Concept Formation Using Incremental Gaussian Mixture Models

Paulo Martins Engel and Milton Roberto Heinen

UFRGS – Informatics Institute
Porto Alegre, CEP 91501-970, RS, Brazil
engel@inf.ufrgs.br, mrheinen@inf.ufrgs.br

Abstract. This paper presents a new algorithm for incremental concept formation based on a Bayesian framework. The algorithm, called IGMM (for Incremental Gaussian Mixture Model), uses a probabilistic approach for modeling the environment, and so, it can rely on solid arguments to handle this issue. IGMM creates and continually adjusts a probabilistic model consistent to all sequentially presented data without storing or revisiting previous training data. IGMM is particularly useful for incremental clustering of data streams, as encountered in the domain of moving object trajectories and mobile robotics. It creates an incremental knowledge model of the domain consisting of primitive concepts involving all observed variables. Experiments with simulated data streams of sonar readings of a mobile robot shows that IGMM can efficiently segment trajectories detecting higher order concepts like "wall at right" and "curve at left".

Keywords: Concept Formation, Incremental Learning, Unsupervised Learning, Bayesian Methods, EM Algorithm, Finite Mixtures, Clustering.

1 Introduction

In this paper, we focus in the so called unsupervised incremental learning [1, 2], which considers building a model, seen as a set of concepts of the environment describing a data flow, where each data point is just instantaneously available to the learning system [3, 4]. In this case, the learning system needs to take into account these instantaneous data to update its model of the environment. An important issue in unsupervised incremental learning is the stability-plasticity dilemma, i.e., whether a new presented data point must be assimilated in the current model or cause a structural change in the model to accommodate the new information that it bears, i.e., a new concept. We show that our algorithm, the so called IGMM (standing for Incremental Gaussian Mixture Model), uses a probabilistic approach for modeling the environment, and so, it can rely on solid arguments to handle this issue [5, 6].

We are interested in problems like the ones encountered in autonomous robotics. To be more specific, we consider the so called perceptual learning that allows an embodied agent to understand the world [7]. Here an important task is the detection of concepts such as "corners", "walls" and "corridors" from the sequence of noisy sensor readings of a mobile robot. The detection of these regularities in data flow allows the robot to localize its position and to detect changes in the environment [8]. In the past, different approaches were presented to this end, but they have scarce means to handle the

I. Bloch and R.M. Cesar, Jr. (Eds.): CIARP 2010, LNCS 6419, pp. 128–135, 2010.

stability-plasticity dilemma and to appropriately model the data. As a typical example of these approaches, Nolfi and Tani [9] proposed a hierarchical architecture to extract regularities from time series, in which higher layers are trained to predict the internal state of lower layers when such states change significantly. In this approach, the segmentation was cast as a traditional error minimization problem [10], which favors the most frequent inputs, filtering out less frequent input patterns as being "noise". The result is that this system recognizes slightly differing walls, that represent frequent input patterns, as distinguish concepts, but is unable to detect corridors or corners that are occasionally (infrequently) encountered.

Focusing in change detection, Linåker and Niklasson [11, 12] proposed an adaptive resource allocating vector quantization (ARAVQ) network, which stores moving averages of segments of the data sequence as vectors allocated to output nodes of the network. New model vectors are incorporated to the model if a mismatch between the moving average of the input signal and the existing model vectors is greater than a specified threshold and a minimum stability criterion for the input signal is fulfilled. However, like other distance based clustering algorithm, the induced model is equivalent to a set of equiprobable spherical distributions sharing the same variance, what barely fits to a data flow with temporal correlation, better described by elongated elliptical distributions [5, 6].

Our approach can be seen as an incremental solution for the problem of probability density estimation, a very important research field in statistical pattern recognition [13, 14]. As the EM algorithm [15, 16], IGMM follows the mixture distribution modeling. However, its model can be effectively *expanded* with new components (i.e. concepts) as new relevant information is identified in the data flow. Moreover, IGMM adjusts the parameters of each distribution after the presentation of every single data point according to recursive equations that are approximate incremental counterparts of the batch-mode update equations used by the EM algorithm. Although in the past several attempts have been made to create an algorithm to learn Gaussian mixture models incrementally [17, 18, 19, 20], most of these attempts require several data points to the correct estimation of the covariance matrices and/or does not handles the stability-plasticity dilemma. The IGMM algorithm, on the other hand, converges after the presentation of few training samples and does not require a predefined number of distributions.

