Fast Discriminative Component Analysis for Comparing Examples

Jaakko Peltonen[1], Jacob Goldberger[2], and Samuel Kaski[1]

[1]Helsinki Institute for Information Technology & Adaptive Informatics Research Centre, Laboratory of Computer and Information Science, Helsinki University of Technology

[2]School of Engineering, Bar-Ilan University

Outline

1. Background
2. Our method
3. Optimization
4. Properties
5. Experiments
6. Conclusions

1. Background

Task: *discriminative component analysis*

(searching for data components that discriminate some auxiliary data of interest, e.g. classes)

1. Background

Task: *discriminative component analysis*

(searching for data components that discriminate some auxiliary data of interest, e.g. classes)

Another application possibility: *supervised unsupervised learning*

1. Background

Linear Discriminant Analysis:
well-known classical method.

1. Background

Linear Discriminant Analysis:
well-known classical method.

Optimal subspace under restrictive assumptions: Gaussian classes with equal cov. matrix, take enough components.

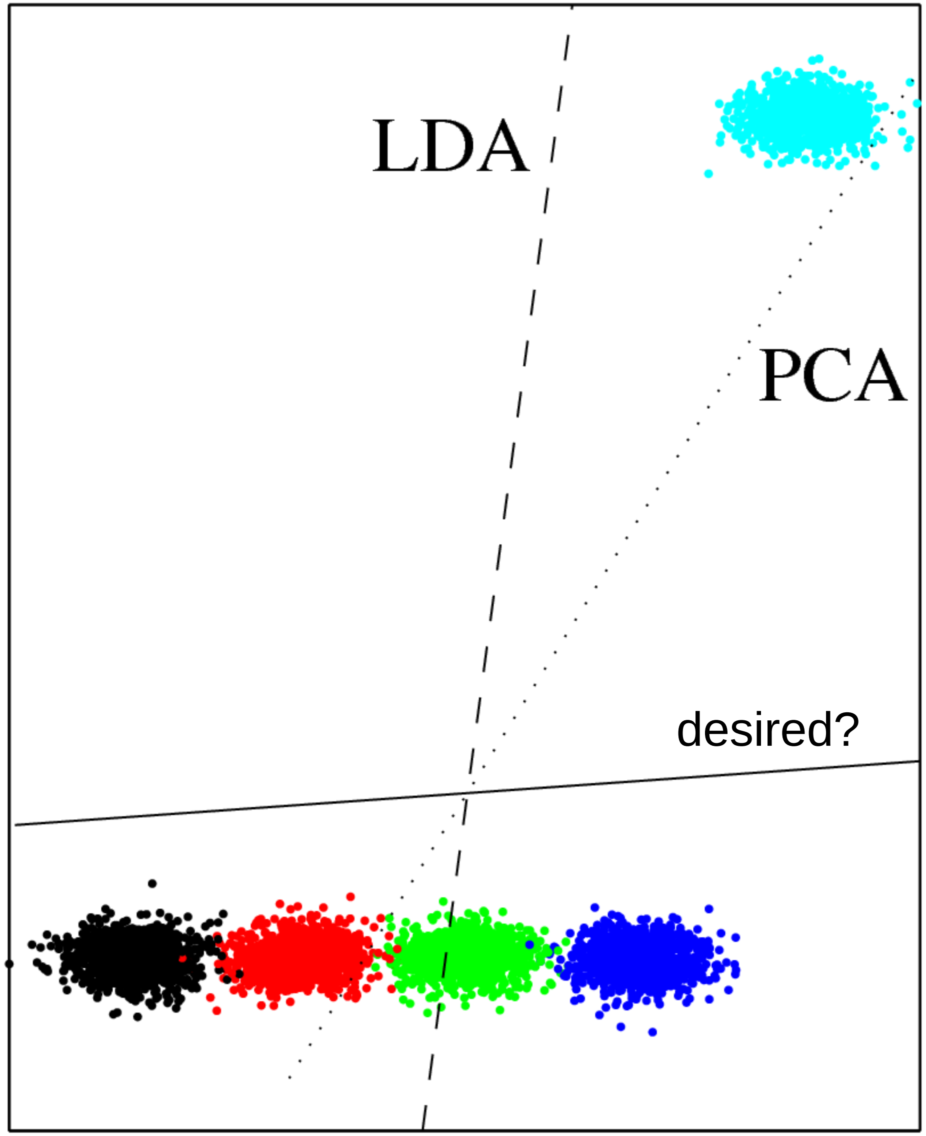
Not optimal otherwise!

1. Background

Linear Discriminant Analysis: well-known classical method.

Optimal subspace under restrictive assumptions: Gaussian classes with equal cov. matrix, take enough components.

Not optimal otherwise!

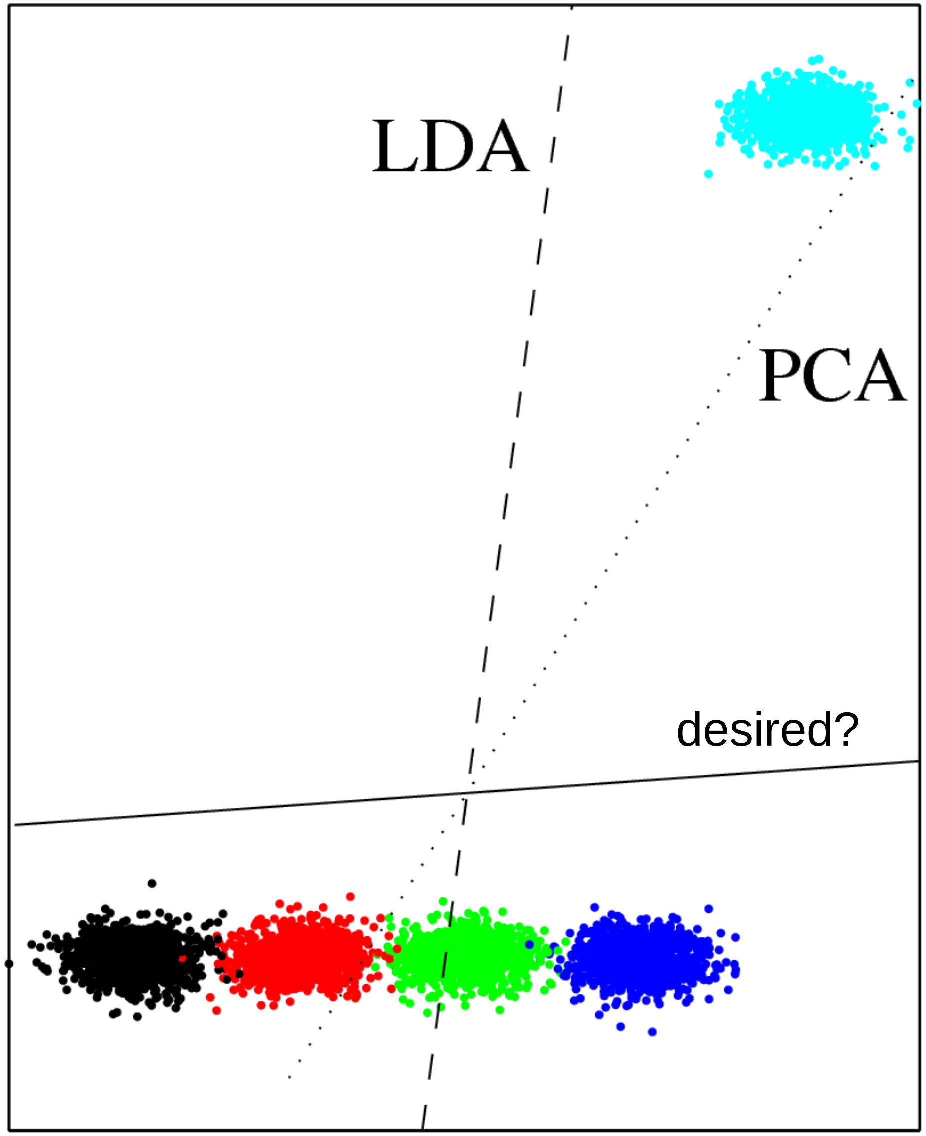


1. Background

Linear Discriminant Analysis:
well-known classical method.

Optimal subspace under restrictive assumptions: Gaussian classes with equal cov. matrix, take enough components.

Not optimal otherwise!



Extensions: HDA, reduced-rank MDA. LDA and many extensions can be seen as models that maximize **joint likelihood** of (x,c)

1. Background

Recent discriminative methods:

1. Background

Recent discriminative methods:

- Information-theoretic methods
  (Torkkola - Renyi entropy based; Leiva-Murillo & Artés-Rodríquez)

1. Background

Recent discriminative methods:

- Information-theoretic methods
  (Torkkola - Renyi entropy based; Leiva-Murillo & Artés-Rodríquez)
- Likelihood ratio-based (Zhu & Hastie)

1. Background

Recent discriminative methods:

- Information-theoretic methods
  (Torkkola - Renyi entropy based; Leiva-Murillo & Artés-Rodríquez)
- Likelihood ratio-based (Zhu & Hastie)
- Kernel-based (Fukumizu et al.)

1. Background

Recent discriminative methods:

- Information-theoretic methods
  (Torkkola - Renyi entropy based; Leiva-Murillo & Artés-Rodríquez)
- Likelihood ratio-based (Zhu & Hastie)
- Kernel-based (Fukumizu et al.)
- Other approaches (e.g. Globerson & Roweis, Hammer & Villmann, ...)

1. Background

Recent discriminative methods:

- Information-theoretic methods
  (Torkkola - Renyi entropy based; Leiva-Murillo & Artés-Rodríquez)
- Likelihood ratio-based (Zhu & Hastie)
- Kernel-based (Fukumizu et al.)
- Other approaches (e.g. Globerson & Roweis, Hammer & Villmann, ...)

Two recent very similar methods:

Informative Discriminant Analysis (IDA)

Neighborhood Components Analysis (NCA)

1. Background

Recent discriminative methods:

- Information-theoretic methods
  (Torkkola - Renyi entropy based; Leiva-Murillo & Artés-Rodríquez)
- Likelihood ratio-based (Zhu & Hastie)
- Kernel-based (Fukumizu et al.)
- Other approaches (e.g. Globerson & Roweis, Hammer & Villmann, ...)

Two recent very similar methods:

Informative Discriminant Analysis (IDA)
Neighborhood Components Analysis (NCA)

Nonparametric: no distributional assumptions, but $O(N^2)$ complexity per iteration.

2. Our Method

Basic idea: instead of optimizing the metric for a nonparametric predictor, optimize it for a **parametric predictor**

2. Our Method

Basic idea: instead of optimizing the metric for a nonparametric predictor, optimize it for a **parametric predictor**

Parametric predictors are much simpler than nonparametric ones: much **less computation**, and can increase **robustness**

2. Our Method

Basic idea: instead of optimizing the metric for a nonparametric predictor, optimize it for a **parametric predictor**

Parametric predictors are much simpler than nonparametric ones: much **less computation**, and can increase **robustness**

Of course, then you have to optimize the predictor parameters too...

2. Our Method

Parametric predictor: mixture of labeled Gaussians

$$p\left(\boldsymbol{Ax}, c; \boldsymbol{\theta}\right) = \sum_{k} \alpha_c \beta_{c,k} N\left(\boldsymbol{Ax}; \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_c\right)$$

2. Our Method

Parametric predictor: mixture of labeled Gaussians

$$p(\boldsymbol{Ax}, c; \boldsymbol{\theta}) = \sum_k \alpha_c \beta_{c,k} N(\boldsymbol{Ax}; \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_c)$$

Objective function: conditional likelihood of classes

$$L = \sum_i p(c_i \mid \boldsymbol{Ax}_i; \boldsymbol{\theta}) = \sum_i \frac{p(\boldsymbol{Ax}_i, c_i; \boldsymbol{\theta})}{\sum_c p(\boldsymbol{Ax}_i, c; \boldsymbol{\theta})}$$

