2)$$

$$= Q_n(x) + \alpha_n (t_n (t_n - 2z_{n+1}) + z_{n+1}^2)$$

$$= Q_n(x) + \alpha_n (t_n - z_{n+1})^2.$$

where

$$t_n = U_{n+1}^T \Sigma_n^{-1} Z_n = A_n^T Z_n = \sum_{k=1}^{n} a_k z_k,$$

$$Z_{n+1} = X_{n+1} - M_{n+1} = \begin{bmatrix} Z_n^T \\ z_{n+1} \end{bmatrix}, and$$

$$A_n^T = U_{n+1}^T \Sigma_n^{-1} = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}.$$

The first term is known from the previous iteration. The vector $U_{n+1}^t \Sigma_n^{-1}$ needs to be calculated only once for the cluster at the begining, then saved in the memory and regarded as a constant at the clustering stage for each sample. At the clustering stage, the calculation of above equation requires only $n+3$ multiplications. Thus, the total number of multiplications for one cluster through all *d* features is $\frac{(d^2+5d)}{2} - 3$.

3.2 Lower Triangular Canonical Form (LTCF) with Matrix Inversion

The covariance matrix can be represented in terms of the Cholesky decomposition and so its inverse. That is, the matrix Σ_k^{-1} can be represented in terms of a lower triangular matrix L_k as $\Sigma_n^{-1} = L_n^T L_n$. According to the Cholesky decomposition,

$$\Sigma_n = [\sigma_{ij}] = L_n L_n^T$$

with

$$L_n = \begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} = [l_{ij}]$$

and l_{ij} is given by:

$$l_{ii}^T = \left(\sigma_{ii} - \sum_{k=1}^{i-1} (l_{ki}^T)^2 \right)^{1/2},$$

$$i = 1, \cdots, n$$

$$l_{ij}^T = \left(\sigma_{ij} - \sum_{k=1}^{i-1} l_{ki}^T l_{kj}^T \right) / l_{ii}^T,$$

$$j = i+1, \cdots, n$$

The inverse of L_n, denoted by $L_n^{-1} = [l_{ij}^{-1}]$ is given by:

$$l_{ij}^{-1} = -\sum_{k=1}^{i-1} l_{ki}^T l_{kj}^{-1} / l_{ii}^T,$$

$$j = 1, \cdots, i-1$$

$$l_{ii}^{-1} = \frac{1}{l_{ii}^T}$$

The quadratic term for each cluster can be rewritten as:

$$Q_n(x) = Z_n^T L_n^{-T} L_n^{-1} Z_n = Y^T Y$$

where

$$Y = L_n^{-1} Z_n = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T,$$

with

$$y_i = \sum_{j=1}^{i} \left(l_{ij}^{-1} \right) z_j,$$

$i = 1,2,\cdots,n.$

Above equation can be rewritten as:

$$Q_n(x) = \sum_{i=1}^{n} y_i^2.$$

A recursive formula for calculating Q(x) can given by:

$$Q_n(x) = \sum_{i=1}^{n} y_i^2 = \sum_{i=1}^{(n-1)} y_i^2 + y_n^2 = Q_{n-1}(x) + y_n^2.$$

The calculation of above equation requires only n + 1 multiplications. Thus, the total number of multiplications for a cluster through all *d* features is $\frac{(d^2+3d)}{2}$.

3.3 Lower Triangular Canonical Form (LTCF) with Forward Substitution

In the above quadratic term calculation $Q_n(x) = Z_n^T L^{-T} L^{-1} Z_n = Y^T Y$, after the Cholesky decomposition of $\Sigma = LL^T$ is performed, Y can be calculated by forward substitution using $LY = Z$. A close-form solution of the equation $LY = Z$ by forward substitution is given by:

$$y_n = \frac{z_n - \sum_{k=1}^{n-1} l_{nk} y_k}{l_{nn}}$$

The number of multiplications of forming $Q_n(x)$ from $Y^T Y$ is the same as that of the previous approach. Note that in this approach Σ^{-1} need not be calculated. The calculation of above equation requires only n + 1 multiplications. Thus, the total number of multiplications for a cluster through all *d* features is $\frac{(d^2+3d)}{2}$.

4. Experiments and Discussion

Implementing the above three methods and the standard EM on synthetic datasets, with 1 million rows, varying the number of dimensions(d) from 50 to 100 in steps of 10 and the number of clusters(k) from 5 to 10 gives the following observations. The experiments are taken up on Intel Dula-Core system with 2.6GHz processor speed, 1 GB RAM, Fedora 10 OS and gcc compiler.

