

Global Likelihood Optimization via the Cross-Entropy Method with an Application to Mixture Models

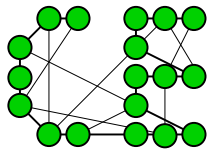
Zdravko Botev[◇] and Dirk P. Kroese

Department of Mathematics

University of Queensland

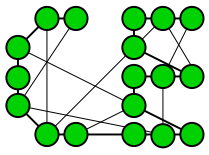
Brisbane, AUSTRALIA

[◇]Supported by the ARC Centre of Excellence: Mathematics and Statistics of
Complex Systems (MASCOS)



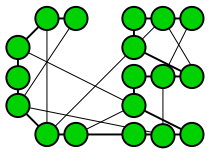
Contents

1. Introduction
2. Cluster Analysis via the Cross-Entropy Method
3. Numerical Experiments
4. Conclusions



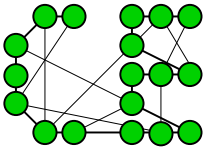
Introduction

- Statistical Analysis often involves **global likelihood maximization**.



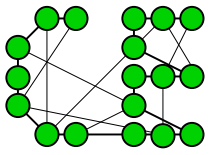
Introduction

- Statistical Analysis often involves **global likelihood maximization**.
 - **Likelihood**: probability (density) of the observed data.



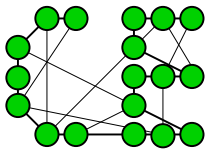
Introduction

- Statistical Analysis often involves **global likelihood maximization**.
 - **Likelihood**: probability (density) of the observed data.
- Often the likelihood function is highly **multi-extremal**.



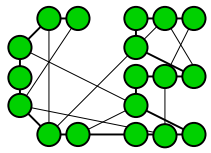
Introduction

- Statistical Analysis often involves **global likelihood maximization**.
 - **Likelihood**: probability (density) of the observed data.
- Often the likelihood function is highly **multi-extremal**.
- Significant **challenge** to standard search procedures.



Introduction

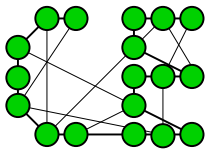
- Statistical Analysis often involves **global likelihood maximization**.
 - **Likelihood**: probability (density) of the observed data.
- Often the likelihood function is highly **multi-extremal**.
- Significant **challenge** to standard search procedures.
- Different approach: **CE method**.



Introduction (cont.)

The **purpose** of this talk is to

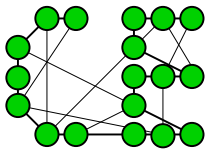
- explain how the **CE method** can be employed as a global likelihood optimization procedure for mixture models in **cluster analysis**,



Introduction (cont.)

The **purpose** of this talk is to

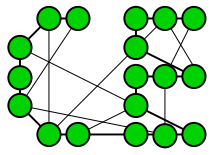
- explain how the **CE method** can be employed as a global likelihood optimization procedure for mixture models in **cluster analysis**,
- introduce a useful modification of CE called the **injection method**,



