A Fixed-Point EM Algorithm for Straight Line Detection

Chonglun Fang and Jinwen Ma *

Department of Information Science, School of Mathematical Sciences and LMAM, Peking University, Beijing, 100871, China

Abstract. Straight line detection is a basic technique in image processing and pattern recognition. It has been investigated from different aspects, but is still very challenging in practical applications. In this paper, based on the finite mixture model and under the EM framework, we maximize the Q-function by differentiation and construct a fixed-point EM algorithm for straight line detection. It is demonstrated by the experiments that this proposed algorithm can effectively detect the straight lines from a digital image or dataset.

Keywords: Straight line detection, Expectation Maximization (EM), Fixed-Point iteration.

1 Introduction

Straight line detection is a basic technique in image processing and pattern recognition. In fact, it is a process of locating the straight lines from a digital image or 2-dimensional dataset and there are a variety of learning algorithms for straight line detection. In the literature, the Hough transform (HT) [1] and its extensions [2] are important tools for straight line detection. Generally, the Hough transform suffers from heavy computational cost. In order to alleviate this weakness, Random Hough Transform (RHT) [3] and the constrained Hough Transform [4] were further established. From the other aspects, there have also established many learning algorithms for straight line or curve detection (e.g., [5]-[8]).

The local principal component analysis (PCA) algorithm [9]-[12], as an extension of PCA [13], is often used for straight line detection. It implements the least mean square error reconstruction (LMSER) principle [14] and detects the straight lines via minimizing the following cost function [11]:

$$E = \sum_{k=1}^{K} E_k = \sum_{k=1}^{K} \sum_{x_t \in \mathcal{L}_k} d^2(x_t, \mathcal{L}_k) \tag{1}$$

where x_t is the t-th sample point belonging to the line $\mathcal{L}_k$ and $d(x_t, \mathcal{L}_k)$ denotes the Euclidean distance from the data point x_t to the line $\mathcal{L}_k$. Actually, the line $\mathcal{L}_k$ can be considered as a special subset of points. On the other hand, the line $\mathcal{L}_k$

* Corresponding author, jwma@math.pku.edu.cn

D. Liu et al. (Eds.): ISNN 2011, Part II, LNCS 6676, pp. 136–143, 2011.

can be also regarded as a cluster. Actually, E reaches its minimum when the k-th straight line is the first principal component vector of the cluster $\mathcal{L}_k$. Therefore, the solution of $\mathcal{L}_k$ is just the same as that of the local principal component analysis method.

Each x_t is assigned to a straight line by the classification membership function given by

$$I(x_t, k) = \begin{cases} 1 & \text{if } k = \arg\min d(x_t, \mathcal{L}_j),\ j = 1, \cdots, K \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In fact, x_t belongs to $\mathcal{L}_k$ if and only if $I(x_t, k) = 1$. It means that the distance from x_t to $\mathcal{L}_k$ is the smallest one. Thus, we can update $\mathcal{L}_k$ by the following rule: $\mathcal{L}_k$ is the first principal component of the subset $\mathcal{L}_k$. Recently, the RPCL algorithm [15] was combined with the local PCA algorithm for straight line detection [6].

In this paper, we utilize the finite mixture model for straight line detection. It is not so easy to solve the maximum solution of the log-likelihood function directly. So, we resort to the Expectation Maximization (EM) [16] and analyze the Q-function. By the differentiation of the Q-function, we construct a fixed-point EM algorithm for straight line detection. It is demonstrated by the experiments that the proposed fixed-point EM algorithm can detect the straight lines from a datatet effectively.

The rest of the paper is organized as follows. We begin to introduce the finite mixture model for straight line detection in Section 2. We then derive and present our fixed-point EM algorithm in Section 3. The experimental results are further demonstrated in Section 4. Finally, a brief conclusion is made in Section 5.

2 The Finite Mixture Model for Straight Line Detection

Here, it is assumed that the number of straight lines in our learning model is equal to the number of actual straight lines in the image or dataset. For simplicity, we only focus on the 2-dimensional situation, but the derivation and analysis can be easily generalized to the situations with higher dimensions. Let the dataset be denoted by $\mathcal{S} = \{x_t\}_{t=1}^N$ and the point $x_t = (x_{t1}, x_{t2})^T$. Then, a point $x = (x_1, x_2)^T$ on a straight line $\mathcal{L}_k$ satisfies:

$$\frac{x_1 - m_{k1}}{\ell_{k1}} = \frac{x_2 - m_{k2}}{\ell_{k2}}, \tag{3}$$

where $m = (m_{k1}, m_{k2})^T$ is a specific point on the line $\mathcal{L}_k$, and