2. Our Method

Parametric predictor: mixture of labeled Gaussians

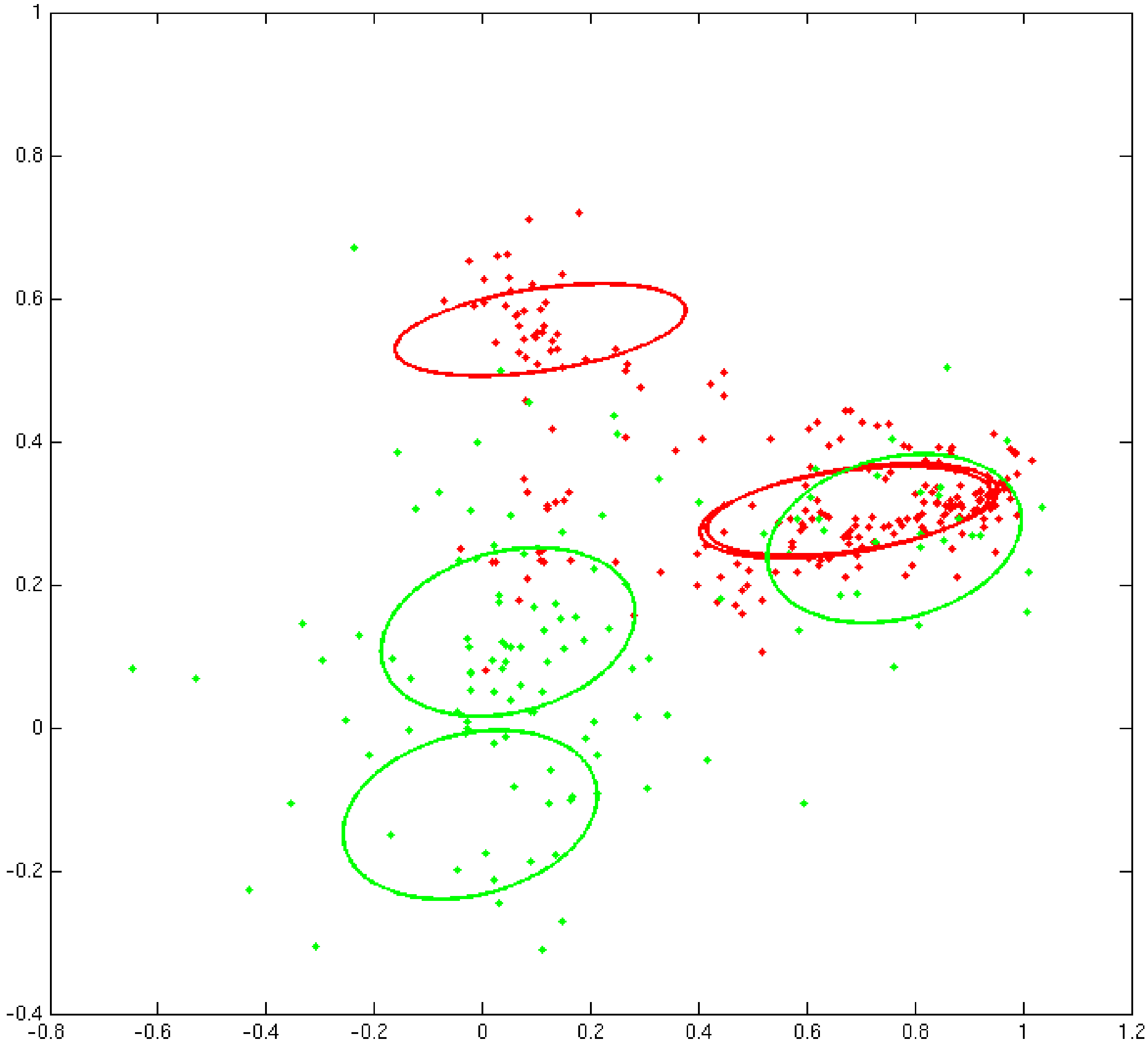
$$p\left(\boldsymbol{Ax},c;\boldsymbol{\theta}\right)=\sum_{k}\alpha_{c}\beta_{c,k}N\left(\boldsymbol{Ax};\boldsymbol{\mu}_{c,k},\boldsymbol{\Sigma}_{c}\right)$$

Objective function: conditional likelihood of classes

$$L=\sum_{i}p\left(c_{i}\mid\boldsymbol{Ax}_{i};\boldsymbol{\theta}\right)=\sum_{i}\frac{p\left(\boldsymbol{Ax}_{i},c_{i};\boldsymbol{\theta}\right)}{\sum_{c}p\left(\boldsymbol{Ax}_{i},c;\boldsymbol{\theta}\right)}$$

We call this “discriminative component analysis by Gaussian mixtures” or DCA-GM

DCA-GM



3. Optimization

Use gradient descent for the matrix $\boldsymbol{A}$

$$\frac{\partial L}{\partial \boldsymbol{A}} = \sum_{i,c,k} \left(p(c,k \mid \boldsymbol{A}\boldsymbol{x};\boldsymbol{\theta}) - \delta_{c,c_i} p(k \mid \boldsymbol{A}\boldsymbol{x},c;\boldsymbol{\theta}) \right) (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\mu}_{c,k}) \boldsymbol{x}^T$$

3. Optimization

Use gradient descent for the matrix $\boldsymbol{A}$

$$\frac{\partial L}{\partial \boldsymbol{A}} = \sum_{i,c,k} \left(p\left(c,k \mid \boldsymbol{Ax};\boldsymbol{\theta}\right) - \delta_{c,c_i}\, p\left(k \mid \boldsymbol{Ax},c;\boldsymbol{\theta}\right) \right) \left(\boldsymbol{Ax} - \boldsymbol{\mu}_{c,k} \right) \boldsymbol{x}^T$$

$$\left(\begin{aligned} p\left(k \mid \boldsymbol{Ax},c;\boldsymbol{\theta}\right) &= \frac{\beta_{c,k} N\left(\boldsymbol{Ax};\boldsymbol{\mu}_{c,k},\boldsymbol{\Sigma}_c\right)}{\sum \beta_{c,l} N\left(\boldsymbol{Ax};\boldsymbol{\mu}_{c,l},\boldsymbol{\Sigma}_c\right)} \\ p\left(c,k \mid \boldsymbol{Ax};\boldsymbol{\theta}\right) &= \frac{\alpha_c \beta_{c,k} N\left(\boldsymbol{Ax};\boldsymbol{\mu}_{c,k},\boldsymbol{\Sigma}_c\right)}{\sum \alpha_{c'} \beta_{c',k} N\left(\boldsymbol{Ax};\boldsymbol{\mu}_{c',k},\boldsymbol{\Sigma}_{c'}\right)} \end{aligned} \right)$$

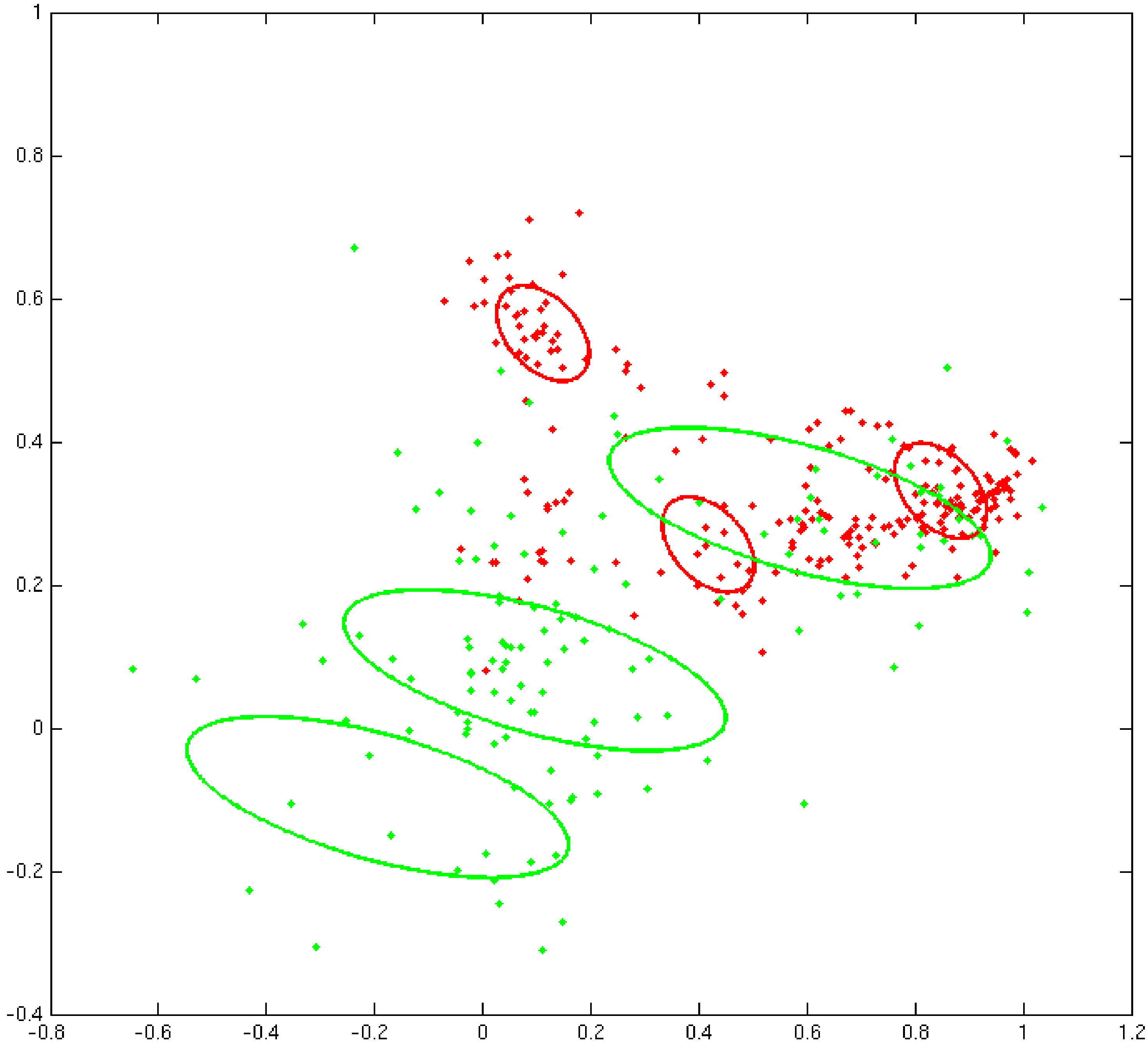
3. Optimization

We could optimize the mixture model parameters by conjugate gradient too.

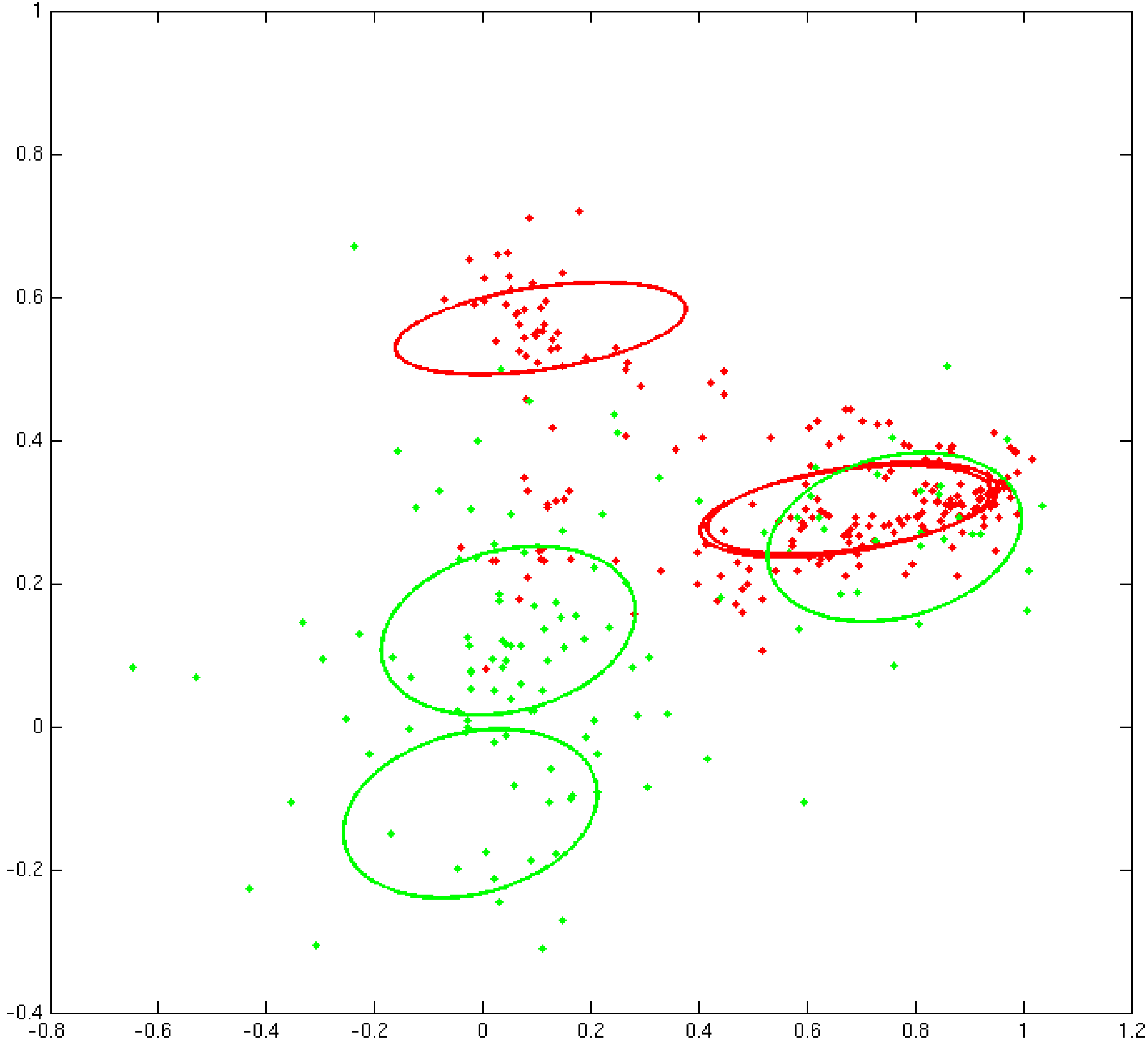
But here we will use a hybrid approach: we optimize the mixture by EM before each conjugate gradient iteration.

Then only the projection matrix A needs to be optimized by conjugate gradient.