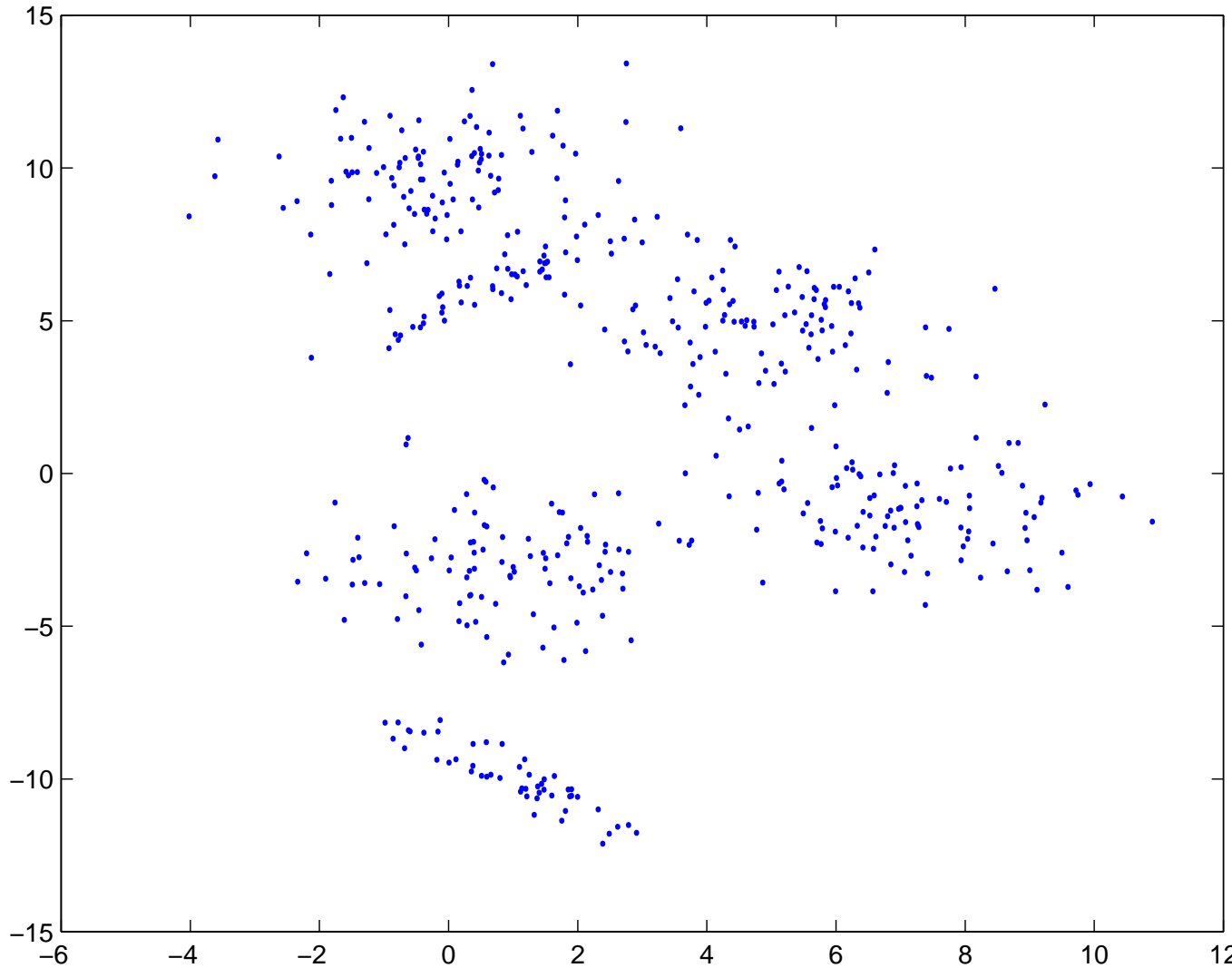
Introduction (cont.)

The **purpose** of this talk is to

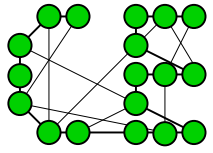
- explain how the **CE method** can be employed as a global likelihood optimization procedure for mixture models in **cluster analysis**,
- introduce a useful modification of CE called the **injection method**,
- compare the CE approach with the classical **EM approach**, for mixture models in cluster analysis.