The promising results obtained with IGMM applied to sonar signal flows from a robot simulator, described later on in this text, point out that it fits the requirements of the so called Embodied Statistical Learning, a desired but still scarce set of statistical methods compatible to the design principles of Embodied AI [7]. The rest of this paper is organized as follows. Section 2 presents in details the proposed algorithm. Section 3 describes an experiment performed to evaluate the proposed model. Finally, Section 4 provides some final remarks and perspectives.

2 The Incremental Gaussian Mixture Model

This section describes the proposed model, called IGMM [5], which was designed to learn Gaussian mixture models from data flows in an incremental and unsupervised way. IGMM assumes that the probability density of the input data $p(\mathbf{x})$ can be modeled by a linear combination of component densities $p(\mathbf{x}|j)$ corresponding to independent probabilistic processes, in the form

$$p(\mathbf{x}) = \sum_{j=1}^{M} p(\mathbf{x}|j)p(j) \tag{1}$$

This representation is called a *mixture model* and the coefficients $p(j)$ are called the mixing parameters, related to the *prior* probability of $\mathbf{x}$ having been generated from component j of the mixture. The priors are adjusted to satisfy the constraints

$$\sum_{j=1}^{M} p(j) = 1 \tag{2}$$

$$0 \leq p(j) \leq 1 \tag{3}$$

Similarly, the component density functions $p(\mathbf{x}|j)$ are normalized so that

$$\int p(\mathbf{x}|j)d\mathbf{x} = 1 \tag{4}$$

The probability of observing vector $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_D)$ belonging to the jth mixture component, is computed by a multivariate normal Gaussian, with mean $\boldsymbol{\mu}_j$ and covariance matrix $\mathbf{C}_j$:

$$p(\mathbf{x}|j) = \frac{1}{(2\pi)^{D/2}\sqrt{|\mathbf{C}_j|}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right\} \tag{5}$$

IGMM adopts an incremental mixture distribution model, having special means to control the number of mixture components that effectively represent the so far presented data. We are interested in modeling environments whose overall dynamics can be described by a set of persistent concepts which will be incrementally learned and represented by a set of mixture components. So, we can now rely on a novelty criterion to overcome the problem of the model complexity selection, related to the decision whether a new component should be added to the current model. The mixture model starts with a single component with unity prior, centered at the first input data, with a baseline covariance matrix specified by default, i. e., $\boldsymbol{\mu}_1 = \mathbf{x}^1$, meaning the value of $\mathbf{x}$ for $t = 1$, and $(\mathbf{C}_1)^1 = \sigma_{ini}^2\mathbf{I}$, where σ_{ini} is user-specified configuration parameter.

New components are added by demand. IGMM uses a *minimum likelihood* criterion to recognize a vector $\mathbf{x}$ as belonging to a mixture component. For each incoming data point the algorithm verifies whether it minimally fits any mixture component. A data point $\mathbf{x}$ is not recognized as belonging to a mixture component j if its probability $p(\mathbf{x}|j)$ is lower than a previously specified *minimum likelihood-* (or *novelty-*) *threshold.* In this case, $p(\mathbf{x}|j)$ is interpreted as a *likelihood function* of the jth mixture component. If $\mathbf{x}$ is rejected by all density components, meaning that it bears new information, a new component is added to the model, appropriately adjusting its parameters. The novelty-threshold value affects the sensibility of the learning process to new concepts, with higher threshold values generating more concepts. It is more intuitive for the user to specify a minimum value for the acceptable likelihood, τ_{nov}, as a *fraction* of the maximum value of the likelihood function, making the novelty criterion independent of the covariance matrix. Hence, a new mixture component is created when the instantaneous data point $\mathbf{x} = (x_1, \dots, x_i, \dots, x_D)$ matches the *novelty criterion* written as

$$p(\mathbf{x}|j) < \frac{\tau_{nov}}{(2\pi)^{D/2}\sqrt{|\mathbf{C}_j|}} \;\; \forall j \qquad (6)$$