$${\ell_{k1}}^2 + {\ell_{k2}}^2 = 1. \tag{4}$$

Thus, the distance from the sample point x_t to the line $\mathcal{L}_k$ can be computed by

$$\begin{aligned} d^2(x_t, \mathcal{L}_k) &= d^2(x_t, \ell_k, m_k) & (5)\\ &= |x_t - m_k|^2 - |(x_t - m_k, \ell_k)|^2 & (6)\\ &= (x_{t1} - m_{k1})^2 + (x_{t2} - m_{k2})^2 - [(x_{t1} - m_{k1})\ell_{k1} + (x_{t2} - m_{k2})\ell_{k2}]^2 \end{aligned}$$

where $(x_t - m_k, \ell_k)$ denotes the inner product of $x_t - m_k$ and ℓ_k.

In this situation, we can establish the following finite mixture model:

$$q(x|\Theta_K) = \sum_{j=1}^{K} \pi_j q(x|\ell_j, m_j, \sigma_j), \tag{7}$$

where

$$q(x|\ell_j, m_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\{-\frac{d^2(x_t, \ell_j, m_j)}{2\sigma_j^2}\}, \tag{8}$$

$$\|\ell_j\|^2 = {\ell_{j1}}^2 + {\ell_{j2}}^2 = 1, \quad j = 1, \ldots, K, \tag{9}$$

$$\sum_{j=1}^{K} \pi_j = 1. \tag{10}$$

In this mixture model, π_i represents the mixing proportion. $d^2(x_t, \ell_k, m_k)$ denotes the distance between the sample point x_t and the line with parameters ℓ_k and m_k. σ_i can be considered as the noise level of the dataset. That is, when σ_i is large, the noise level is high.

As it is a finite mixture model, we are difficult to find the maximum of its log-likelihood function directly. So, we resort to the EM algorithm [16]. Under the EM framework, we introduce a missing variable j and construct the Q-function:

$$Q(\Theta_K^h, \Theta_K^{h+1}) = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} p(j|x_t, \Theta_K^h) \ln q(x_t|j, \Theta_K^{h+1}) \tag{11}$$

$$= \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} \frac{\pi_j^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)}{\sum_{i=1}^{K} \pi_i^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)} \ln[\pi_j^{h+1} q(x_t|\ell_j^{h+1}, m_j^{h+1}, \sigma_j^{h+1})]$$

$$= \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} p_j(t) \ln[\pi_j^{h+1} q(x_t|\ell_j^{h+1}, m_j^{h+1}, \sigma_j^{h+1})] \tag{12}$$

where $p_j(t) = \frac{\pi_j^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)}{\sum_{i=1}^{K} \pi_i^h q(x_t|\ell_j^h, m_j^h, \sigma_j^h)}$. For simplicity, the Q-function is denoted by

$$Q = \frac{1}{N} \sum_{t=1}^{N} \sum_{j=1}^{K} p_j(t) \ln[\pi_j q(x_t|\ell_j, m_j, \sigma_j)]. \tag{13}$$

3 Proposed Fixed-Point EM Algorithm

In the EM algorithm, it is key to solve the maximum of the Q-function. We now analyze the Q-function and try to establish a fixed-point learning algorithm to solve the maximum of Q-function.

Since $\sum_{j=1}^{K} \pi_j = 1$ and ${\ell_{j1}}^2 + {\ell_{j2}}^2 = 1$ for any j, we introduce the Lagrange multiplier $\beta, \lambda_j (j = 1, \ldots, K)$ and the Lagrange function

$$L(\Theta_K, \beta, \lambda_1, \ldots, \lambda_K) = Q + \beta(1 - \sum_{j=1}^{K} \pi_j) + \sum_{j=1}^{K} \lambda_j (1 - \ell_{j1}^2 - \ell_{j2}^2). \tag{14}$$