4.1 Synthetic dataset varying number of dimensions

Table I gives the time taken by proposed methods and standard EM for quadratic term computation on synthetic data varying the number of dimensions(d) from 50 to 100 in steps of 10 and the number of samples(m) fixed to 1 Million rows and number of clusters(k) 5. Table II gives comparitive statement of time taken for quadratic term computation by proposed approaches against that of standard EM. Fig 1 is a graphical representation of Table II.

Table I. Comparision of execution times (in sec) of proposed methods against standard EM varying the number of dimensions

	EM	Recursive	LTCF Inversion	LTCF Forward Substitution
d=50	120.07	58.25	56.04	62.22
d=60	172.06	83.15	79.52	87.46
d=70	232.9	112.56	106.62	118.71
d=80	298.41	146.25	137.62	152.09
d=90	385.87	184.62	173.24	192.23
d=100	484.78	225.07	213.33	235.94

Table II. Comparision of % execution times of proposed methods against standard EM varying the number of dimensions

	Recursive	LTCF Inversion	LTCF Forward Substitution
d=50	48.5133672	46.67277422	51.8197718
d=60	48.32616529	46.21643613	50.83110543
d=70	48.32975526	45.77930442	50.97037355
d=80	49.00975168	46.11775745	50.96679066
d=90	47.84512919	44.89594941	49.81729598
d=100	46.42724535	44.00552828	48.66949957
Average	48.07523566	45.61462498	50.51247283
Rank	2	1	3

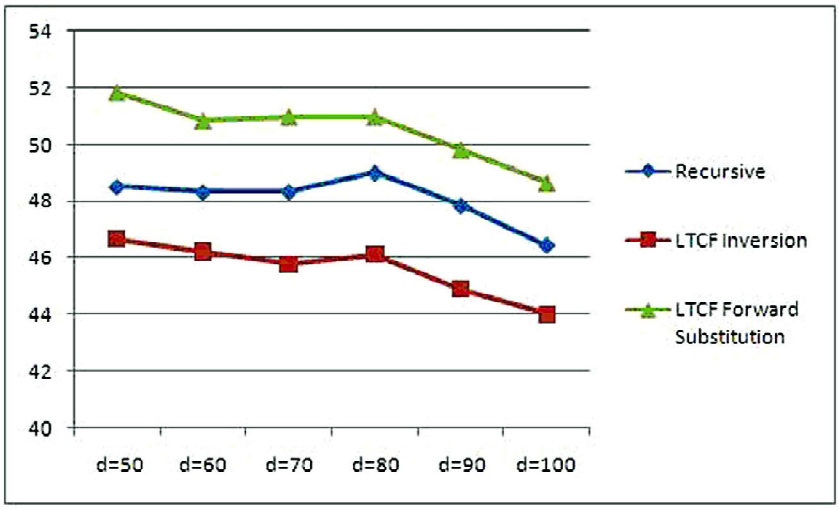

Fig 1. Comparision of % execution times of proposed methods against standard EM varying the number of dimensions

The conclusion is that approach LTCF Inverse is outperforming rest of the methods. It takes only 44.61% of time compared to standard EM.

4.2 Synthetic dataset varying number of clusters

Table III gives the time taken by proposed methods and standard EM for quadratic term computation on synthetic data of 1 Million rows varying the number of clusters(k) from 5 to 10 keeping the number of dimensions(d) fixed to 50. Table III gives comparitive statement of time taken for quadratic term computation by proposed approaches against that of standard EM. Fig 2 is a graphical representation of Table IV.

Table III. Comparision of execution times (in sec) of proposed methods against standard EM varying the number of clusters

	EM	Recursive	LTCF Inversion	LTCF Forward Substitution
k=5	120.07	58.25	56.04	62.22
k=6	144.01	70.11	67.47	74.72
k=7	168.27	82.84	79.47	86.86
k=8	191.96	94.32	90.50	101.81
k=9	216.28	106.7	102.19	112.16
k=10	240.17	118.18	114.09	126.41

Table IV. Comparision of % execution times of proposed methods against standard EM varying the number of clusters

	Recursive	LTCF Inversion	LTCF Forward Substitution
k=5	48.5133672	46.67277422	51.8197718
k=6	48.68411916	46.85091313	51.88528574
k=7	49.23040352	47.22766982	51.61942117
k=8	49.13523651	47.14523859	53.03709106
k=9	49.33419641	47.24893656	51.85870168
k=10	49.20681184	47.50385144	52.63355123
Average	49.01735577	47.10823063	52.14230378
Rank	2	1	3