An instantaneous data point that does not match the novelty criterion needs to be assimilated by the current mixture distribution, causing an update in the values of its parameters due to the information it bears. IGMM follows an incremental version for the usual iterative process to estimate the parameters of a mixture model based on two steps: an estimation step (E) and a maximization step (M). The update process begins computing the posterior probabilities of component membership for the data point, $p(j|\mathbf{x})$, the *estimation* step. These can be obtained through Bayes' theorem, using the current component-conditional densities $p(\mathbf{x}|j)$ and priors $p(j)$ as follows:

$$p(j|\mathbf{x}) = \frac{p(\mathbf{x}|j)p(j)}{\sum_{j=1}^{M} p(\mathbf{x}|j)p(j)} \;\; \forall j \qquad (7)$$

The posterior probabilities can then be used to compute new estimates for the values of the mean vector $\boldsymbol{\mu}_i^{new}$ and covariance matrix $\mathbf{C}_j^{new}$ of each component density $p(\mathbf{x}|j)$, and the priors $p^{new}(j)$ in the *maximization* step. Next, we derive the recursive equations used by IGMM to successively estimate these parameters.

The parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_M)^T$, corresponding to the means, $\boldsymbol{\mu}_j$, covariances matrices, $\mathbf{C}_j$, and priors $p(j)$ of a mixture model involving D-dimensional Gaussian distributions $p(\mathbf{x}|j)$, can be estimated from a data sequence of t vectors, $\mathbf{X} = \{\mathbf{x}^1, \dots, \mathbf{x}^n, \dots, \mathbf{x}^t\}$ assumed to be drawn independently from this mixture distribution. The estimates of the parameters are random vectors whose statistical proprieties are obtained from their joint density function. Starting from an initial "guess", each observation vector is used to update the estimates according to a successive estimation procedure.

IGMM follows the Robbins-Monro stochastic approximation method to derive the recursive equation used to successively estimate the priors [21]. For this, in the maximization step the parameters of the current model are updated based on the maximization of the likelihood of the data.

In this case, the *likelihood* of $\boldsymbol{\theta}$ for the given $\mathbf{X}$, $L(\boldsymbol{\theta})$, is the joint probability density of the whole data stream $\mathbf{X}$, given by

$$L(\theta) \equiv p(\mathbf{X}|\theta) = \prod_{n=1}^{t} p(\mathbf{x}^n|\theta) = \prod_{n=1}^{t} \left[\sum_{j=1}^{M} p(\mathbf{x}^n|j)p(j) \right] \qquad (8)$$

The technique of maximum likelihood sets the value of $\boldsymbol{\theta}$ by maximizing $L(\boldsymbol{\theta})$.

Although the maximum likelihood technique for estimating the priors is straightforward, it becomes quite complex when applied to estimate the mean vector and the covariance matrix directly from (5). Instead, we follow the natural conjugate technique to estimate these parameters [22]. When $\boldsymbol{\mu}$ and $\mathbf{C}$ are estimated by the sample mean vector and sample covariance matrix, and $\mathbf{X}$ is a normally distributed random vector, the joint density function $p(\boldsymbol{\mu}, \mathbf{C}|\mathbf{x}^1, \ldots, \mathbf{x}^i, \ldots, \mathbf{x}^n)$ is known to be the reproducible Gauss-Wishart distribution, the natural conjugate density for the model of (5) [22]. In this case, when we estimate both the expected vector and the covariance matrix of a single distribution, starting with a priori distribution with an expected vector $\boldsymbol{\mu}^0$ and covariance matrix $\mathbf{C}^0$, these parameters are transformed through n observations in the following manner [22, 13]:

$$\omega^1 = \omega^0 + n \qquad v^1 = v^0 + n$$

$$\boldsymbol{\mu}^1 = \frac{\omega^0 \boldsymbol{\mu}^0 + n \langle \mathbf{X} \rangle}{\omega^0 + n} \tag{9}$$

$$\mathbf{C}^1 = \frac{\left(v^0 \mathbf{C}^0 + \omega^0 \boldsymbol{\mu}^0 (\boldsymbol{\mu}^0)^T \right) + n \langle \mathbf{X} \rangle \langle \mathbf{X} \rangle^T - \omega^1 \boldsymbol{\mu}^1 (\boldsymbol{\mu}^1)^T}{v^0 + n} \tag{10}$$

where ω^0 and v^0 reflect the confidence about the initial estimates of $\boldsymbol{\mu}^0$ and $\mathbf{C}^0$ respectively, corresponding to the number of samples used to compute these initial estimates.