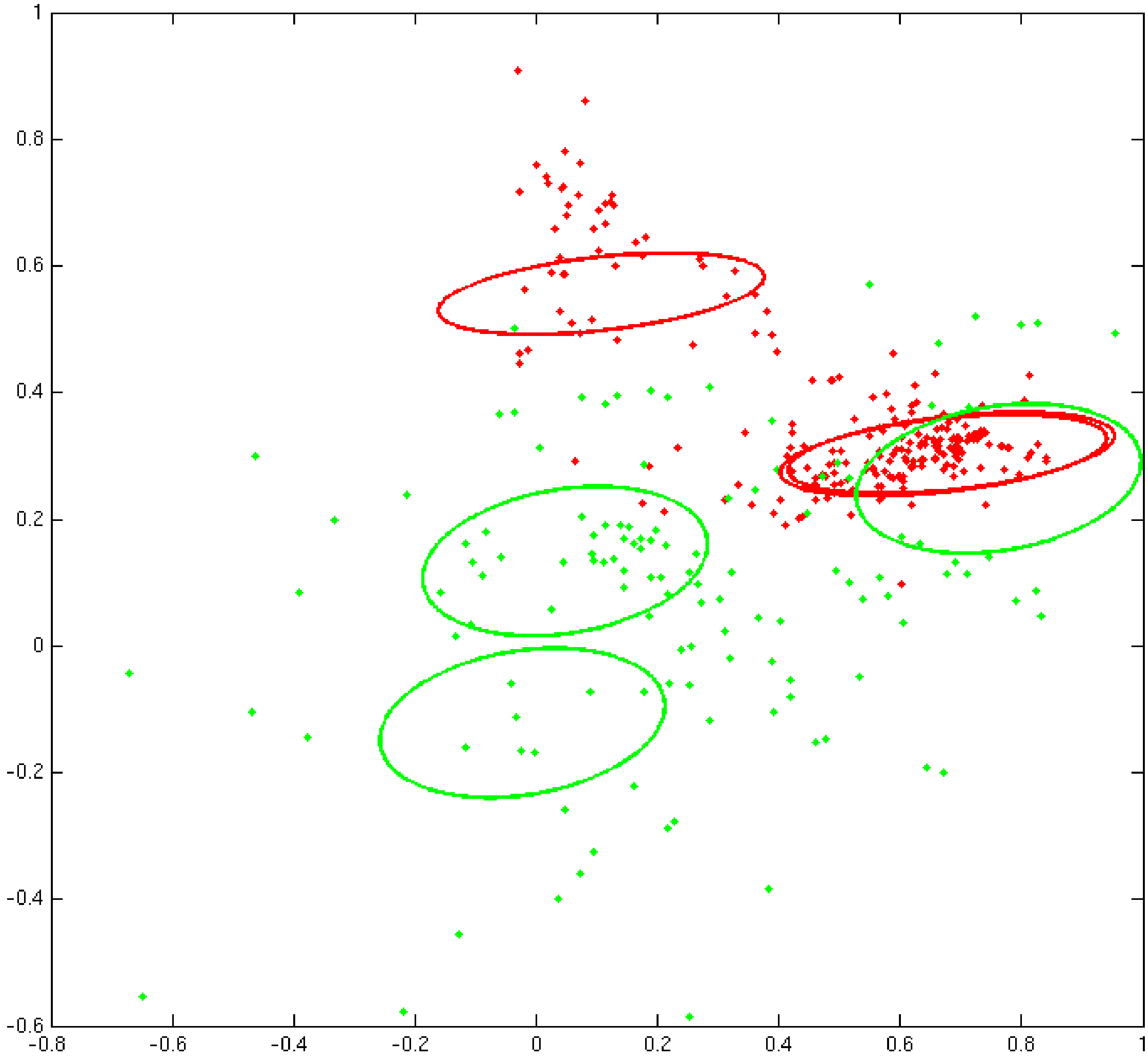
Initialization



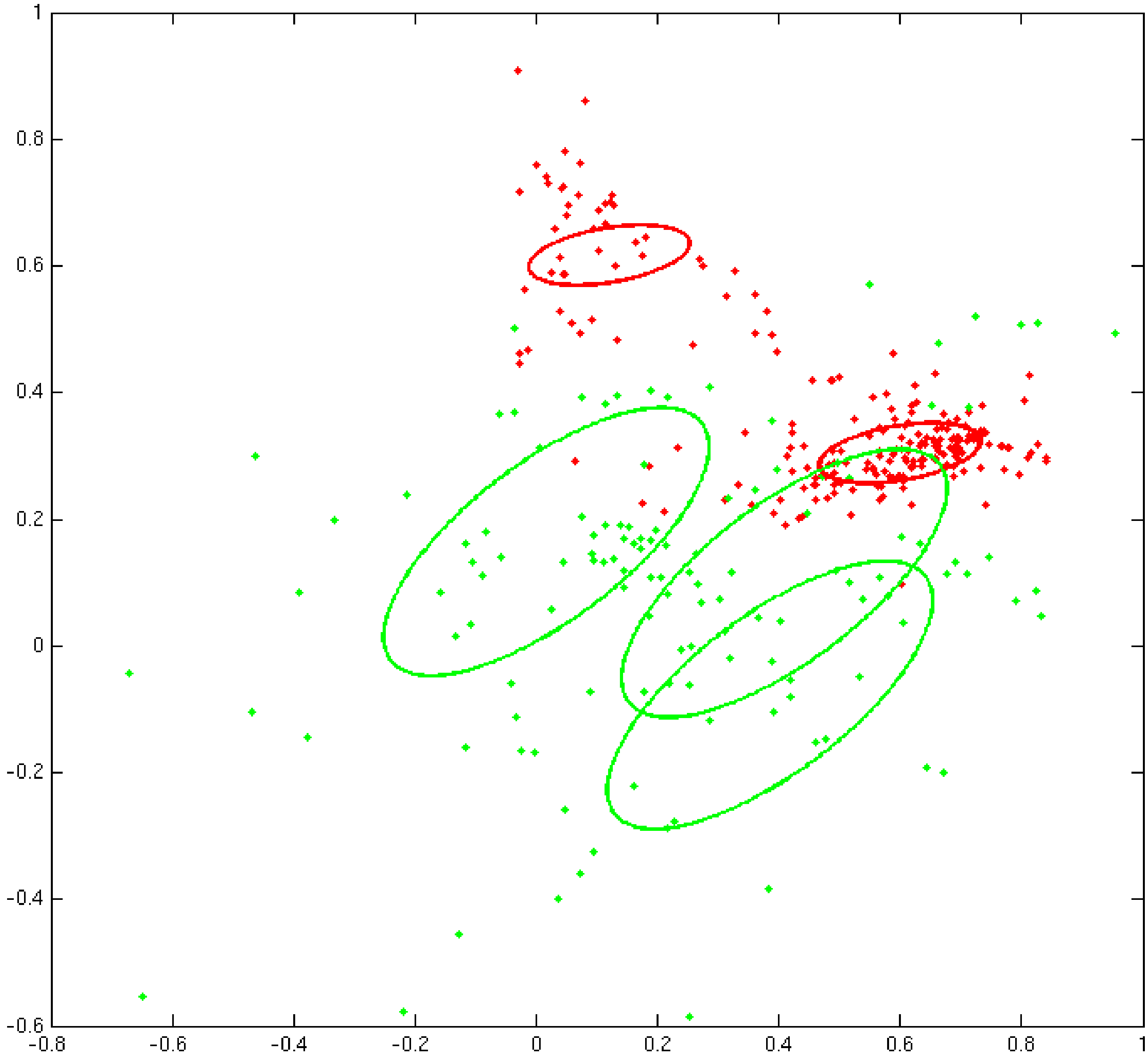
Iteration 1, after EM



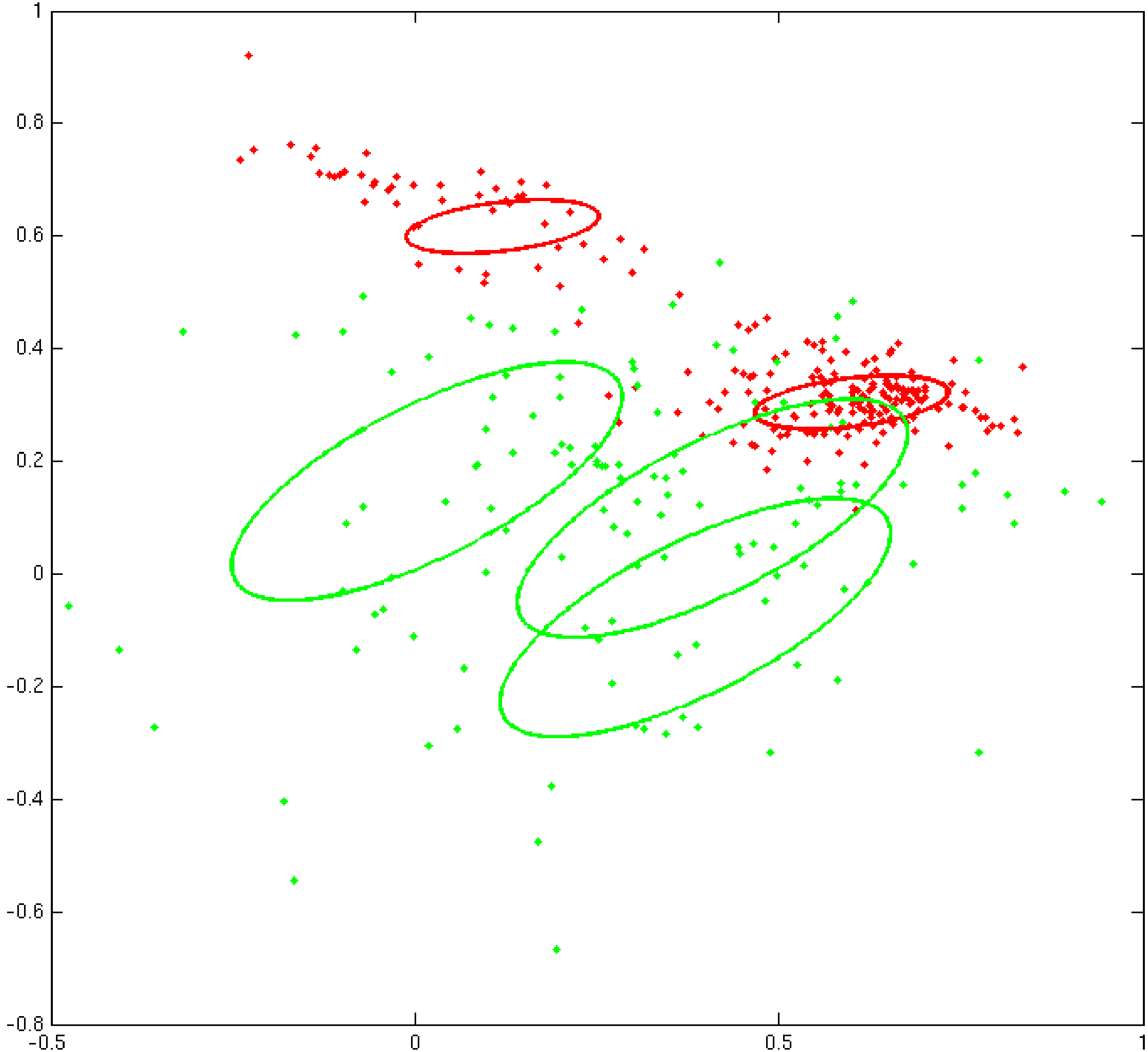
Iteration 1, after CG



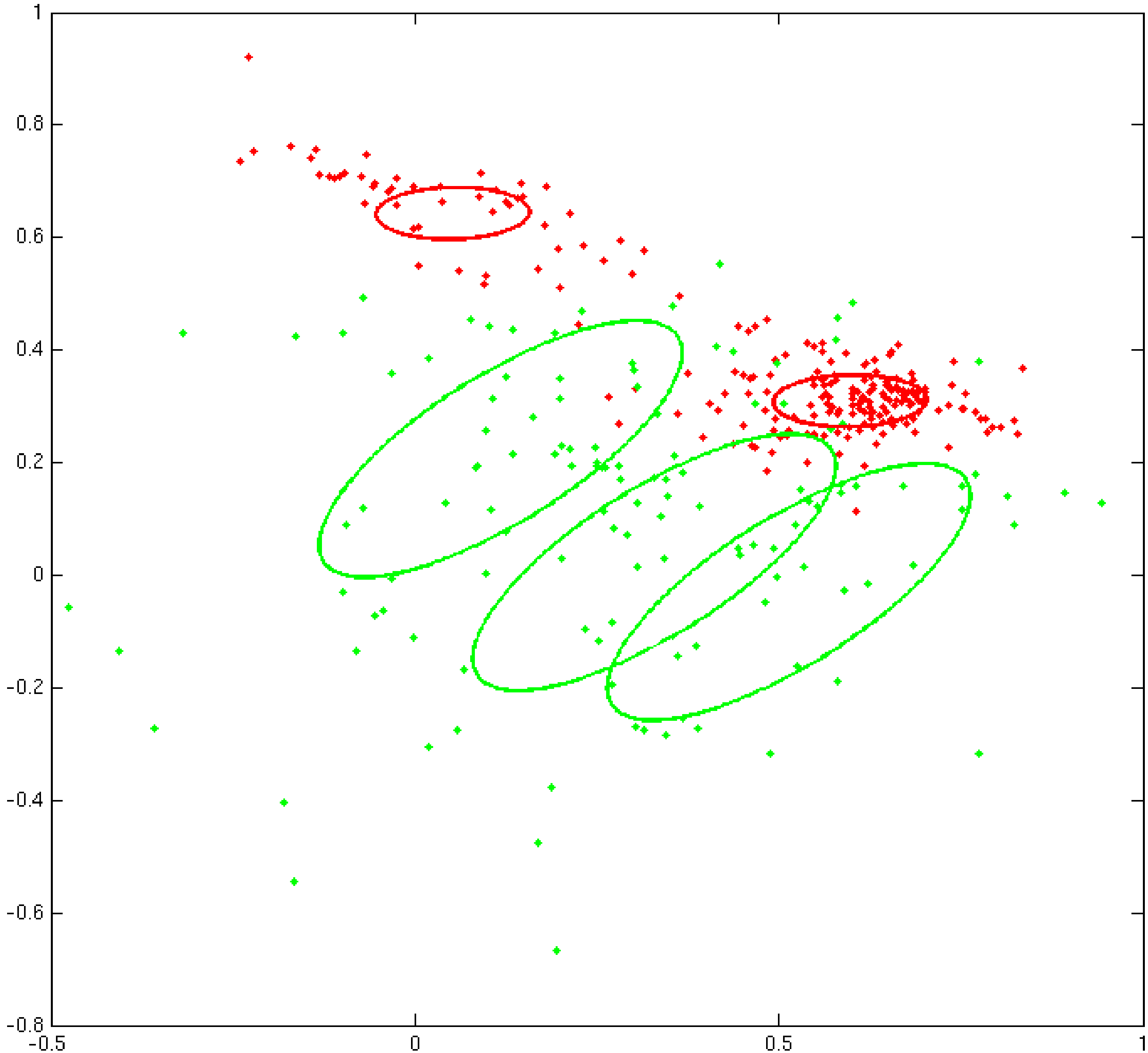
Iteration 2, after EM



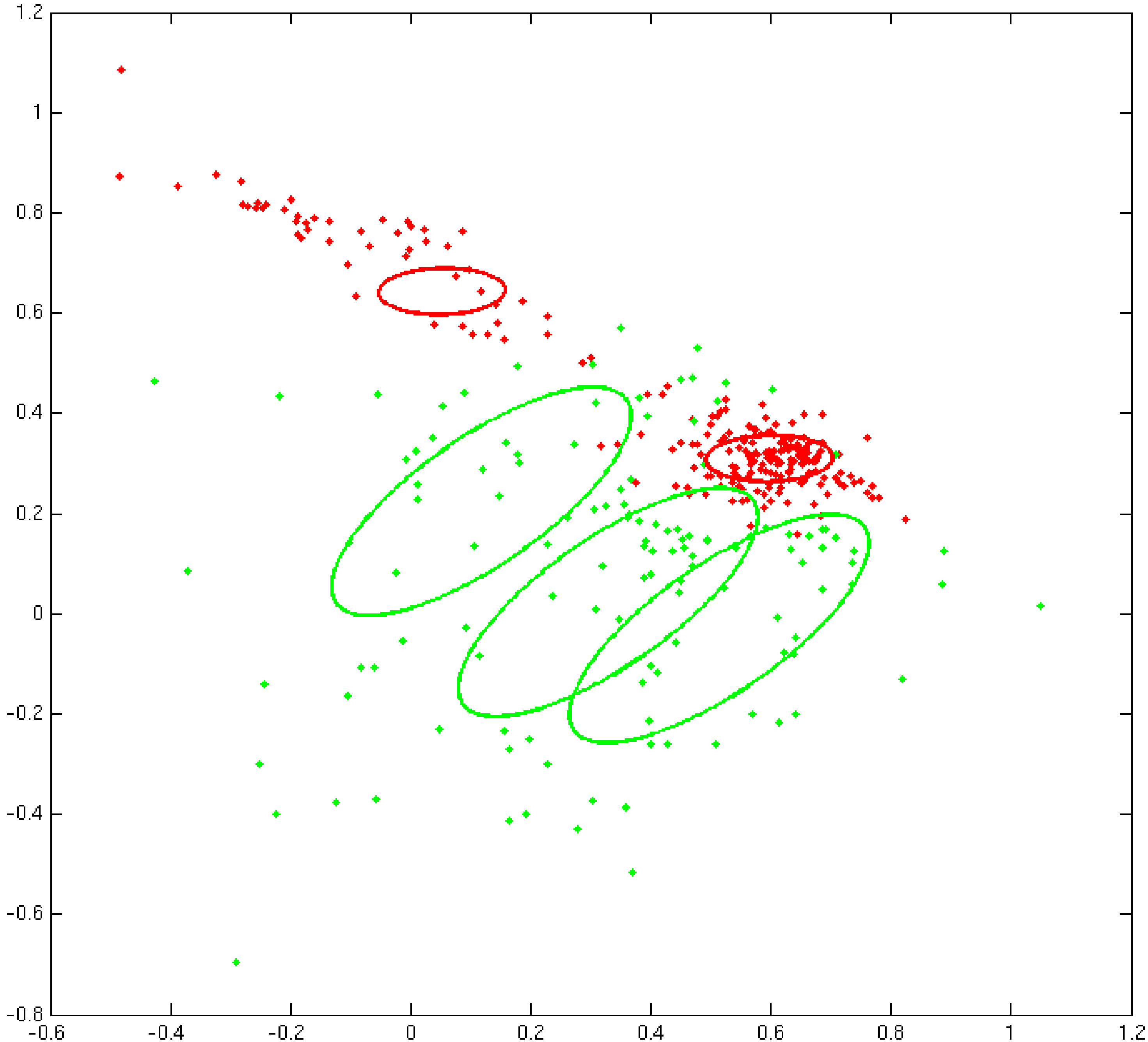
Iteration 2, after CG



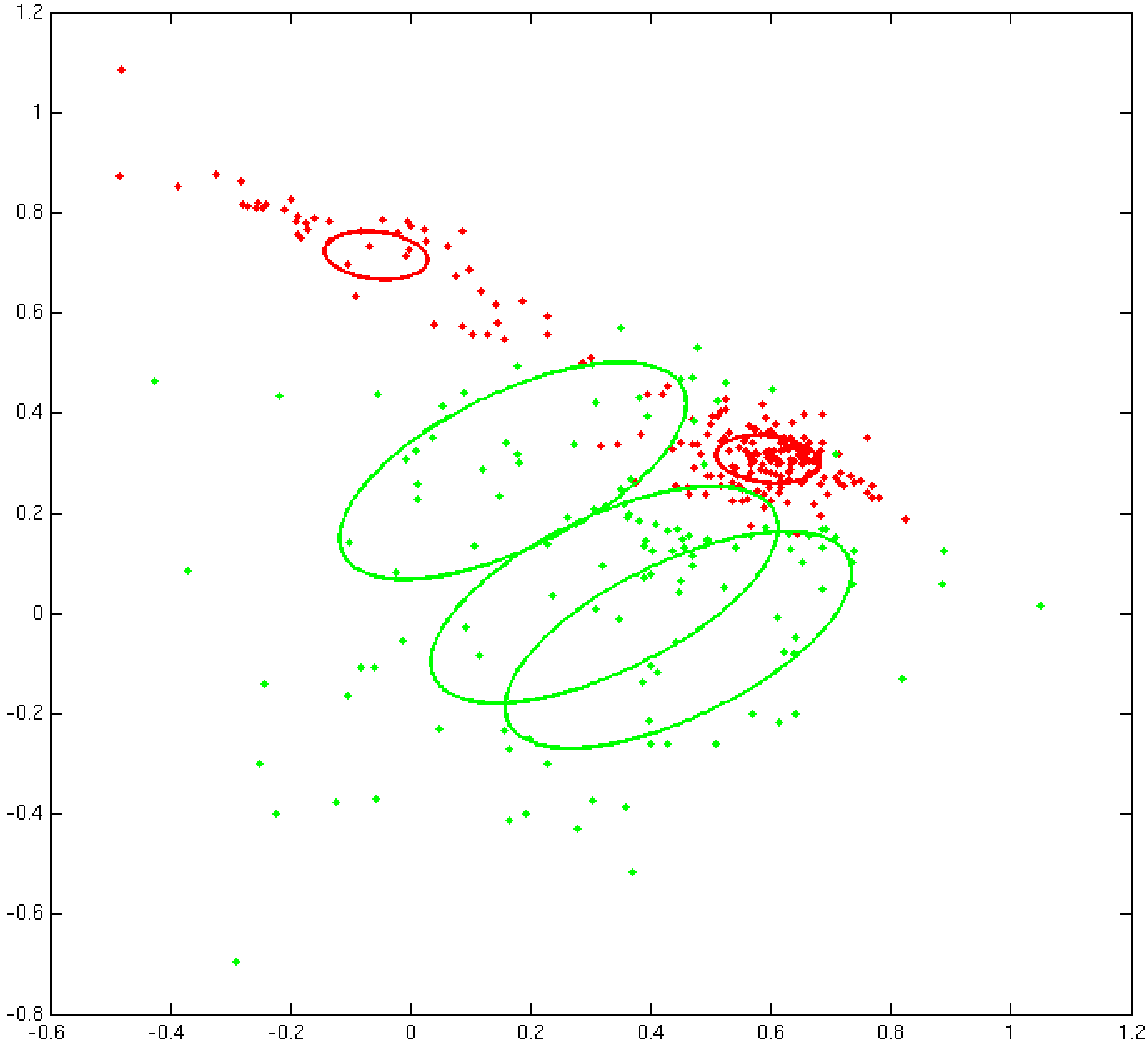
Iteration 3, after EM



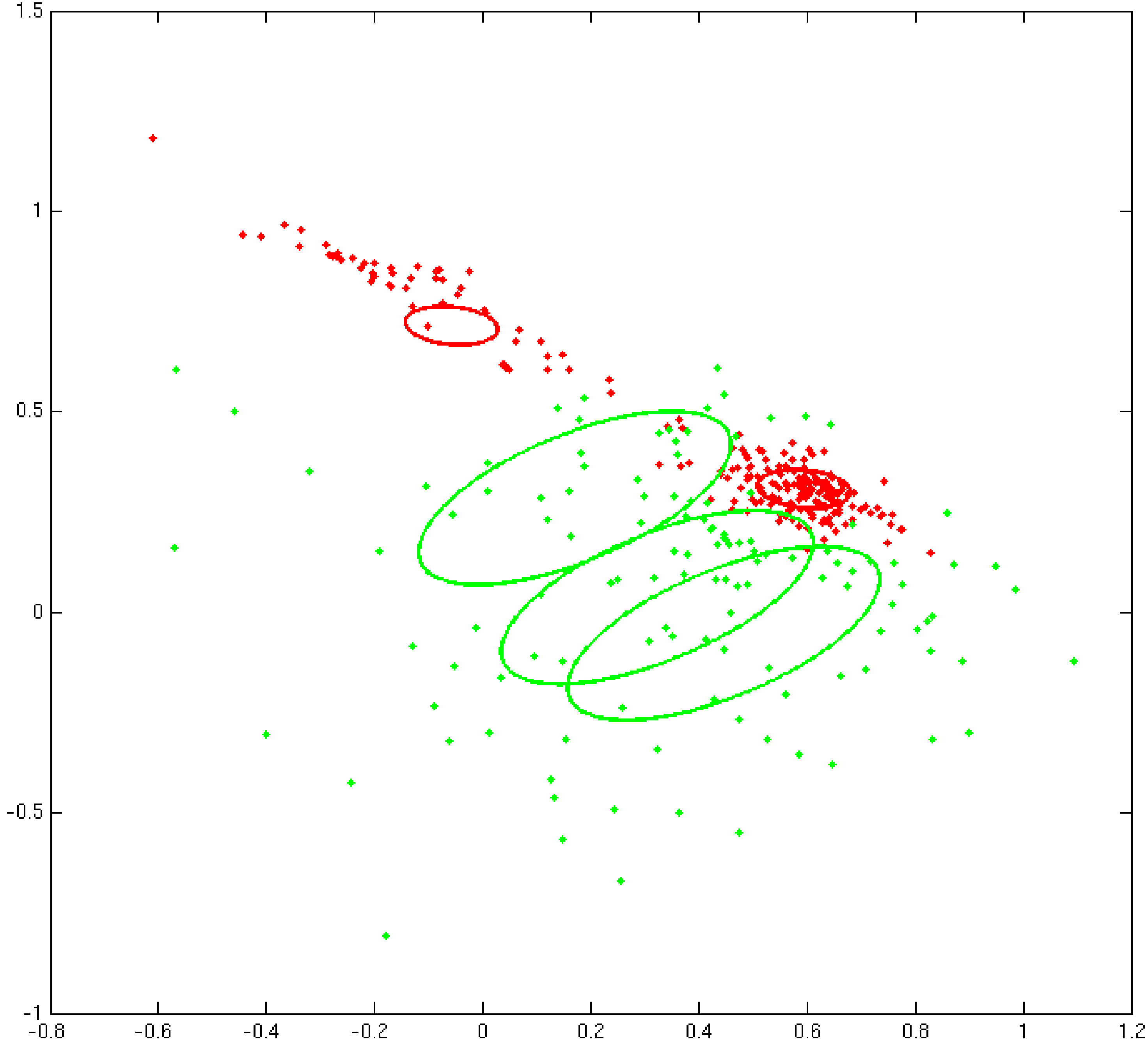
Iteration 3, after CG



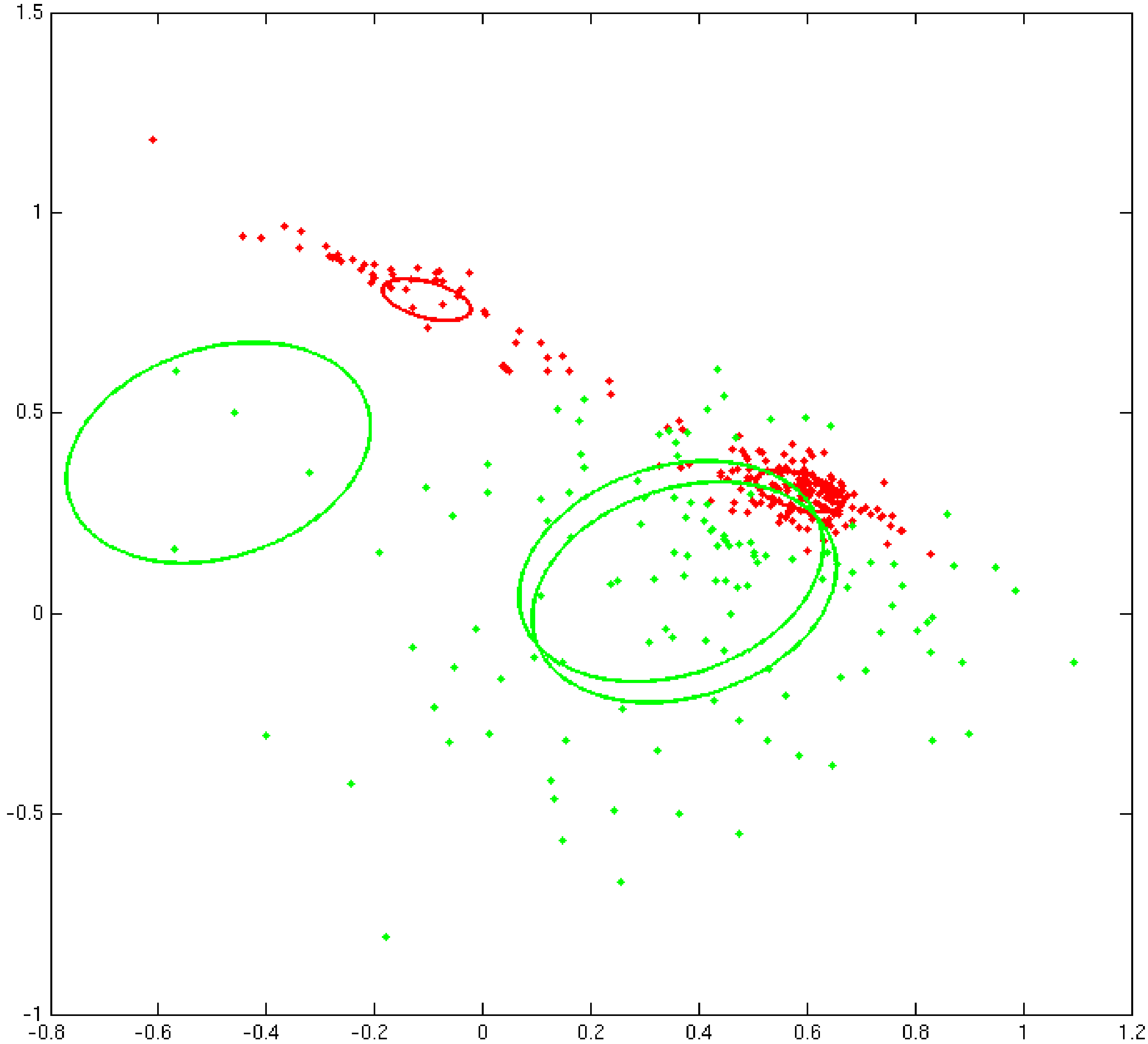
Iteration 4, after EM



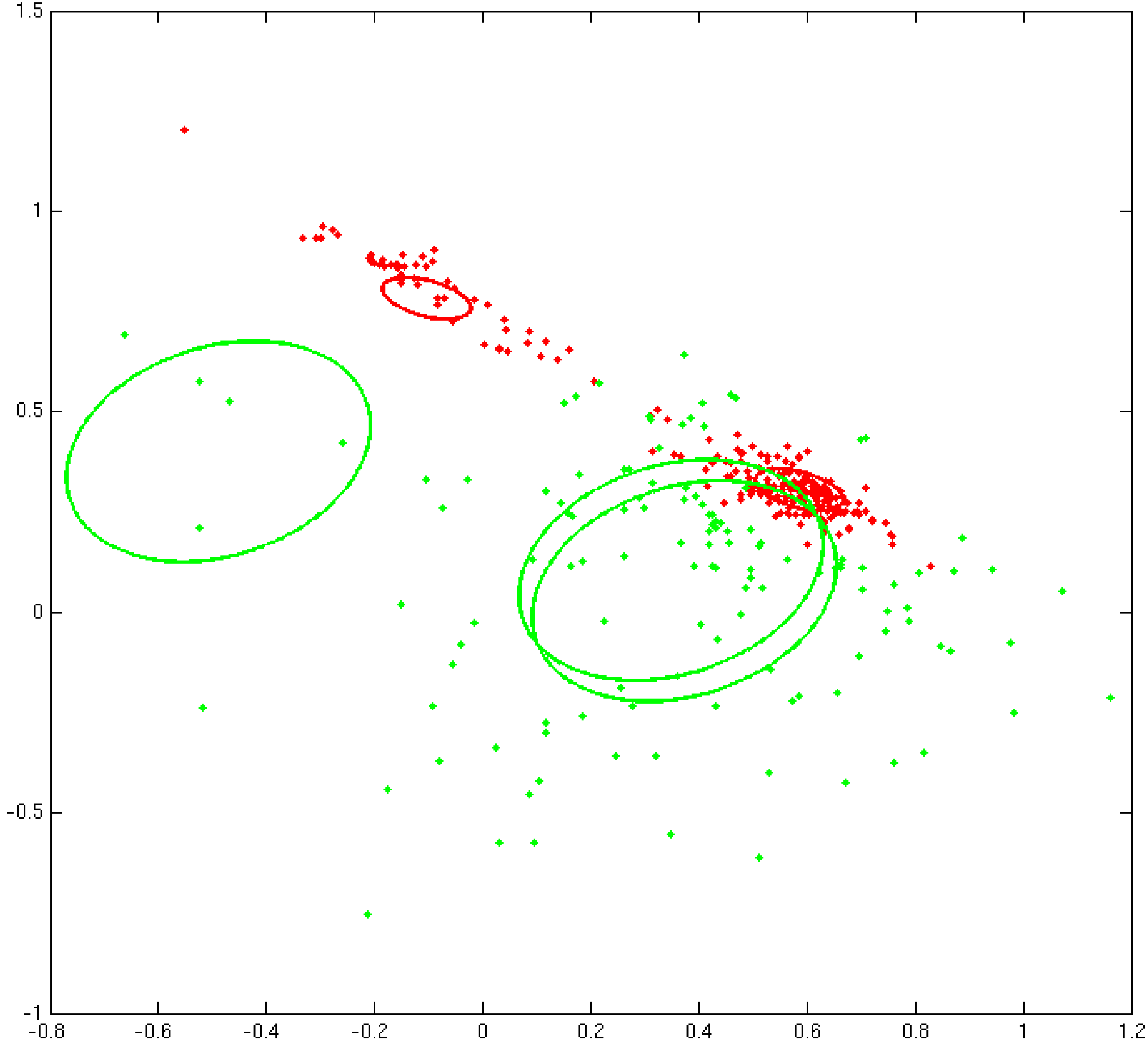