By differentiation, we have the following derivatives:

$$\frac{\partial L}{\partial \pi_j} = \frac{1}{N}\sum_{j=1}^{K}\frac{1}{\pi_j}p_j(t) - \beta, \tag{15}$$

$$\frac{\partial L}{\partial \beta} = 1 - \sum_{j=1}^{K}\pi_j, \tag{16}$$

$$\frac{\partial L}{\partial \lambda_j} = 1 - l_{j1}^2 - l_{j2}^2, \tag{17}$$

$$\frac{\partial L}{\partial \ell_{j1}} = \frac{1}{N}\sum_{j=1}^{K}p_j(t)\frac{1}{\sigma_j^2}\{-(x_{t1}-m_{j1})[(x_{t1}-m_{j1})\ell_{j1} + (x_{t2}-m_{j2})\ell_{j2}]\}, \tag{18}$$

$$\frac{\partial L}{\partial m_{j1}} = \frac{1}{N}\sum_{j=1}^{K}p_j(t)\frac{1}{\sigma_j^2}\{-(x_{t1}-m_{j1}) + \ell_{j1}(x_t - m_j, \ell_j)\}, \tag{19}$$

$$\frac{\partial L}{\partial \sigma_j} = \frac{1}{N}\sum_{j=1}^{K}p_j(t)\{\frac{1}{\sigma_j} + \frac{1}{\sigma_j^3}d^2(x_t, \ell_j, m_j)\}. \tag{20}$$

By letting these derivatives given by Eqs. (15)-(20) be 0, we have

$$\beta = \frac{1}{N}\sum_{j=1}^{K}\sum_{t=1}^{N}p_j(t), \tag{21}$$

and further obtain the following fixed-point learning algorithm:

$$m_j^{h+1} = \frac{\sum_t p_j(t)x_t}{\sum_t p_j(t)}, \tag{22}$$

$$\pi_j^{h+1} = \frac{1}{N}\sum_t p_j(t), \tag{23}$$

$$(\sigma_j^{h+1})^2 = \frac{\sum_t p_j(t)d^2(x_t, \ell_k, m_k)}{\sum_t p_j(t)}, \tag{24}$$

and ℓ_j is the eigenvector of $\Sigma_j = \sum_{j=1}^{N} p_j(t)(x_t - m_j)(x_t - m_j)^T$ corresponding to the largest eigenvalue.

Based on the above fixed-point learning algorithm, we can establish the fixed-point EM algorithm which consists of the following three steps:

(i) Initialization of the parameters.
(ii) Update m_j, π_j, σ_j^2 by Eqs. (22)-(24). Update ℓ_j by the eigenvector of $\Sigma_j = \sum_{j=1}^{N} p_j(t)(x_t - m_j)(x_t - m_j)^T$ corresponding to the largest eigenvalue.
(iii) Repeat Step (ii) until the values of parameters are unchanged.

4 Experiments Results

In this section, several simulation experiments are carried out to demonstrate the performance of the fixed-point EM algorithm for straight line detection. We consider the binary images or datasets of four straight lines with three kinds of noises. The true parameters of the finite mixture models for the three datasets are listed in Table 1. Obviously, the noise levels in $\mathcal{S}_2$ and $\mathcal{S}_3$ are much higher than that of $\mathcal{S}_1$.

In our experiments, we set the number of straight lines to be the true number of straight lines, i.e., $K = 4$. We implement the fixed-point EM algorithm on each dataset, with the parameters being initialized by the random Hough transform [3]. The algorithm stops if $|Q(\Theta_K^{new}) - Q(\Theta_K^{old})| < 10^{-6}$. The results of the straight line detection as well as the obtained Q-function are shown in Fig. 1-3, respectively. The learned parameters of the finite mixture model on each experiment are listed in Table. 2.

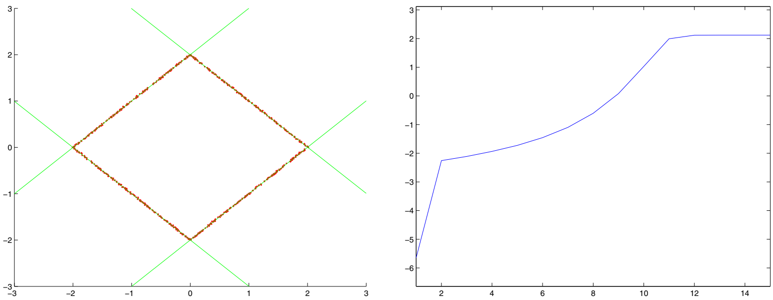
Table 1. The true parameters of the finite mixture models for the datasets $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$, respectively