On the other hand, when the probability density of the input data is a Gaussian Mixture Model with M components, an observation $\mathbf{x}^t$ is probabilistic assigned to a distribution j by the corresponding posterior probability $p(j|\mathbf{x}^t)$. In this case, the equivalent number of samples used to compute the parameter estimates of the jth distribution component corresponds to the sum of posterior probabilities that the data presented so far were generated from component j, the so called *0th-order moment of $p(j|\mathbf{x})$ over the data*, or simply the *0th-order data moment for j*. IGMM stores this summation as the variable sp_j which is periodically restarted to avoid an eventual saturation.

The recursive equations used by IGMM to update the model distributions are:

$$sp_j = sp_j + p(j|\mathbf{x}) \tag{11}$$

$$\boldsymbol{\mu}_j = \boldsymbol{\mu}_j + \frac{p(j|\mathbf{x})}{sp_j} (\mathbf{x} - \boldsymbol{\mu}_j) \tag{12}$$

$$\mathbf{C}_j = \mathbf{C}_j - (\boldsymbol{\mu}_j - \boldsymbol{\mu}_j^{old})(\boldsymbol{\mu}_j - \boldsymbol{\mu}_j^{old})^T + \frac{p(j|\mathbf{x})}{sp_j} \left[(\mathbf{x} - \boldsymbol{\mu}_j)(\mathbf{x} - \boldsymbol{\mu}_j)^T - \mathbf{C}_j \right] \tag{13}$$

$$p(j) = sp_j / \textstyle\sum_{q=1}^{M} sp_q \tag{14}$$

where $p(j|\mathbf{x})$ $\boldsymbol{\mu}_j^{old}$ refers to the value of $\boldsymbol{\mu}_j$ at time $t-1$ (i.e., before updating). One important property of these update equations is the fact that they continuously compute a instantaneous approximation of the parameters that represent the mixture distribution.

The IGMM algorithm has just two configuration parameters, σ_{ini} and τ_{nov}. The σ_{ini} parameter is not critical – its only requirement for σ_{ini} is be large enough to avoid singularities. In our experiments we have simply used $\sigma_{ini} = (\mathbf{x}_{max} - \mathbf{x}_{min})/10$. The τ_{nov} parameter, on the other hand, is more critical and must be defined carefully. It indicates how distant $\mathbf{x}$ must be from $\boldsymbol{\mu}_j$ to be consider a non-member of j. For instance, $\tau_{nov} = 0.01$ indicates that $p(\mathbf{x}|j)$ must be lower than one percent of the Gaussian height (probability in the center of the Gaussian) for $\mathbf{x}$ be considered a non-member of j. If $\tau_{nov} < 0.01$, few pattern units will be created and the regression will be coarse. If $\tau_{nov} > 0.01$, more pattern units will be created and consequently the regression will be more precise. In the limit, if $\tau_{nov} = 1$ one unit per training pattern will be created.

3 Experimental Results

This section describes the experiments devised to evaluate IGMM using data obtained from simulated mobile robot sonars. In these experiments, the data consist of a sequence of 4 continuous values (s_1, s_2, s_3, s_4) corresponding to the readings of a sonar array located at the left/right side (s_1, s_4) and at $-10°/+10°$ from the front (s_2, s_3) of a robot, generated using the Pioneer 3-DX simulator software ARCOS (Advanced Robot Control & Operations Software). The first experiment was accomplished in an environment composed of six corridors (four external and two internal), and the robot performed a complete cycle in the external corridors. Fig. 1 shows the segmentation of the trajectory obtained by IGMM when the robot follows the corridors of this environment. IGMM created four clusters corresponding to the concepts of “corridor” (red), “wall at right” (blue), “corridor / obstacle front” (black) and “curve at left” (cyan). The colored filled dots in this figure correspond to the location where each cluster was created. A square represents a robot position and has the same color of the cluster with the largest posterior probability for the corresponding data point.