Iteration 4, after CG



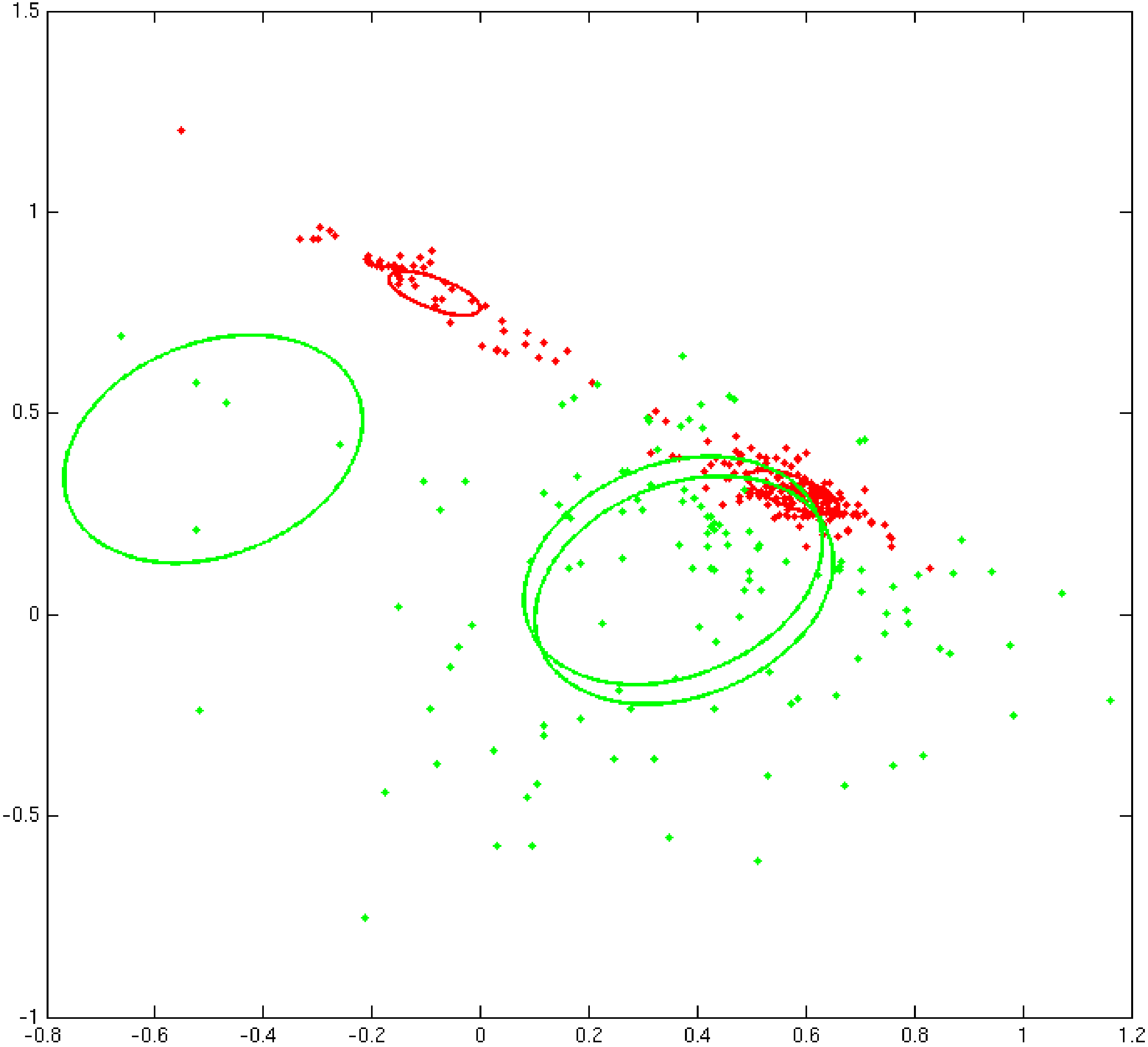
Iteration 5, after EM



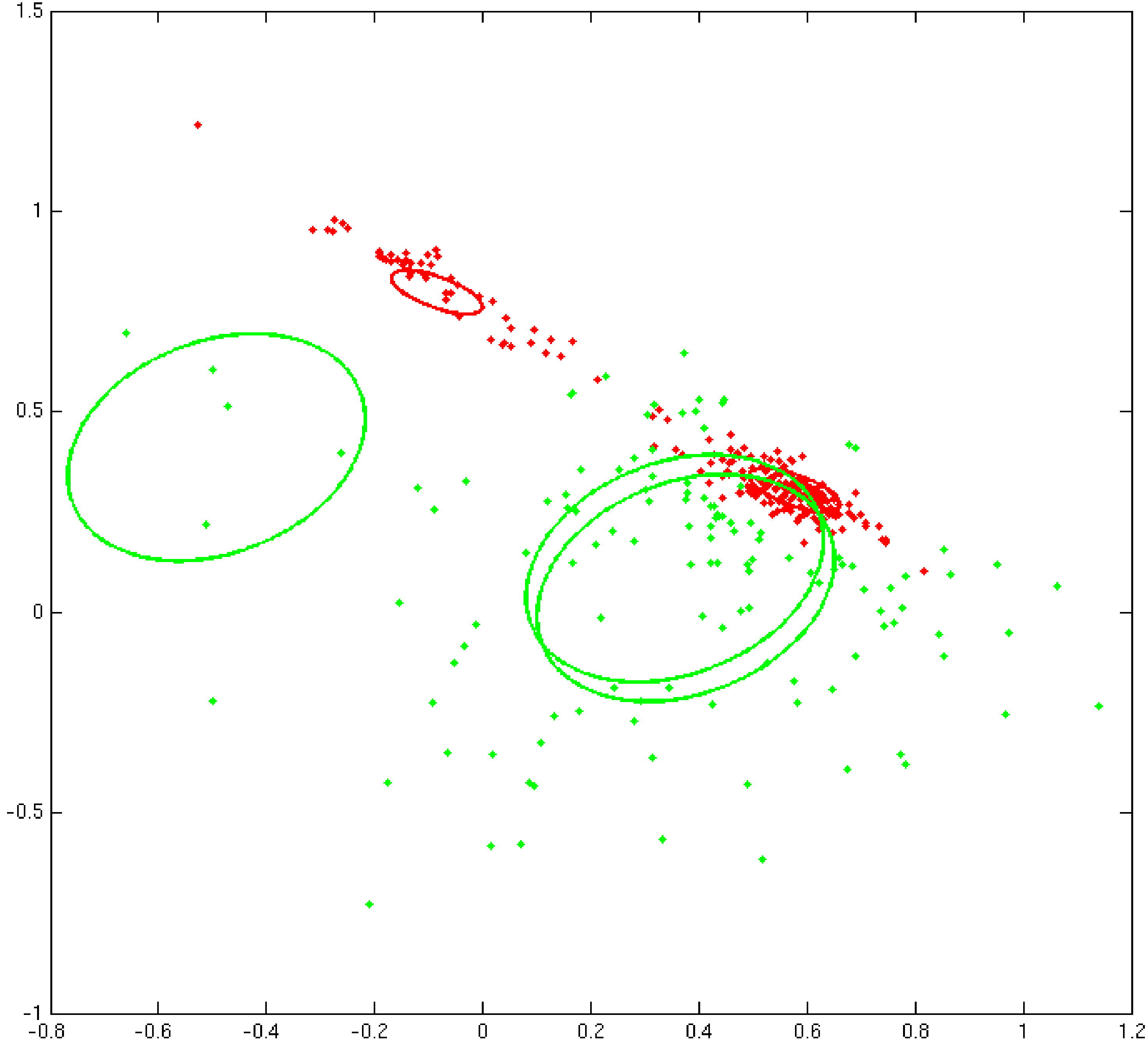
Iteration 5, after CG



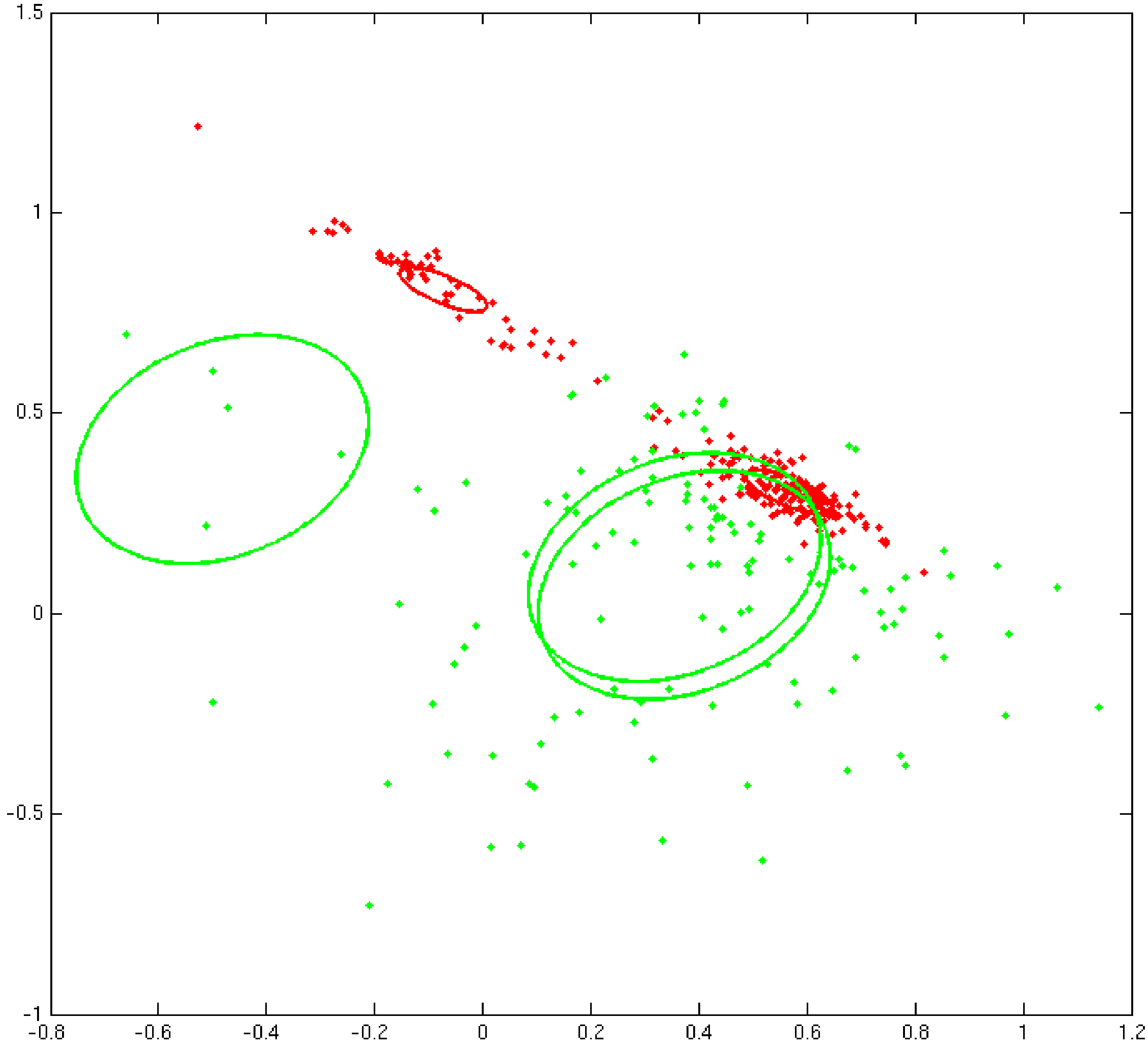
Iteration 6, after EM



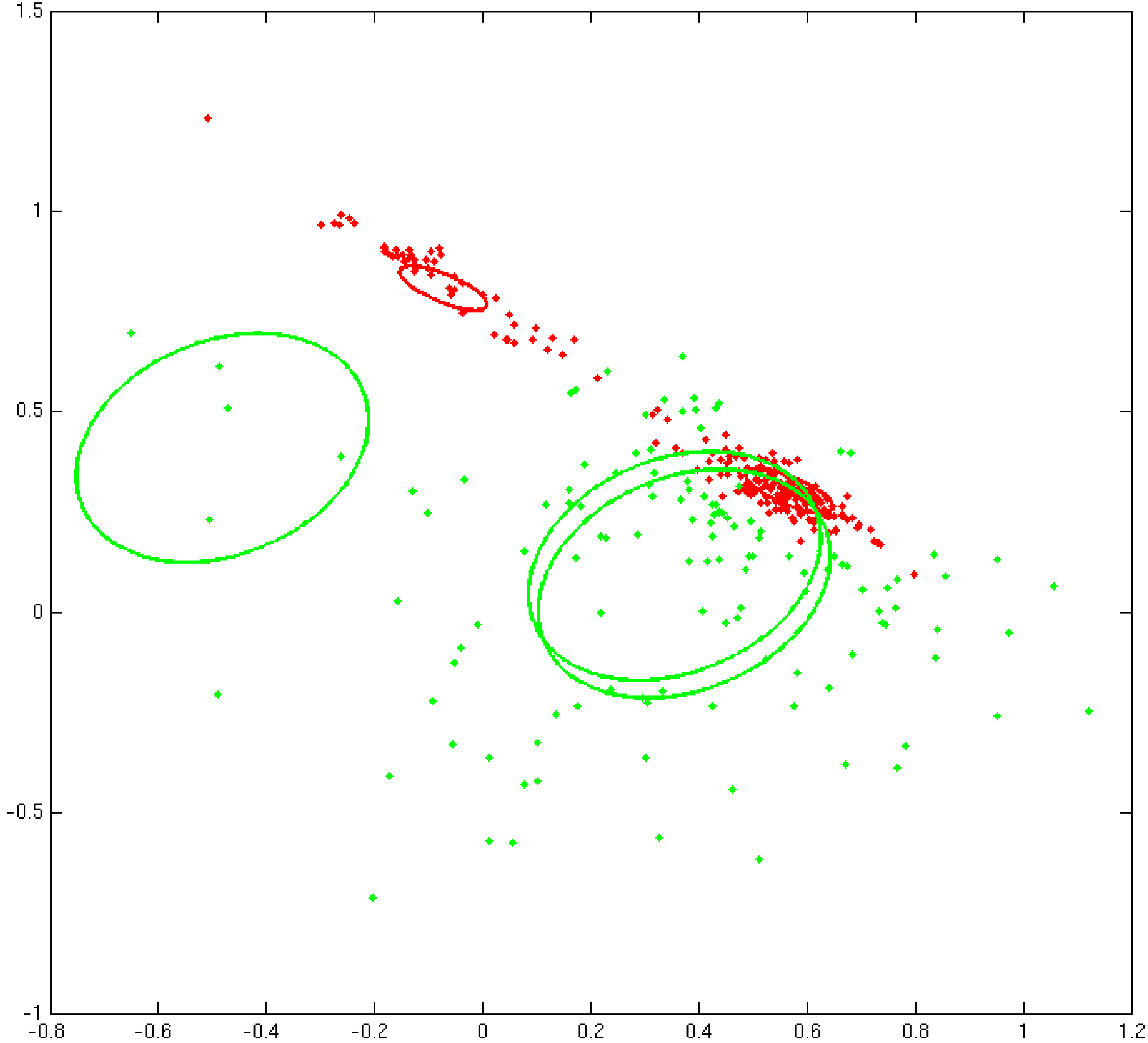
Iteration 6, after CG



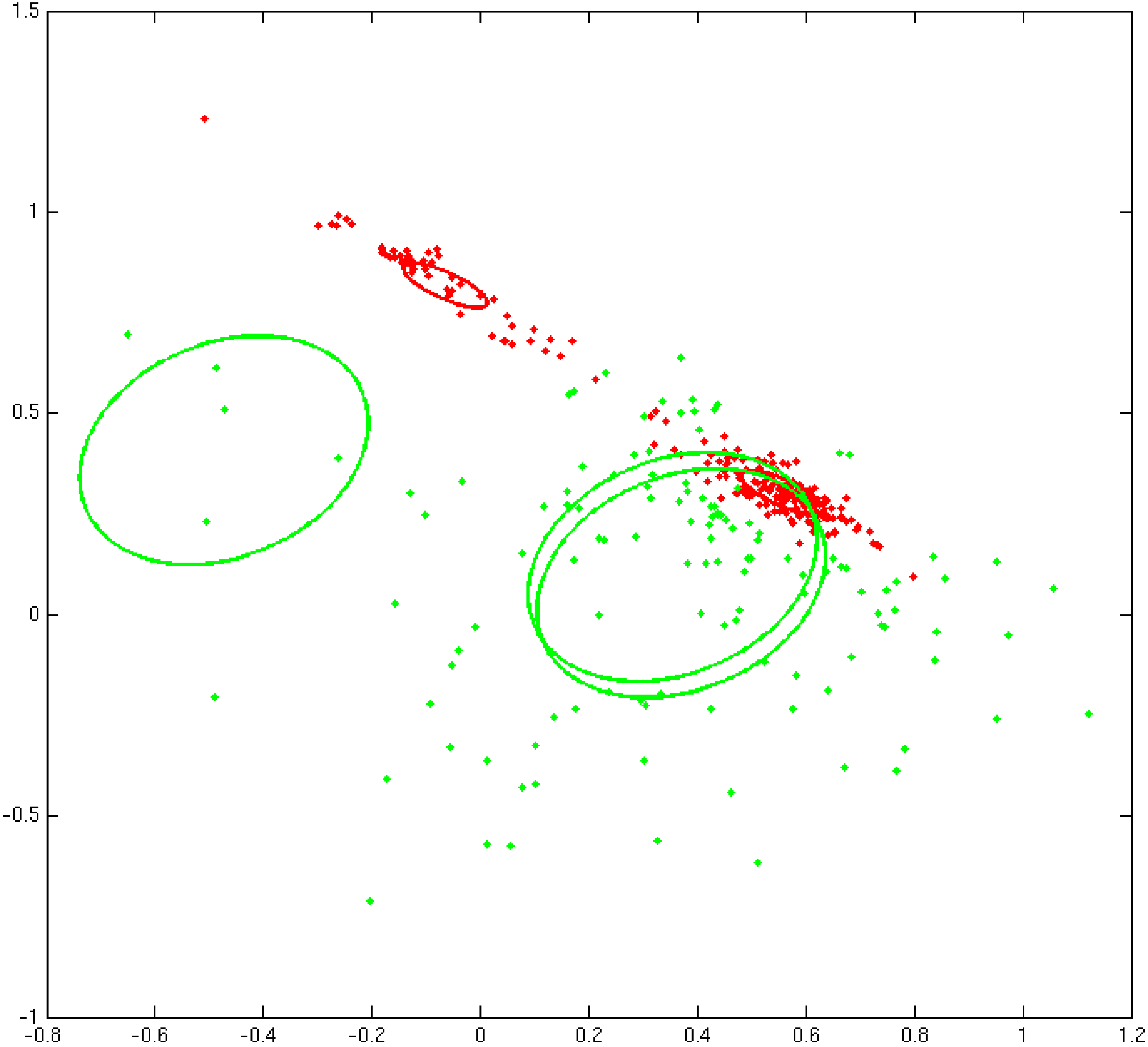
Iteration 7, after EM



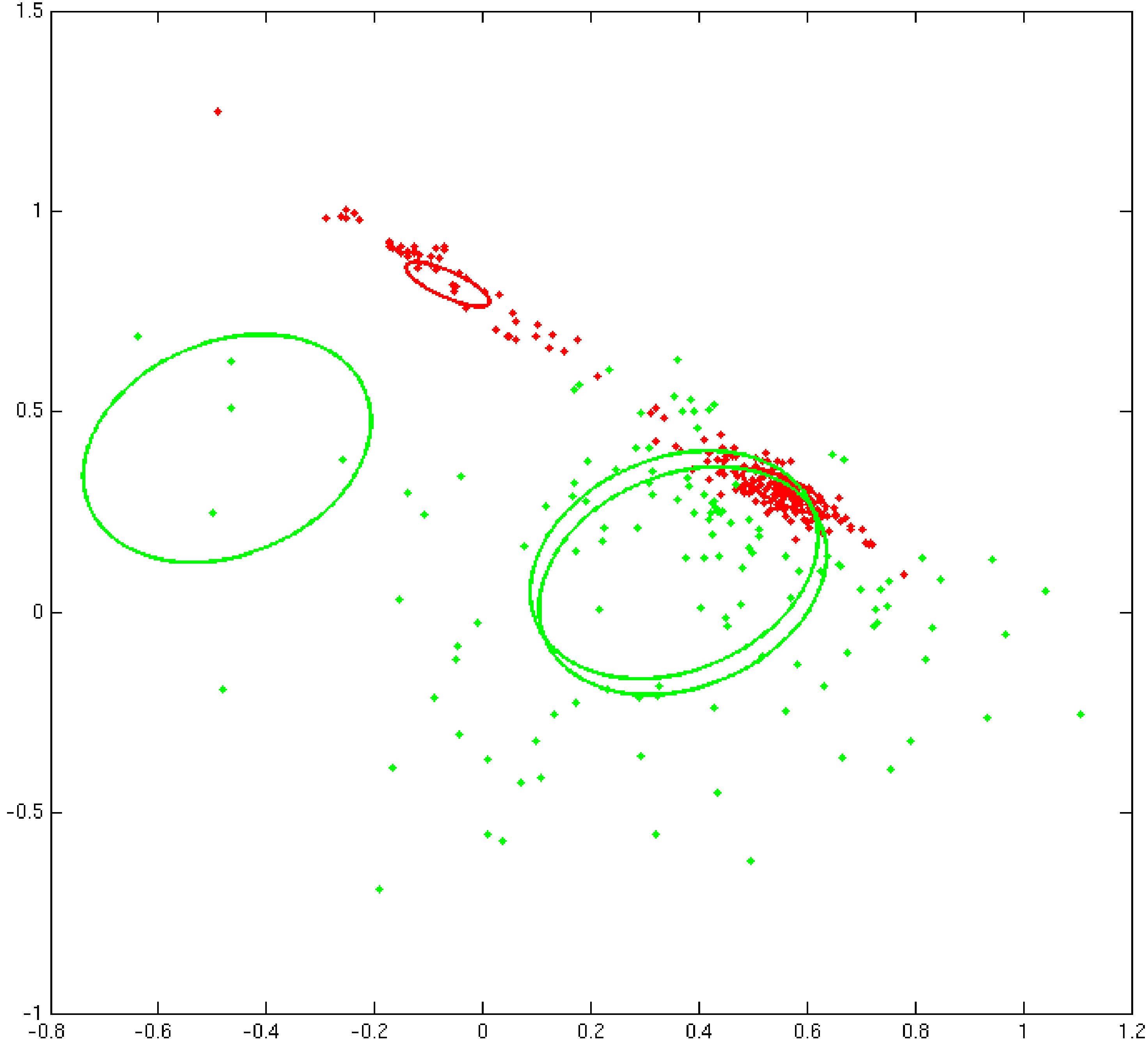
Iteration 7, after CG



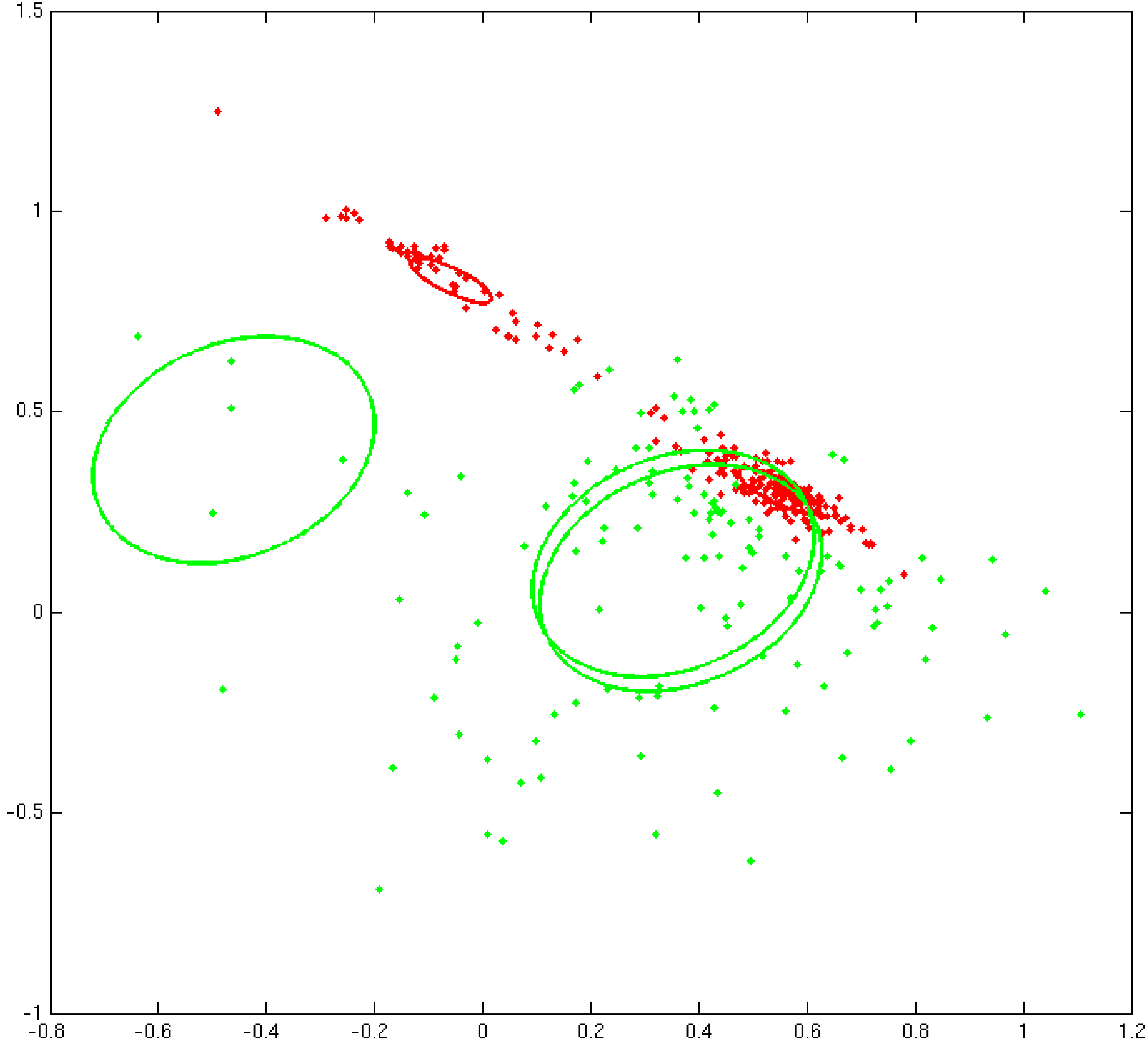
Iteration 8, after EM



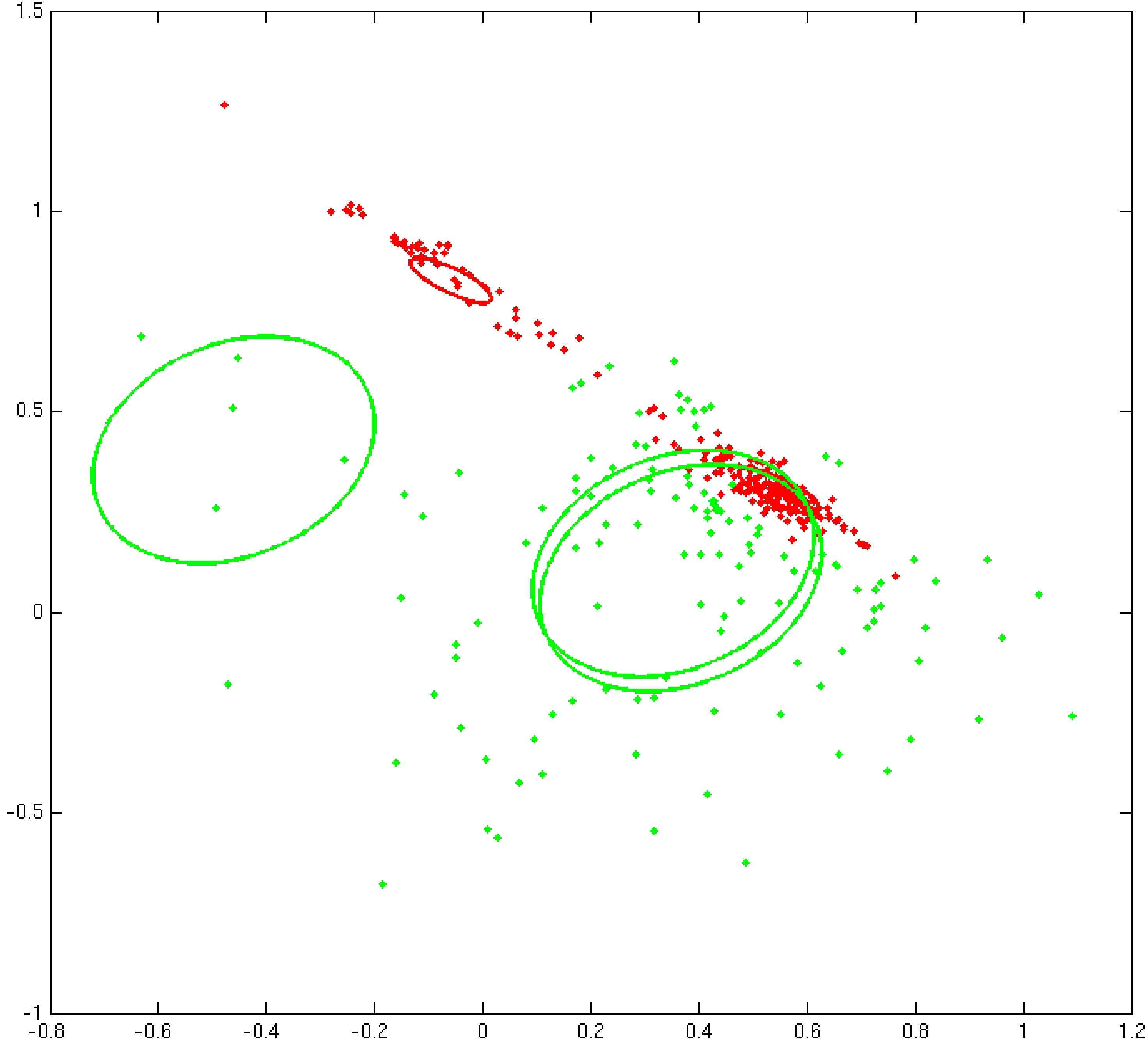
Iteration 8, after CG



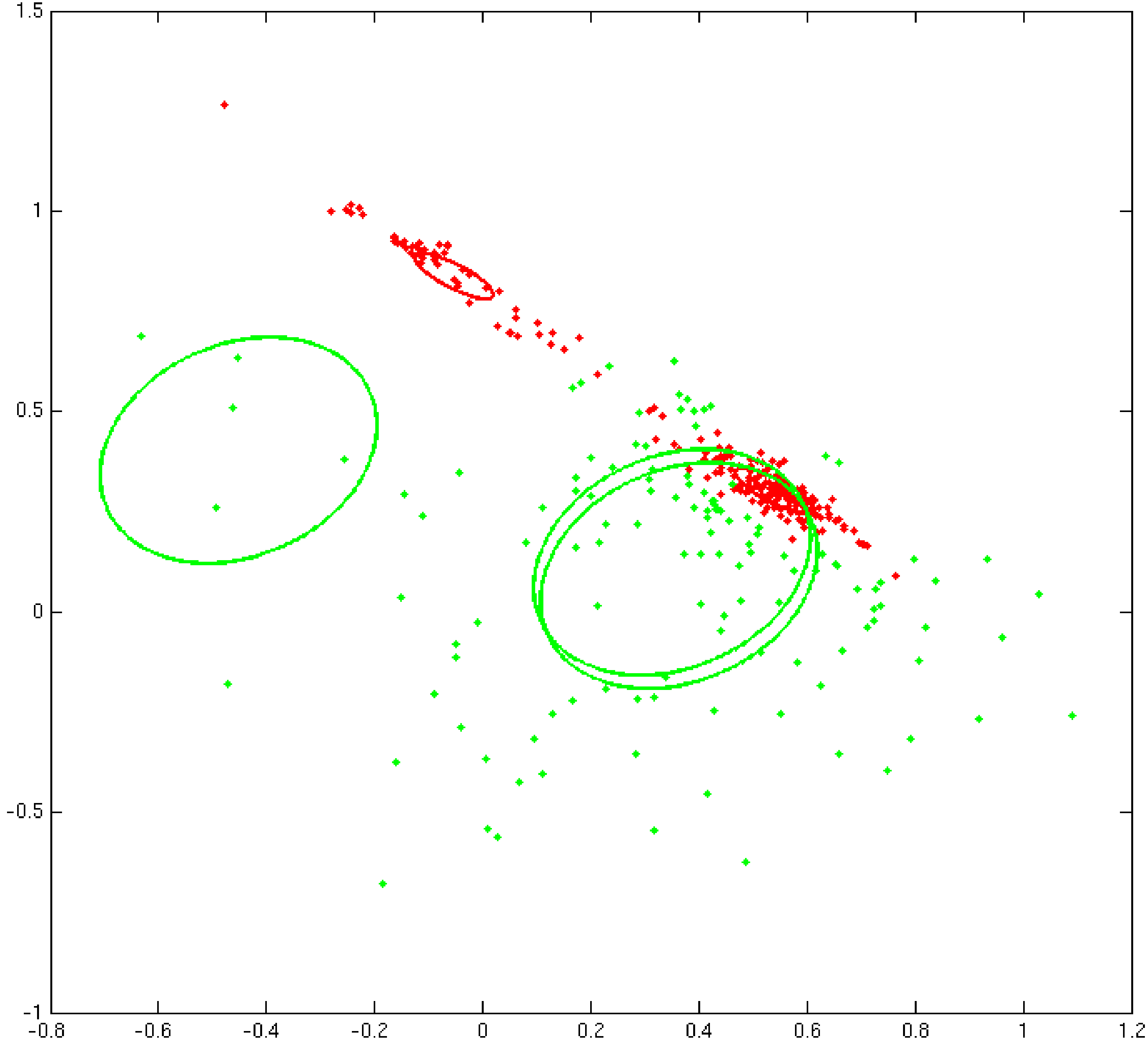