Sample set	π_i	ℓ_i	m_i	σ_i
$\mathcal{S}_1$	0.25	(-0.7071,0.7071)	(1,1)	0.01
	0.25	(-0.7071,0.7071)	(-1,1)	0.01
	0.25	(-0.7071,0.7071)	(-1,-1)	0.01
	0.25	(0.7071,0.7071)	(1,-1)	0.01
$\mathcal{S}_2$	0.25	(-0.7071,0.7071)	(1,1)	0.2
	0.25	(-0.7071,0.7071)	(-1,1)	0.2
	0.25	(-0.7071,0.7071)	(-1,-1	0.2
	0.25	(0.7071,0.7071)	(1,-1)	0.2
$\mathcal{S}_3$	0.25	(-0.7071,0.7071	(1,1)	0.3
	0.25	(-0.7071,0.7071	(-1,1)	0.3
	0.25	(-0.7071,0.7071	(-1,-1	0.3
	0.25	(0.7071,0.7071)	(1,-1)	0.3

It can be observed from the figures in Fig. 1 that the Q-function increases during the iterations and finally reaches its maximum. Meanwhile, the straight lines are accurately located in each case. We can also observe that the Q-function increases sharply at the beginning of the iterations. The reason may be that the parameters are initialized by the random Hough transform, being only some rough estimates of the parameters. As the initialization of the parameters becomes better, the curve of the Q-function will be more smooth.

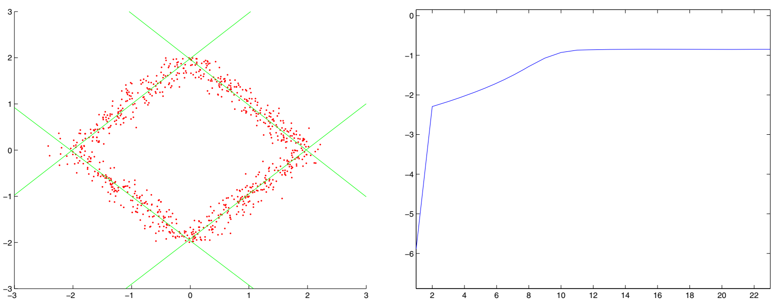
From these experimental results, we are sure that our proposed fixed-point EM algorithm can effectively detect the straight lines in all the three datasets with different noise levels. Moreover, it is shown in Fig. 3-(c) that the fixed-point EM algorithm also performs well on the strongly noisy situation.

It is pity that the number of straight lines in the learning mixture model should be known in advance. But this information may be not available in



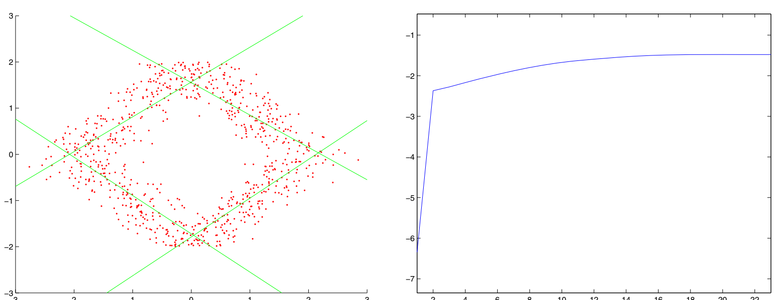
(a) (b)

Fig. 1. (a). The experimental result of straight line detection on the dataset $\mathcal{S}_1$, (b). The sketch of the Q-function on the iterations



(a) (b)

Fig. 2. (a). The experimental result of straight line detection on the dataset $\mathcal{S}_2$, (b). The sketch of the Q-function on the iterations



(a) (b)

Fig. 3. (a). The experimental result of straight line detection on the dataset $\mathcal{S}_3$, (b). The sketch of the Q-function on the iterations

Table 2. The learned parameters of the finite mixture models on the three datasets $\mathcal{S}_1, \mathcal{S}_2$ and $\mathcal{S}_3$, respectively

Sample set	π_i	ℓ_i	m_i	σ_i
$\mathcal{S}_1$	0.2498	(-0.7084,0.7058)	(0.9715,1.0290)	0.0103
	0.2489	(-0.7084,0.7058)	(-0.9695,1.0302)	0.0083
	0.2502	(-0.7077,0.7065)	(-0.9516,-1.0491)	0.0089
	0.2511	(0.7067,0.7075)	(1.0120,-0.9871)	0.0097
$\mathcal{S}_2$	0.2617	(0.7101,-0.7041)	(1.0192,0.9542)	0.1726
	0.2421	(0.7119, 0.7023)	(-1.0138,0.9835)	0.1812
	0.2380	(0.7210,-0.6930)	(-0.9720,-1.0285)	0.1760
	0.2582	(-0.7157,-0.6984)	(0.9778,-0.9707)	0.2138
$\mathcal{S}_3$	0.2386	(-0.7908,0.6120)	(0.8080,1.0306)	0.2695
	0.2557	(-0.7459,-0.6661)	(-0.8380,1.0016)	0.2826
	0.2335	(0.7296,-0.6839)	(-0.8854,-0.9882)	0.2867
	0.2722	(-0.7323,-0.6810)	(0.8837,-0.9461)	0.3386

practical applications. In order to overcome this weakness, we can introduce the Bayesian Ying-Yang (BYY) harmony learning system [17]-[18] and the entropy penalized automated model selection mechanism [19] into the fixed-point learning algorithm.