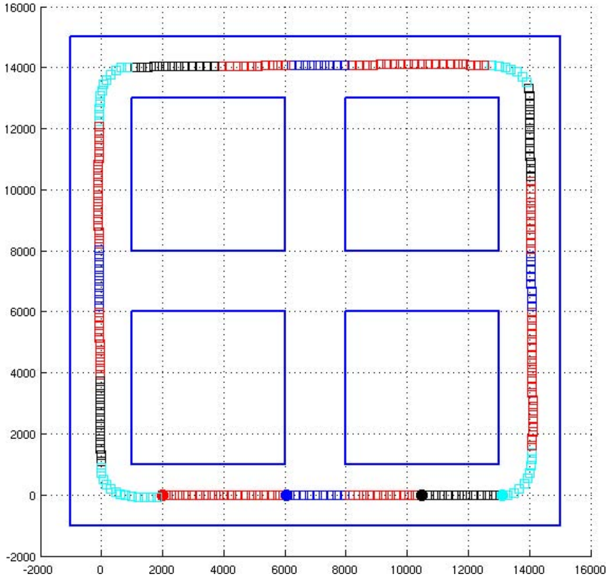

Fig. 1. Concepts created in the environment composed of six corridors

The next experiment was performed in an environment with two different sized rooms connected by a short corridor. This more complex environment is inspired in those used in [9] and [11,12]. Fig. 2 shows the segmentation of the trajectory of a robot following the walls in this environment. IGMM created seven clusters corresponding to the concepts ”wall at right” (1: red), ”corridor” (2: blue), ”wall at right / obstacle front” (3: black), ”curve at left” (4: cyan), ”bifurcation / obstacle front” (5: magenta), ”bifurcation / curve at right” (6: green) and ”wall at left / curve at right” (7: yellow).

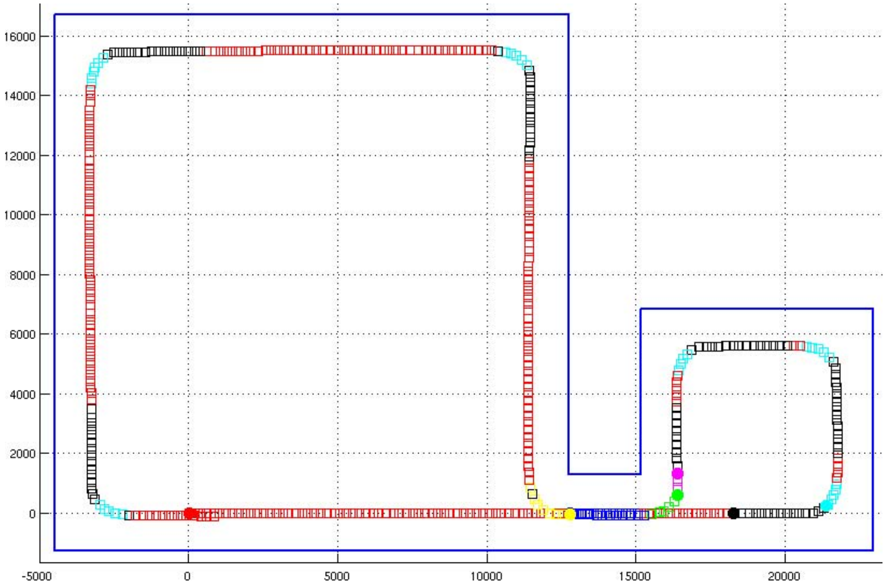

Fig. 2. Concepts created in the environment with two different sized rooms

Comparing the experiments, it can be noticed that some similar concepts, like "curve at left" (cyan) and "obstacle front" (black), were discovered in both experiments, although the environments are different. This points out that concepts extracted from a data flow corresponding to a specific sensed environment are not restricted to it, but they form an alphabet that can be reused in other contexts. This is a useful aspect, that can improve the learning process in complex environments.