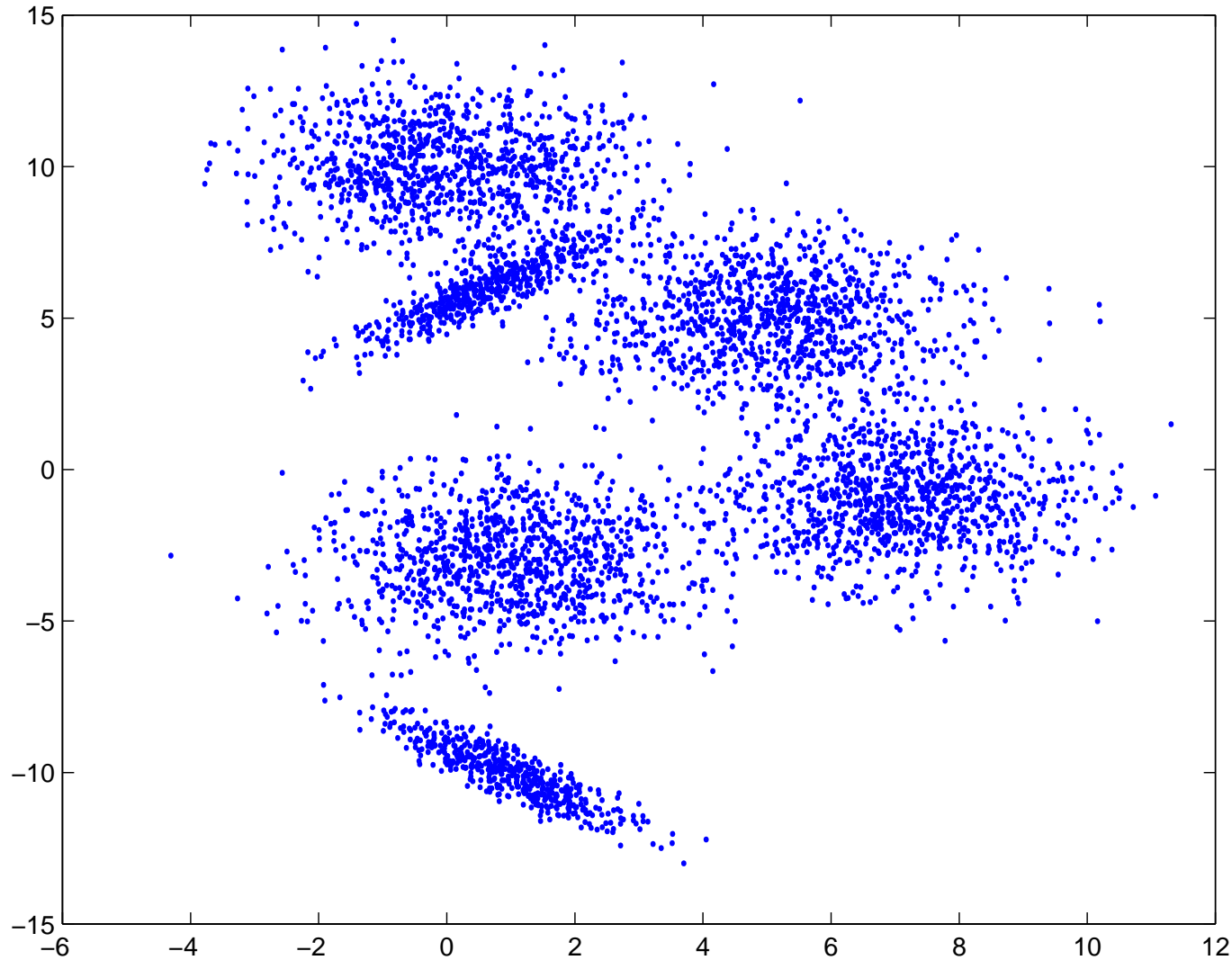
Cluster Analysis



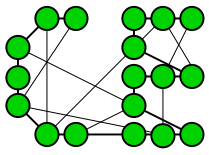
500 points



Cluster Analysis

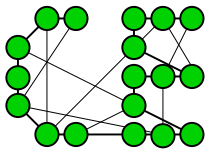


5000 points



Cluster Analysis (cont.)

- Data are assumed to come from a **mixture** of (usually) **Gaussian** densities.
- The objective is to **estimate the parameters** by maximizing the **likelihood function**.
- Traditional methods
 - Local search methods (e.g. **gradient-based**)
 - Simplex methods (**Nelder-Mead**)
 - **EM algorithm**
 - Global search techniques (e.g., **genetic algorithms**)



Cluster Analysis

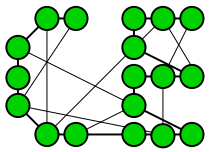
Data: points $\mathcal{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ in \mathbb{R}^d . Assume data are outcomes of i.i.d. random vectors $\mathbf{Y}_1, \dots, \mathbf{Y}_n$, with **mixture density**

$$f(\mathbf{y}) = \sum_{c=1}^k w_c f_c(\mathbf{y}; \boldsymbol{\eta}_c),$$

with unknown **weights** $\mathbf{w} = (w_1, \dots, w_k)$ and distributional **parameters** $\boldsymbol{\eta} = (\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_k)$.

Standard example: each density $f_c \sim \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$.

Objective: estimate $\boldsymbol{\theta} = (\mathbf{w}, \boldsymbol{\eta})$ from the data \mathcal{Y} .

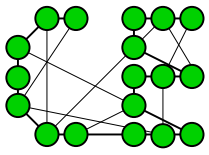


Maximum Likelihood Estimation

A fundamental approach to estimating the parameter $\theta = (w, \eta)$ from the data \mathcal{Y} is to choose the estimate such that the likelihood function

$$\mathcal{L}(\theta; \mathcal{Y}) := \prod_{i=1}^n f(\mathbf{y}_i; \theta)$$

(or, equivalently, its logarithm) is maximized.