Iteration 9, after EM



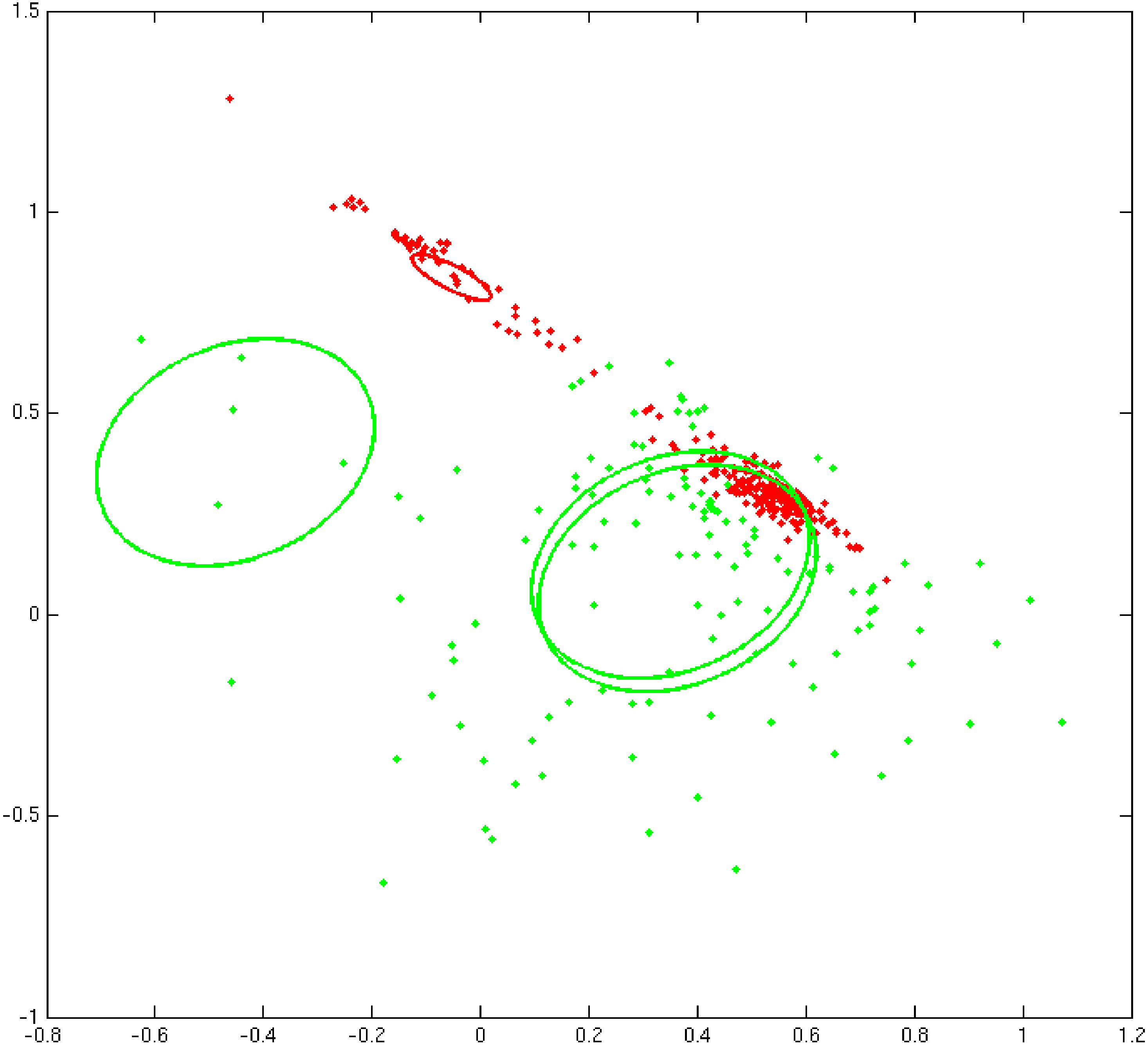
Iteration 9, after CG



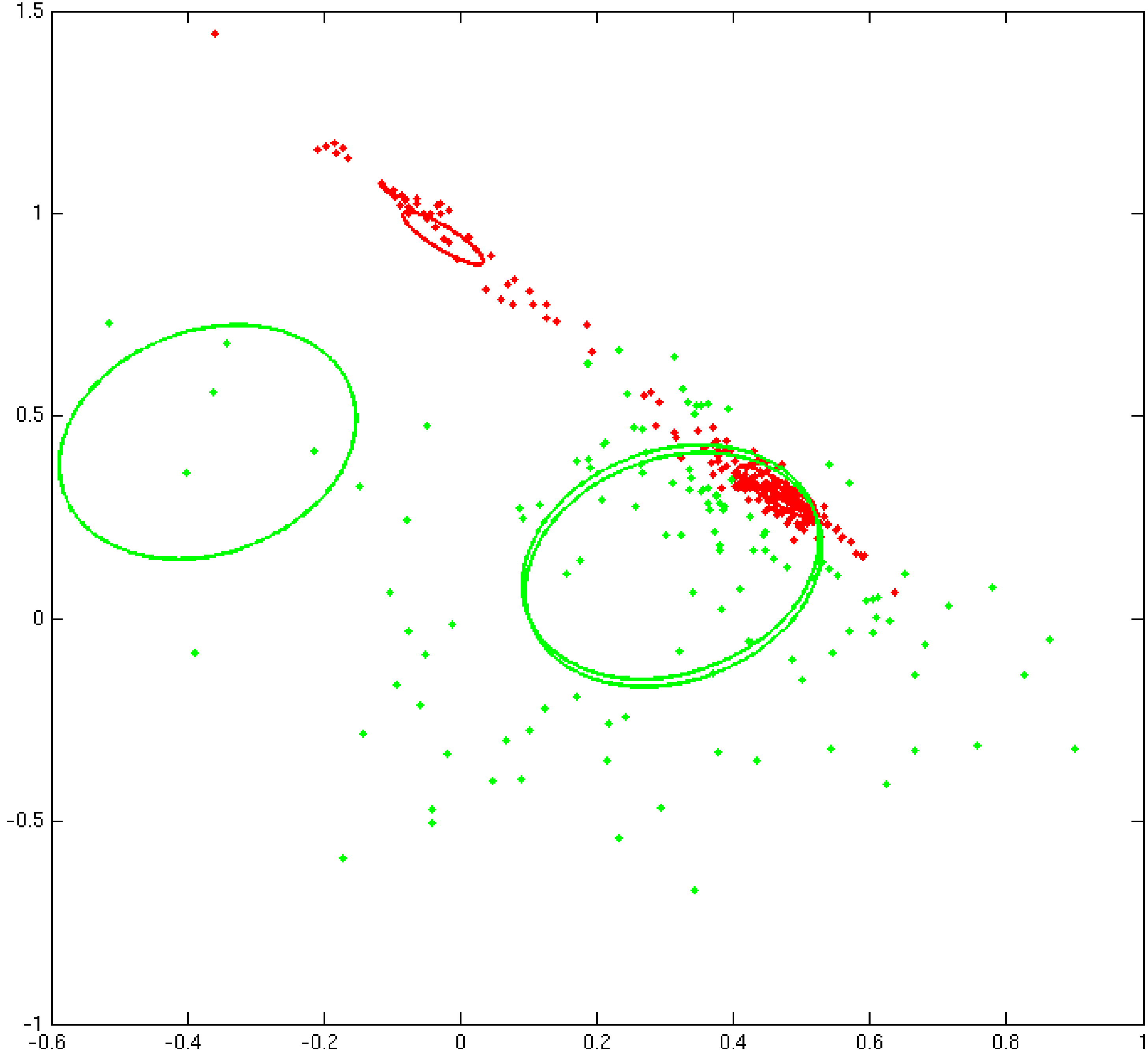
Iteration 10, after EM



Iteration 10, after CG



Iteration 19, after CG



3. Optimization

In the hybrid optimization, the mixture parameters do not change during optimization of the $\boldsymbol{A}$ matrix.

We can make the centers change:
reparameterize $\boldsymbol{\mu}_{c,k} = \boldsymbol{A}\boldsymbol{\mu}'_{c,k}$

Causes only small changes to the gradient and EM step.

4. Properties

- Gradient computation and EM step are both O(N)