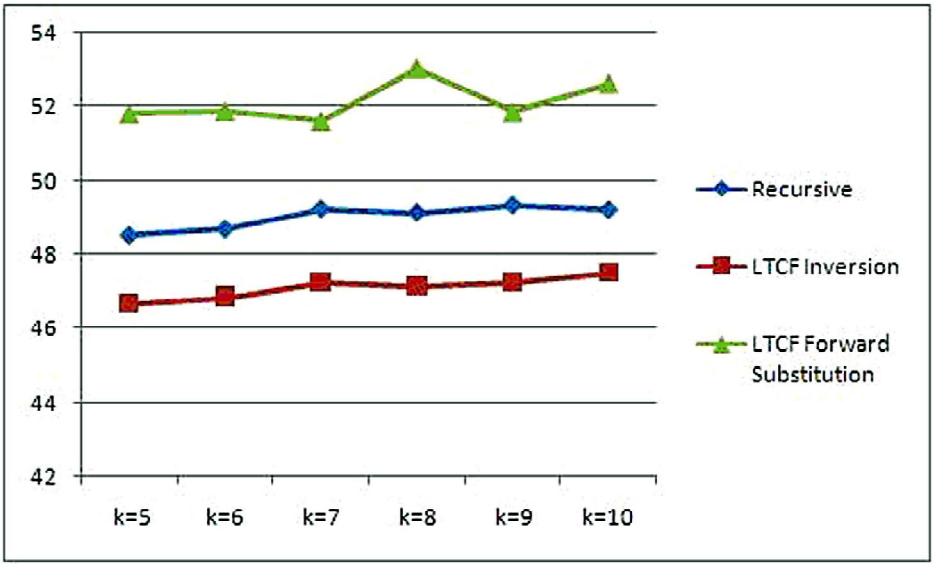

Fig 2. Comparision of % execution times of proposed methods against standard EM varying the number of clusters

The conclusion is that approach LTCF with Matrix Inversion is outperforming rest of the methods, taking only 47.11% of time compared to standard EM.

5. Conclusion :

The observation is that all the proposed approaches are working as expected. Out of all the three approaches, LTCF with Matrix Inversion is the fastet approach. We are trying to further improve the speed of Expectation-step by applying additional matrix algebra techniques and to study them in various Operating System and Computing environments.

References

[1] Cédric Archambeau and John Aldo Lee and Michel Verleysen. On Convergence Problems of the EM Algorithm for Finite Gaussian Mixtures. *ESANN*, pages 99—106, 2003.

[2] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[3] Jeff A. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1997.

[4] Sean Borman. The expectation maximization algorithm: A short tutorial. Unpublished paper available at http://www.seanborman.com/publications. Technical report, 2004.

[5] Nizar Bouguila. Clustering of Count Data Using Generalized Dirichlet Multinomial Distributions. *IEEE Transactions on Knowledge and Data Engineering*, 20:462-474, 2008.

[6] Paul S. Bradley and Usama M. Fayyad and Cory A. Reina and P. S. Bradley and Usama Fayyad and Cory Reina. Scaling EM (Expectation-Maximization) Clustering to Large Databases. 1999.

[7] A. P. Dempster and N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1—38, 1977.

[8] Wenbin Guo and Shuguang Cui. A q-Parameterized Deterministic Annealing EM Algorithm Based on Nonextensive Statistical Mechanics. *IEEE Transactions on Signal Processing*, 56(7-2):3069—3080, 2008.

[9] Radu Horaud and Florence Forbes and Manuel Yguel and Guillaume Dewaele and Jian Zhang. Rigid and Articulated Point Registration with Expectation Conditional Maximization. 2009.

[10] A. K. Jain and M. N. Murty and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3), 1999.

[11] H. D. Jin and M. L. Wong and K. S. Leung. Scalable Model-Based Clustering for Large Databases Based on Data Summarization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(??):1710—1719, 2005.

[12] F. -X. Jollois and M. Nadif. Speed-up for the expectation-maximization algorithm for clustering categorical data. 2007.

[13] Michael I. Jordan. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181—214, 1994.

[14] Chanho Jung. Unsupervised Segmentation of Overlapped Nuclei Using Bayesian Classification. IEEE 2010

[15] Wojtek Kowalczyk and Nikos Vlassis. Newscast EM. *In NIPS 17*, pages 713—720, 2005. MIT Press.