Clustering via CE

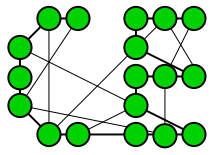
As an alternative to the EM algorithm we consider the CE approach, where we view the clustering problem as a

continuous multi-extremal optimization problem with constraints.

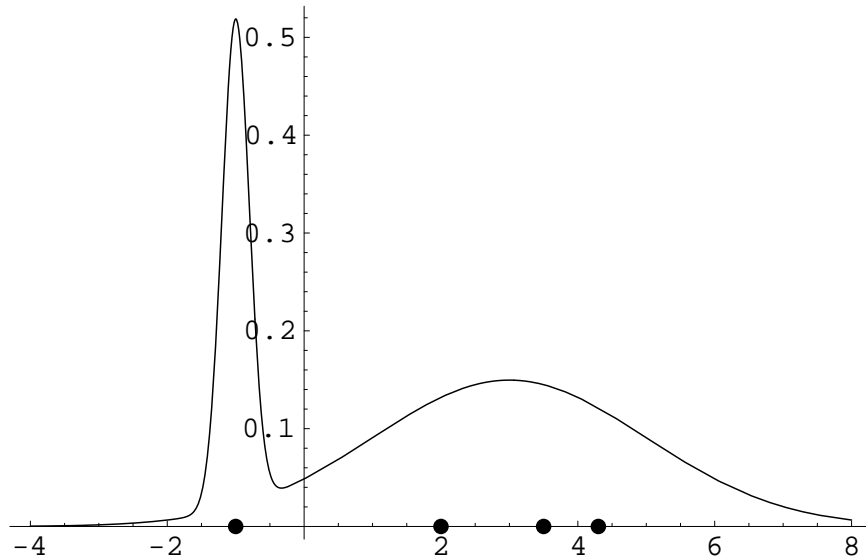
Specifically, we wish to maximize the log-likelihood function

$$\log \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^n \log f(\mathbf{y}_i; \boldsymbol{\theta})$$

over the set Ω of all possible $\boldsymbol{\theta}$.

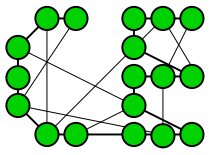


Spurious Clusters



Choosing “**degenerate**” clusters, one can make the value of the (log-)likelihood **infinite**.

Restrict the parameter set such that spurious clusters are not allowed.



CE Method

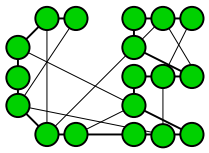
The CE method can be used to solve: $\max_{\mathbf{x} \in \mathcal{X}} S(\mathbf{x})$. See:

<http://www.cemethod.org>

The **basic procedure** of the CE method is to iteratively

- (a) **generate** random samples in \mathcal{X} according to a specified sampling distribution, followed by
- (b) **update** the parameters on the basis of the **best scoring samples**, in order to produce better scoring samples in the next iteration.

The updating rules follow from **cross-entropy minimization** and often have a **simple form**.



A Constrained Optimization Problem

Consider k two-dimensional clusters. Maximize

$$S(\boldsymbol{\theta}) = \log \mathcal{L}(\boldsymbol{\theta}), \quad \boldsymbol{\theta} \in \mathbb{R}^{6k-1},$$

such that

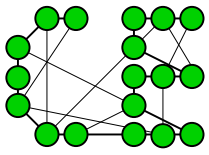
$$\theta_i \in \mathbb{R}, \quad i = 1, \dots, 2k \quad (\text{means})$$

$$\theta_i^{\text{low}} \leq \theta_i, \quad i = 2k + 1, \dots, 4k \quad (\text{std. dev.})$$

$$-\rho_i^{\text{low}} \leq \theta_i \leq \rho_i^{\text{high}}, \quad i = 4k + 1, \dots, 5k \quad (\text{corr. coeff.})$$

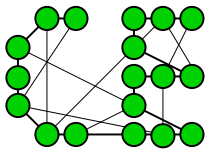
$$0 \leq \theta_i \leq 1, \quad i = 5k + 1, \dots, 6k - 1 \quad (\text{weights})$$

$$\sum_{i=5k+1}^{6k-1} \theta_i \leq 1$$



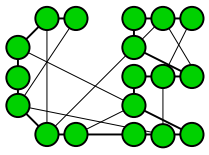
Generating Random θ

- With each parameter θ_i in θ we associate a 1-dimensional Gaussian distribution $N(a_i, b_i^2)$.
- Generate the components of θ **independently**.
- Update $\{(a_i, b_i^2)\}$ via the **sample mean** and **sample variance** of the **elite** samples; i.e., those that give the highest likelihood.
- For θ_i in a constrained region, sample from a **truncated** Gaussian distribution (same updating!)
- Summarize the a_i and b_i^2 into vectors \mathbf{a} and \mathbf{b}^2 .



