Machine Learning for Sensory Signals

Gaussian and Mixture Gaussian Models

02-03-2017

Established
1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is defined as

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{k=1}^{K} \alpha_k p(\mathbf{x}|\boldsymbol{\theta}_k)$$

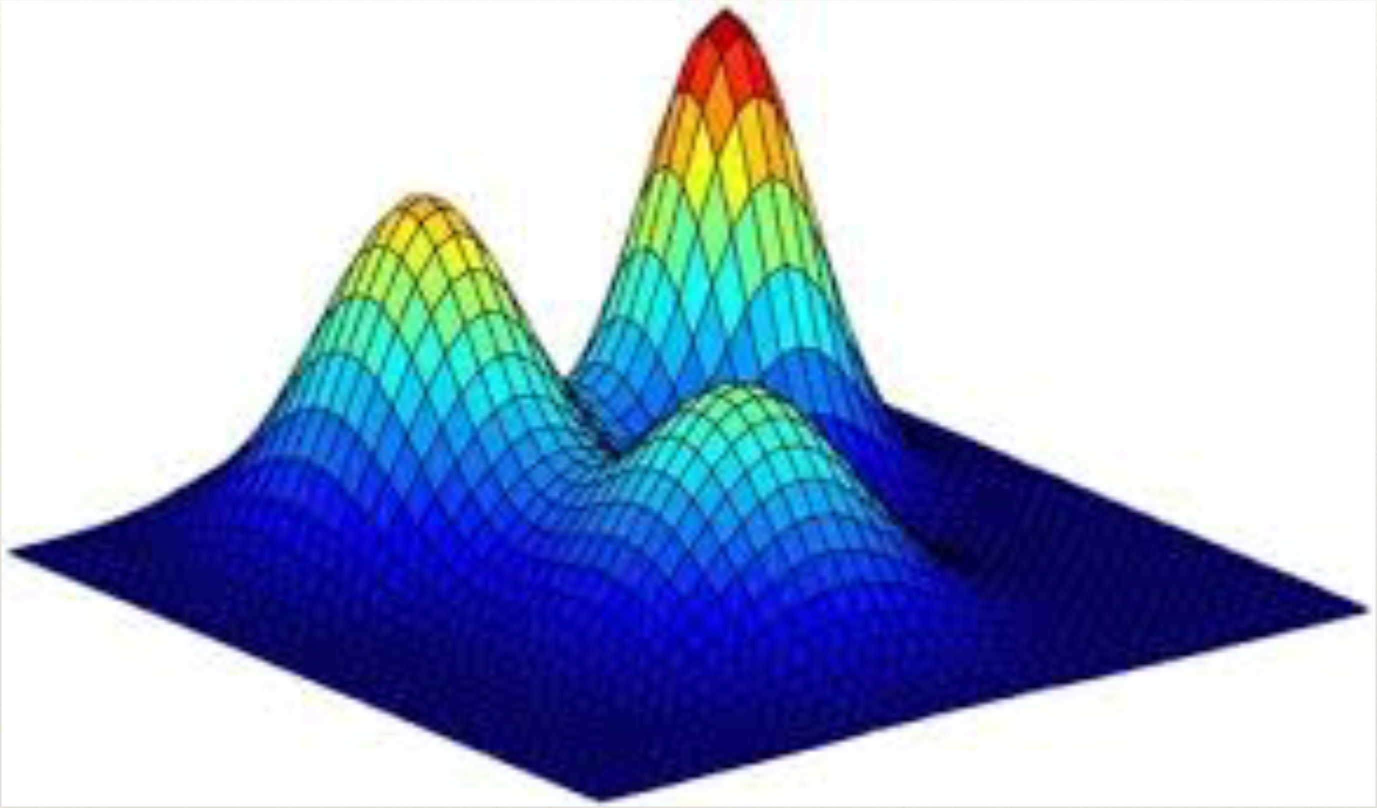
$$p(\mathbf{x}|\boldsymbol{\theta}_k) = \frac{1}{\sqrt{(2\pi)^D|\mathbf{\Sigma}_k|}} exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^*\mathbf{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) \right\}$$