[16] Martin H. C. Law and Mário A. T. Figueiredo and Anil K. Jain. Simultaneous Feature Selection and Clustering Using Mixture Models. *IEEE Trans. Pattern Anal. Mach. Intell*, 26(??):1154—1166, 2004.

[17] Adam Lopez. Statistical machine translation. *ACM Comput. Surv*, 40(3), 2008.

[18] M. W. Mak, S. Y. Kung, S. . Lin. Expectation-Maximization Theory. 2005.

[19] Andrew Mccallum and Kamal Nigam. Employing EM in Pool-Based Active Learning for Text Classification. 1998.

[20] Marina Meilã and David Heckerman. An experimental comparison of several clustering and intialization methods. Technical report, 1998.

[21] Norwati Mustapha. Expectation Maximization Clustering Algorithm for User Modeling in Web Usage Mining Systems EUJSR 2009.

[22] Radford Neal and Geoffrey E. Hinton. A View Of The EM Algorithm That Justifies Incremental, Sparse, And Other Variants. *Learning in Graphical Models*, pages 355—368, 1998. Kluwer Academic Publishers.

[23] Kamal Nigam and Andrew Kachites Mccallum and Sebastian Thrun and Tom Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103, 2000.

[24] C. Nikou and N. P. Galatsanos and A. C. Likas. A Class-Adaptive Spatially Variant Mixture Model for Image Segmentation. *IEEE Trans. Image Processing*, 16(4):1121—1130, 2007.

[25] Zachary A. Pardos and Neil T. Heffernan. Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm. In Ryan Shaun Joazeiro de Baker and Agathe Merceron and Philip I. Pavlik Jr, editors, *Educational Data Mining 2010, The 3rd International Conference on Educational Data Mining, Pittsburgh, PA, USA, June 11-13, 2010. Proceedings*, pages 161—170, 2010. www.educationaldatamining.org.

[26] C. K. Reddy and H. D. Chiang and B. Rajaratnam. TRUST-TECH-Based Expectation Maximization for Learning Finite Mixture Models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(7):1146—1157, 2008.

[27] Leonardo Rigutini and Marco Maggini. An EM based training algorithm for Cross-Language Text Categorization. *in In Proceedings of the Web Intelligence Conference (WI*, pages 529—535, 2005.

[28] Jonathan Le Roux and Hirokazu Kameoka and Nobutaka Ono and Alain de Cheveigné and Shigeki Sagayama. Single and Multiple F_0 Contour Estimation Through Parametric Spectrogram Modeling of Speech in Noisy Environments. *IEEE Transactions on Audio, Speech & Language Processing*, 15(4):1135—1145, 2007.

[29] Sam Roweis. EM Algorithms for PCA and SPCA. *in Advances in Neural Information Processing Systems*, pages 626—632, 1998. MIT Press.

[30] S.P. Smith, C.Y. Lin. Efficient implementation of the new restricted maximum likelihood algorithm. 1989.

[31] Ruslan Salakhutdinov and Sam Roweis and Zoubin Ghahramani. Optimization with EM and Expectation-Conjugate-Gradient. pages 672—679, 2003.

[32] Christian D. Sigg and Joachim M. Buhmann. Expectation-maximization for sparse and non-negative PCA. In William W. Cohen and Andrew McCallum and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-*

Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008 in ACM International Conference Proceeding Series, pages 960—967, 2008. ACM.

[33] Bo Thiesson and Christopher Meek and David Heckerman. Accelerating EM for large databases. Technical report, Machine Learning, 2001.

[34] Michael E. Tipping and Chris M. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society, Series B*, 61:611—622, 1999.

[35] Rudolph Triebel and Wolfram Burgard. Using hierarchical EM to extract planes from 3d range scans. *In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA*, 2005.

[36] N B Venkateswarlu and S. Balaji and R D Boyle. Some Further Results of Three Stage ML Classification Applied to Remotely Sensed Images. Technical report, 1993.

[37] Jakob J. Verbeek and Jan R. J. Nunnink and Nikos Vlassis. Accelerated EM-based clustering of large data sets. *Data Mining and Knowledge Discovery*, 13:2006, 2006.

[38] Jason Wolfe and Aria Haghighi and Dan Klein. Fully distributed EM for very large datasets. In William W. Cohen and Andrew McCallum and Sam T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008* in ACM International Conference Proceeding Series, pages 1184—1191, 2008. ACM.

[39] Lei Xu and Michael I. Jordan. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8:129—151, 1995.

[40] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645—678, 2005.