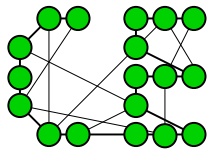
CE Algorithm

1. **Initialize** \mathbf{a}_0 and \mathbf{b}_0^2 . Set $t = 1$ (level counter).
2. **Generate** a sample $\Theta_1, \dots, \Theta_N$ from $N(\mathbf{a}_{t-1}, \mathbf{b}_{t-1}^2)$ (or its truncated version) and compute the log-likelihoods.
3. Let $\tilde{\mathbf{a}}_t$ and $\tilde{\mathbf{b}}_t^2$ be the **sample means** and **variances** based on the best N^{elite} samples.
4. **Update** the \mathbf{a} and \mathbf{b}^2 in a **smooth** way, as
$$\mathbf{a}_t = \alpha \tilde{\mathbf{a}}_t + (1 - \alpha) \mathbf{a}_{t-1},$$
$$\mathbf{b}_t^2 = \alpha \tilde{\mathbf{b}}_t^2 + (1 - \alpha) \mathbf{b}_{t-1}^2.$$
5. **Stop** at iteration $t = T$ if some **stopping criterion** is met. Output \mathbf{a}_T . Otherwise increase t by 1 and return to step 2.



Convergence

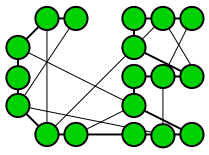
- The algorithm produces a sequence $(\mathbf{a}_0, \mathbf{b}_0^2), (\mathbf{a}_1, \mathbf{b}_1^2), \dots$ that tends to some $(\boldsymbol{\theta}^*, \mathbf{0})$ where $\boldsymbol{\theta}^*$ is the estimate of the global maximum.
- The algorithm is quite robust under the choice of the initial parameters \mathbf{a}_0 and \mathbf{b}_0^2 , provided that the initial variances are chosen large enough. A convenient choice is to let the initial means and variances be equal to the means and variances of the data.



Injection

Convergence to a degenerate distribution may happen too quickly, which would “freeze” the algorithm in a sub-optimal solution. One way to prevent this is to use **Dynamic Smoothing**. Another idea is **Variance Injection**:

1. If at some iteration t the maximum of the variances in \mathbf{b}_t^2 is less than ε , **add** $|S_t^* - S_{t-1}^*| h$, to the variances. (Here S_t^* is the best value in iteration t .)
2. If the number of variance injections **exceeds** some number d , say 5, then stop, otherwise increase t continue.