5 Conclusions

We have investigated the straight line detection problem from the finite mixture modeling. Since it is difficulty to solve the maximum of the log-likelihood function, we resort to the EM algorithm and analyze the Q-function. By differentiation, we derive a fixed-point learning procedure for maximizing the Q-function and thus construct a fixed-point EM algorithm for straight line detection. It is demonstrated by the experiments that the proposed fixed-point EM algorithm can effectively locate the straight lines in a dataset.

Acknowledgements

This work was supported by the Natural Science Foundation of China for Grant 60771061.

References

1. Ballard, D.: Generalizing the Hough Transform to Detect Arbitrary Shapes. Pattern Recognition 13(2), 111–122 (1981)
2. Wolfson, H.: Generalizing the Generalized Hough Transform. Pattern Recognition Letters 12(9), 565–573 (1991)
3. Xu, L., Oja, E., Kultanen, P.: A New Curve Detection Method - Randomized Hough Transform (RHT). Pattern Recognition Letters 11(5), 331–338 (1990)

4. Olson, C.F.: Constrained Hough Transforms for Curve Detection. Computer Vision and Image Understanding 73, 329–345 (1998)
5. Liu, Z., Qiao, H., Xu, L.: Multisets Mixture Learning-Based Ellipse Detection. Pattern Recognition 39(4), 731–735 (2006)
6. Liu, Z., Chiu, K., Xu, L.: Strip Line Detection and Thinning by RPCL-Based Local PCA. Pattern Recognition Letters 24(14), 2335–2344 (2003)
7. Lu, Z.W., Cheng, Q.S., Ma, J.W.: A Gradient BYY Harmony Learning Algorithm on Mixture of Experts for Curve Detection. In: Gallagher, M., Hogan, J.P., Maire, F. (eds.) IDEAL 2005. LNCS, vol. 3578, pp. 250–257. Springer, Heidelberg (2005)
8. Chen, G., Li, L., Ma, J.: A Gradient BYY Harmony Learning Algorithm for Straight Line Detection. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part I. LNCS, vol. 5263, pp. 618–626. Springer, Heidelberg (2008)
9. Xu, L.: Multisets Modeling Learning: an Unified Theory for Supervised and Unsupervised Learning. In: IEEE International Conference on Neural Networks, 1994. IEEE World Congress on Computational Intelligence, vol. 1, pp. 315–320 (1994)
10. Kambhatla, N., Leen, T.: Dimension Reduction by Local Principal Component Analysis. Neural Computation 9(7), 1493–1516 (1997)
11. Xu, L.: Vector Quantization by Local and Hierarchical LMSER. In: Proc. of 1995 Intl. Conf. on Artificial Neural Networks, vol. 2, pp. 575–579 (1995)
12. Xu, L.: An Overview on Unsupervised Learning from Data Mining Perspective. In: Allinson, N., Yin, H., Allinson, L., Slack, J. (eds.) Advances in Self-Organising Maps, pp. 181–209 (2001)
13. Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, Heidelberg (2002)
14. Xu, L.: Least Mean-Square Error Reconstruction Principle for Self-Organizing Neural-Nets. Neural Networks 6(5), 627–648 (1993)
15. Xu, L., Krzyzak, A., Oja, E.: Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. IEEE Transactions on Neural Networks 4(4), 636–649 (1993)
16. Dempster, A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via EM Algorithm. Journal of the Royal Statistical Society Series B-Methodological 39(1), 1–38 (1977)
17. Ma, J.W., He, X.F.: A Fast Fixed-Point BYY Harmony Learning Algorithm on Gaussian Mixture with Automated Model Selection. Pattern Recognition Letters 29(6), 701–711 (2008)
18. Ma, J.W., Liu, J.F.: The BYY Annealing Learning Algorithm for Gaussian Mixture with Automated Model Selection. Pattern Recognition 40(7), 2029–2037 (2007)
19. Ma, J.W., Wang, T.J.: Entropy Penalized Automated Model Selection on Gaussian Mixture. International Journal of Pattern Recognition and Artificial Intelligence 18(8), 1501–1512 (2004)