4. Properties

- Gradient computation and EM step are both O(N)

- Finds a subspace.

- Metric within the subspace unidentifiable (mixture parameters can compensate for metric changes within the subspace)

4. Properties

- Gradient computation and EM step are both O(N)

- Finds a subspace.

- Metric within the subspace unidentifiable (mixture parameters can compensate for metric changes within the subspace)

- Metric within the subspace can be found by various methods.

5. Experiments

- Four benchmark data sets from UCI Machine Learning Repository (Wine, Balance, Ionosphere, Iris)

5. Experiments

- Four benchmark data sets from UCI Machine Learning Repository (Wine, Balance, Ionosphere, Iris)
- 30 divisions of data into training and test sets

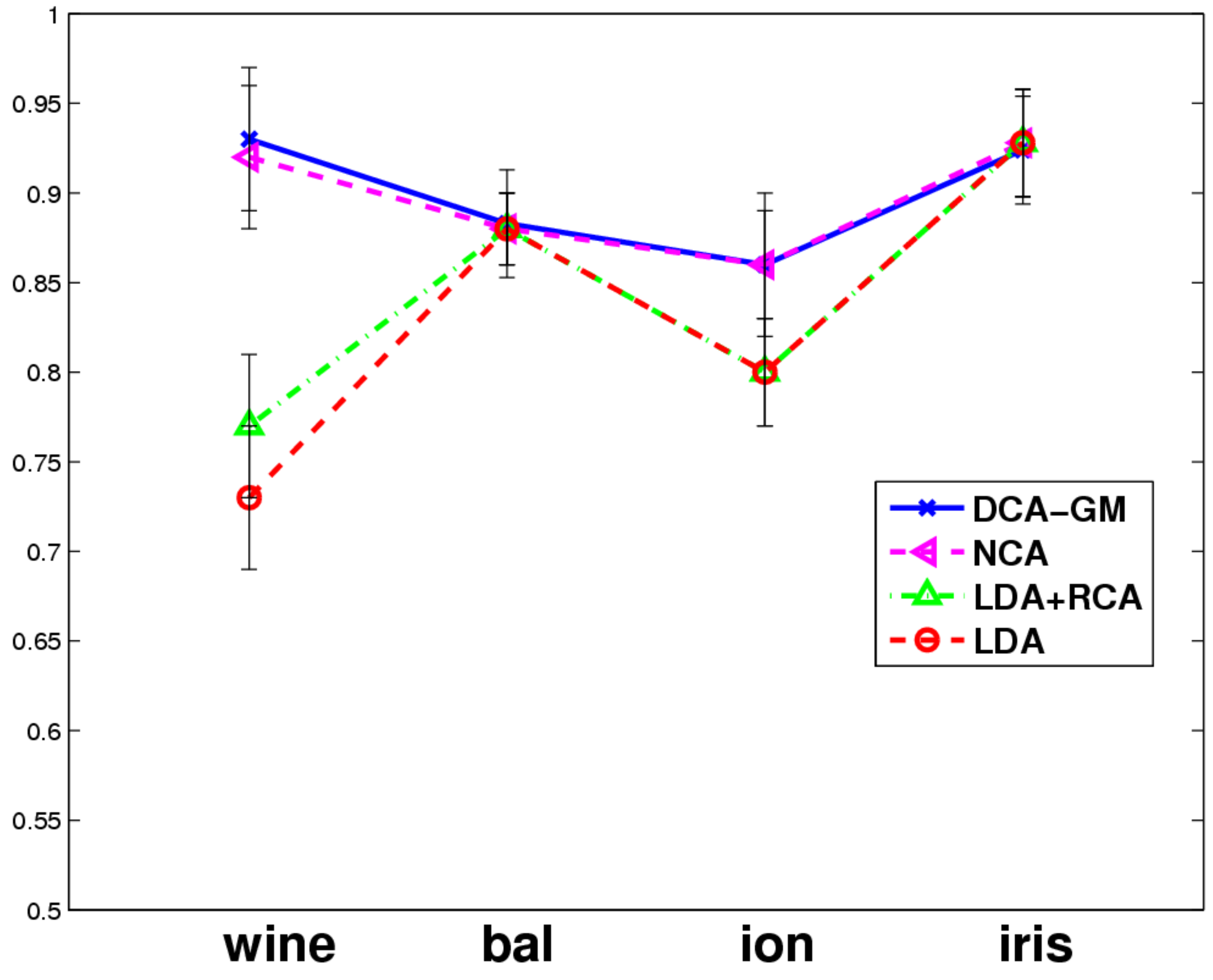
5. Experiments

- Four benchmark data sets from UCI Machine Learning Repository (Wine, Balance, Ionosphere, Iris)
- 30 divisions of data into training and test sets
- Performance measured by test-set accuracy of 1-NN classification

5. Experiments

- Four benchmark data sets from UCI Machine Learning Repository (Wine, Balance, Ionosphere, Iris)
- 30 divisions of data into training and test sets
- Performance measured by test-set accuracy of 1-NN classification
- 4 comparison methods:
  - LDA
  - LDA+RCA
  - NCA
  - DCA-GM, 3 Gaussians per class

5. Experiments



- DCA-GM is comparable to NCA
- For these small data sets both methods run fast

6. Conclusions

- Method for discriminative component analysis
- Optimizes a subspace for a Gaussian mixture model
- O(N) computation
- Works equally well as NCA

6. Conclusions

Web links:

www.cis.hut.fi/projects/mi/
www.eng.biu.ac.il/~goldbej/