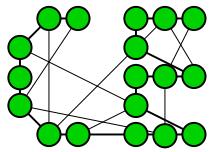


Numerical Experiment

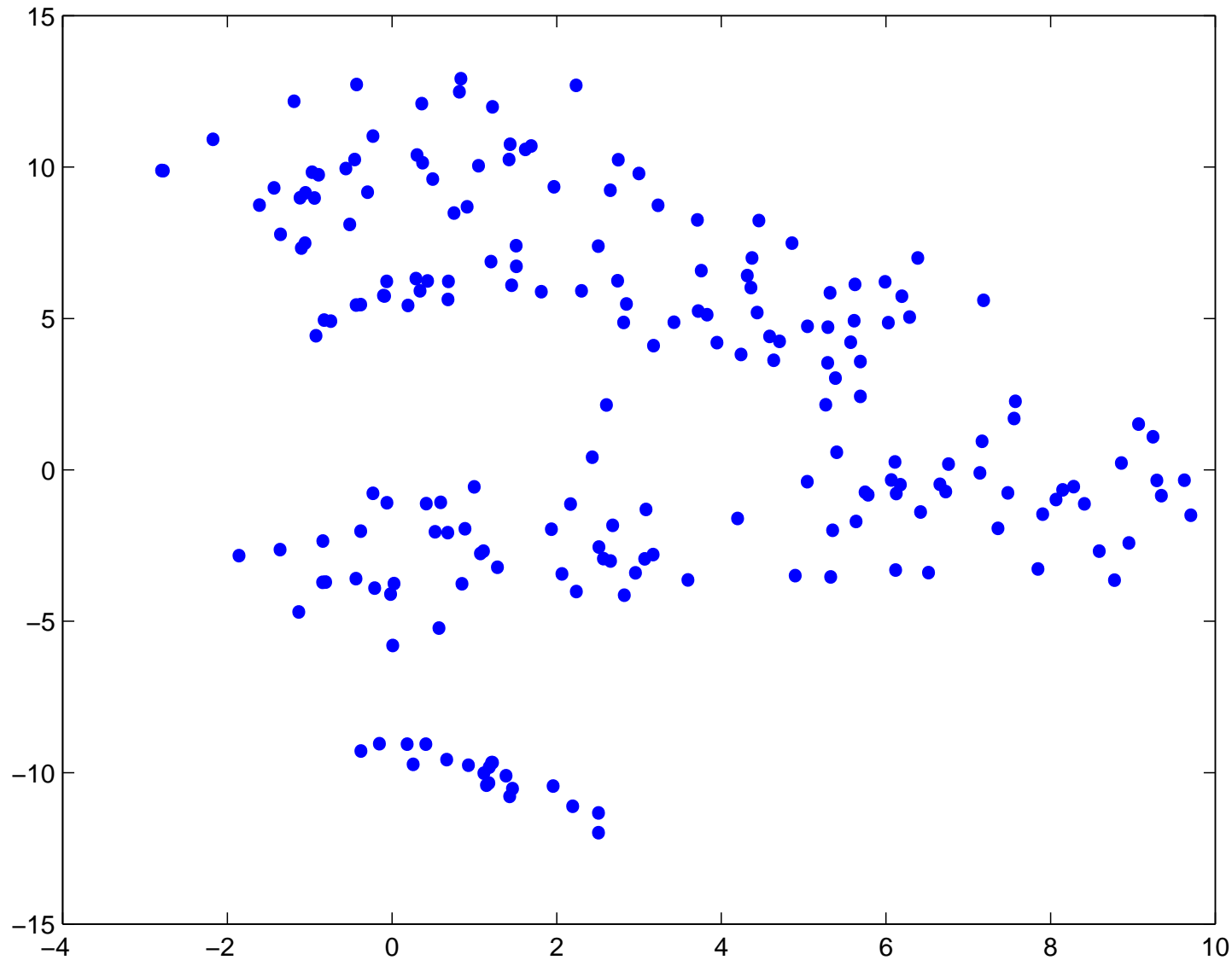
Draw 200 data points from a 6-Gaussian mixture distribution:

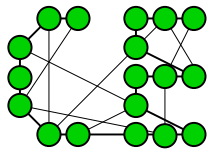
| Cluster | μ | σ^2 | Cov | w |
|---------|---------------|-------------|-------|------|
| 1 | (0.60,6.00) | (1.00,1.00) | 0.90 | 0.10 |
| 2 | (1.00,-10.00) | (1.00,1.00) | -0.90 | 0.10 |
| 3 | (10.00,-1.00) | (2.00,2.00) | 0.00 | 0.20 |
| 4 | (0.00,10.00) | (2.00,2.00) | 0.00 | 0.20 |
| 5 | (1.00,-3.00) | (2.00,2.00) | -0.00 | 0.20 |
| 6 | (-5.00,5.00) | (2.00,2.00) | 0.00 | 0.20 |

Constraints: variances greater or equal to 0.75, and correlation coefficients between -0.95 and 0.95 .



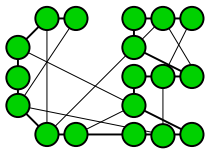
Find 6 Gaussian Clusters





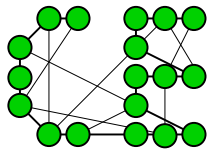
CE Parameters

- Sample size $N = 90$. Elite sample size $N^{\text{elite}} = 12$.
- Smoothing parameters: 0.9 (means) and 0.3 (variances).
- \mathbf{a}_0 : Initial cluster means and variances: sample mean and variance of data. Initial cluster weights: all equal.
- \mathbf{b}_0^2 : large enough to ensure a “uniform” sampling in the first iteration.
- Injection parameters $\varepsilon = 0.01$ and $h = 2$.
- Stopping parameter $d = 5$ (5 injections).