4 Conclusion

In this paper we presented IGMM, an algorithm for modeling data flows that fulfills the requirements of the so called Embodied Statistical Learning [7]. It is rooted in the well established field of statistical learning, using an incremental Gaussian Mixture Model to represent the probability density of the input data flow, and adding new density components to the model whenever a new regularity, or concept, is identified in the incoming data. The experimental results confirmed that IGMM was able to extract useful concepts of the data flow from just a single iteration over the training data. This experiment have also shown the representational power of the generated statistical model, since from the values of the computed parameters and the corresponding plots one could readily interpret and label each extracted concept.

Acknowledgment

This work is supported by CNPq, an entity of the Brazilian government for scientific and technological development.

References

1. Arandjelovic, O., Cipolla, R.: Incremental learning of temporally-coherent Gaussian mixture models. In: Proc. 16th British Machine Vision Conf. (BMVC), Oxford, UK, pp. 759–768 (September 2005)
2. Kristan, M., Skocaj, D., Leonardis, A.: Incremental learning with Gaussian mixture models. In: Proc. Computer Vision Winter Workshop, Moravske Toplice, Slovenia, pp. 25–32 (2008)
3. Fisher, D.H.: Knowledge acquisition via incremental conceptual learning. Machine Learning 2, 139–172 (1987)
4. Gennari, J.H., Langley, P., Fisher, D.: Models of incremental concept formation. Artificial Intelligence 40, 11–61 (1989)
5. Engel, P.M., Heinen, M.R.: Incremental learning of multivariate Gaussian mixture models. In: Proc. 20th Brazilian Symposium on AI (SBIA), São Bernardo do Campo, SP, Brazil. Springer, Heidelberg (October 2010)
6. Heinen, M.R., Engel, P.M.: An incremental probabilistic neural network for regression and reinforcement learning tasks. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010, Part II. LNCS, vol. 6353, pp. 170–179. Springer, Heidelberg (2010)
7. Burfoot, D., Lungarella, M., Kuniyoshi, Y.: Toward a theory of embodied statistical learning. In: Asada, M., Hallam, J.C.T., Meyer, J.-A., Tani, J. (eds.) SAB 2008. LNCS (LNAI), vol. 5040, pp. 270–279. Springer, Heidelberg (2008)
8. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. In: Intelligent Robotics and Autonomous Agents. MIT Press, Cambridge (2006)
9. Nolfi, S., Tani, J.: Extracting regularities in space and time through a cascade of prediction networks: The case of a mobile robot navigating in a structured environment. Connection Science 11(2), 125–148 (1999)
10. Haykin, S.: Neural Networks and Learning Machines, 3rd edn. Prentice-Hall, Upper Saddle River (2008)
11. Linåker, F., Niklasson, L.: Time series segmentation using an adaptive resource allocating vector quantization network based on change detection. In: Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks (IJCNN 2000), Los Alamitos, CA, USA, pp. 323–328 (2000)
12. Linåker, F., Niklasson, L.: Sensory flow segmentation using a resource allocating vector quantizer. In: Amin, A., Pudil, P., Ferri, F., Iñesta, J.M. (eds.) SPR 2000 and SSPR 2000. LNCS, vol. 1876, pp. 853–862. Springer, Heidelberg (2000)
13. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, London (1990)
14. Bishop, C.: Neural Networks for Pattern Recognition. Oxford Univ. Press, New York (1995)
15. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39(1), 1–38 (1977)
16. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Boston (2006)
17. Titterington, D.M.: Recursive parameter estimation using incomplete data. Journal of the Royal Statistical Society 46(2), 257–267 (1984)
18. Wang, S., Zhao, Y.: Almost sure convergence of titterington's recursive estimator for mixture models. Statistics & Probability Letters (76), 2001–2006 (2006)
19. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Learning in Graphical Models, pp. 355–368. Kluwer Academic Publishers, Dordrecht (1998)
20. Cappé, O., Moulines, E.: Online EM algorithm for latent data models. Journal of the Royal Statistical Society (September 2008)
21. Robbins, H., Monro, S.: A stochastic approximation method. Annals of Mathematical Statistics 22, 400–407 (1951)
22. Keehn, D.G.: A note on learning for Gaussian proprieties. IEEE Trans. Information Theory 11, 126–132 (1965)