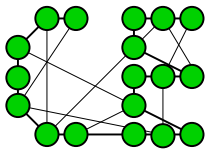


Typical Evolution of CE

| t | S_t^* | $\min_{u \leq t} S_u^*$ | $\max_i b_t^2(i)$ |
|-----|---------|-------------------------|-------------------|
| 20 | 1160.89 | 1142.00 | 35.30 |
| 40 | 1053.52 | 1052.65 | 24.62 |
| 60 | 1013.71 | 1013.38 | 3.01 |
| 80 | 1005.13 | 1005.13 | 0.98 |
| 100 | 1001.60 | 1001.55 | 0.91 |
| 120 | 998.45 | 998.45 | 0.06 |
| 140 | 1028.42 | 997.96 | 0.12 |
| 160 | 1002.02 | 997.96 | 0.05 |
| 180 | 998.81 | 997.96 | 0.03 |
| 200 | 997.00 | 997.00 | 0.02 |
| 220 | 1023.58 | 996.03 | 0.12 |
| 240 | 1000.91 | 996.03 | 0.19 |

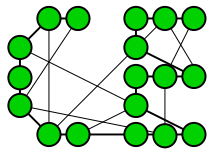


| t | S_t^* | $\min_{u \leq t} S_u^*$ | $\max_i b_t^2(i)$ |
|-----|---------|-------------------------|-------------------|
| 260 | 990.64 | 990.64 | 0.08 |
| 280 | 982.11 | 982.11 | 0.02 |
| 300 | 1013.77 | 981.52 | 0.22 |
| 320 | 988.15 | 981.52 | 0.15 |
| 340 | 981.94 | 981.52 | 0.05 |
| 360 | 980.53 | 980.34 | 0.01 |
| 380 | 994.29 | 980.34 | 0.08 |
| 400 | 983.26 | 980.34 | 0.05 |
| 420 | 980.73 | 980.34 | 0.02 |
| 440 | 1015.13 | 980.34 | 0.21 |
| 460 | 985.78 | 980.34 | 0.13 |
| 480 | 981.75 | 980.34 | 0.04 |
| 500 | 980.43 | 980.34 | 0.01 |



Comparison with EM

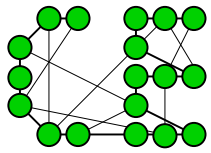
- Run **multiple** copies of the EM algorithm.
- **Total time** for all EM runs is **no less** than the time taken by the CE algorithm.
- Solutions of the EM algorithm that do not satisfy the constraints are **discarded**.



Results

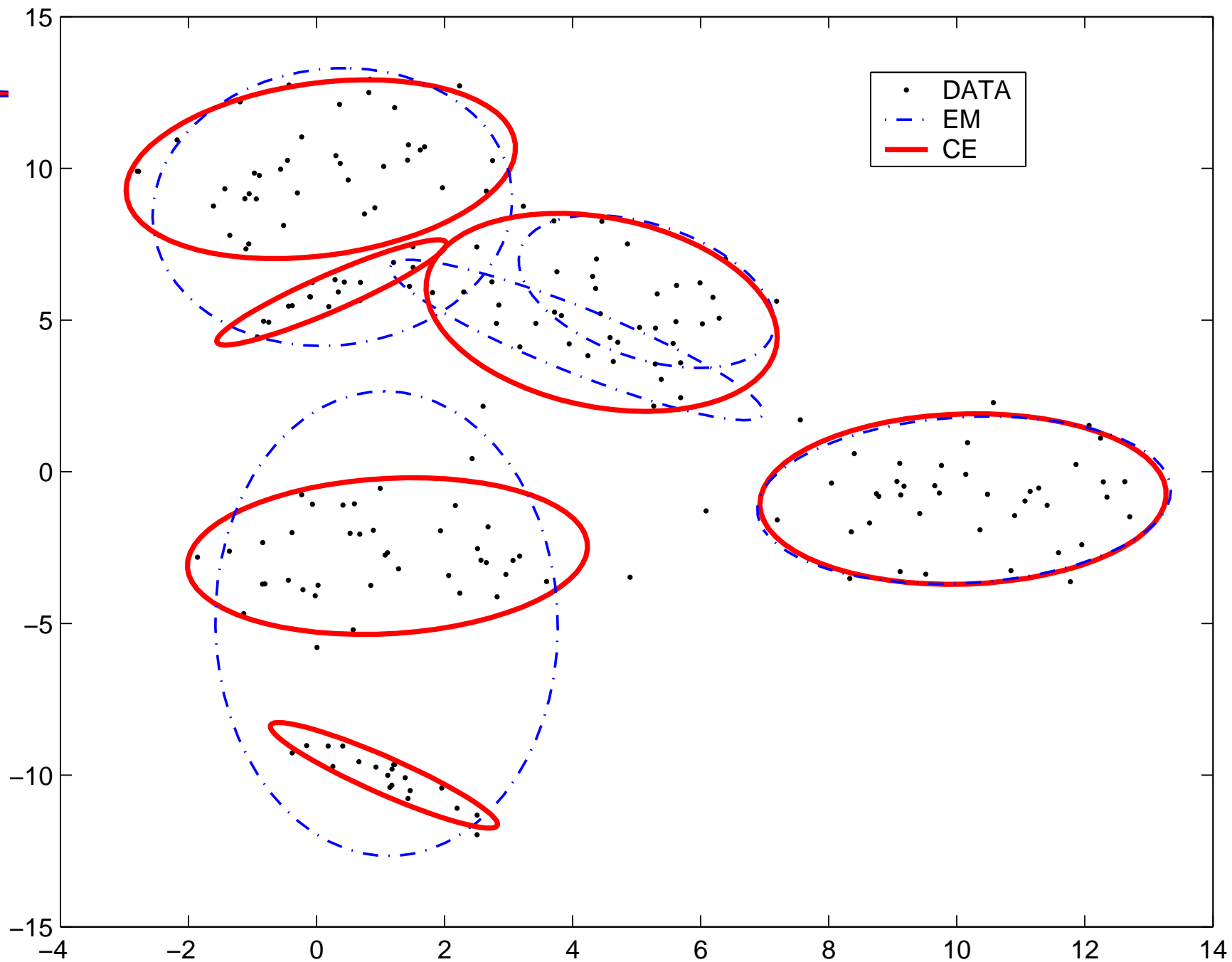
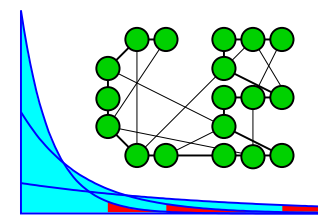
In 10 repetitions CE found the correct clusters 4 out of 10 times, but EM failed.

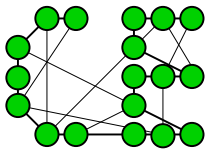
| CE | EM | time | CE | EM | time |
|--------|---------|------|---------|---------|------|
| 980.33 | 1048.62 | 38 | 997.98 | 1052.99 | 34 |
| 982.75 | 1052.99 | 32 | 994.19 | 1047.83 | 40 |
| 998.01 | 1041.86 | 34 | 1004.76 | 1057.18 | 39 |
| 980.36 | 1047.83 | 31 | 994.08 | 1052.99 | 35 |
| 980.32 | 1047.83 | 36 | 980.62 | 1052.99 | 34 |



Example (experiment 1):

| | μ | σ^2 | Cov | w |
|-------|-----------------|---------------|-------|------|
| | (4.07, 4.34) | (2.12, 1.75) | -1.73 | 0.09 |
| | (5.16 5.94) | (1.00, 1.58) | -0.51 | 0.10 |
| EM | (10.11, -0.94) | (2.61, 1.90) | 0.26 | 0.20 |
| | (1.09, -5.01) | (1.79 14.678) | -0.06 | 0.31 |
| | (0.25, 8.73) | (1.97, 5.25) | 0.20 | 0.29 |
| | (0, 0) | (1 , 1) | 0 | 0 |
| <hr/> | | | | |
| | (0.24, 5.91) | (0.81,0.75) | 0.72 | 0.08 |
| | (1.05 , -10.01) | (0.79,0.76) | -0.71 | 0.11 |
| CE | (10.10, -0.90) | (2.52,1.98) | 0.14 | 0.21 |
| | (0.07, 9.97) | (2.31,2.17) | 0.53 | 0.19 |
| | (1.11, -2.78) | (2.44,1.67) | 0.25 | 0.20 |
| | (-4.45 ,5.25) | (1.88,2.66) | -0.55 | 0.21 |





Conclusions and Future Research

- The CE method is not sensitive to the initial conditions, in contrast to EM.
- CE can more easily handle constraints (spurious clusters!)
- EM is faster.
- CE is more consistent, especially for small data sets. .
- CE can handle non-Gaussian clusters
- Comparison with other techniques, e.g., KM, LVQ.
- Modifications: different sampling distributions, componentwise updating, MaxEnt, multi-agent optimization, gradual feeding.