The weighting coefficients have the property

$$\sum_{k=1}^{K} \alpha_k = 1$$

Gaussian Mixture Modeling

- Properties of GMM
  - Can model multi-modal data.
  - Identify data clusters.
  - Can model arbitrarily complex data distributions



The set of parameters for the model are

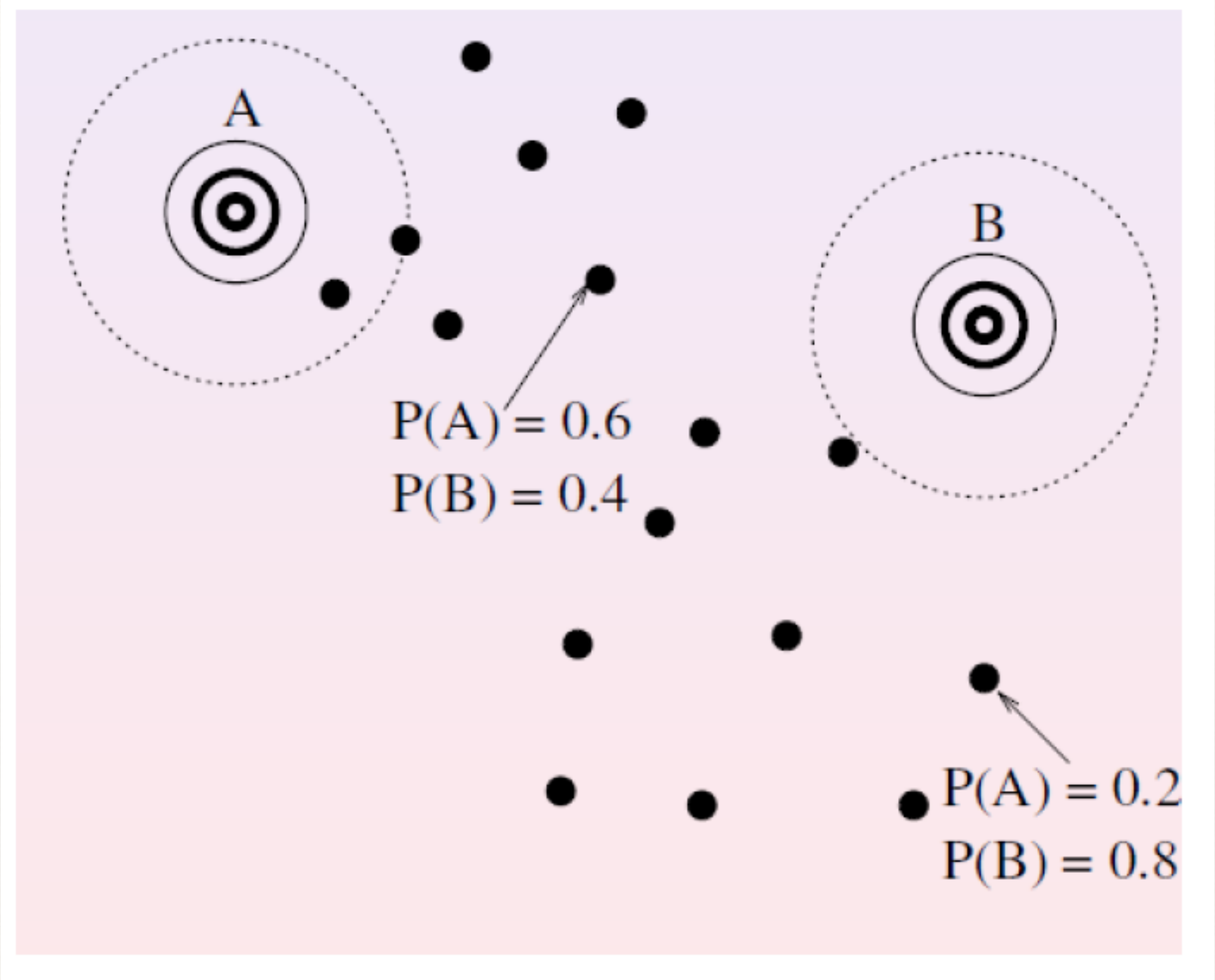
$$\mathbf{\Theta}_k = \{\alpha_k, \boldsymbol{\theta}_k\}_{k=1}^{K} \qquad \boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}$$

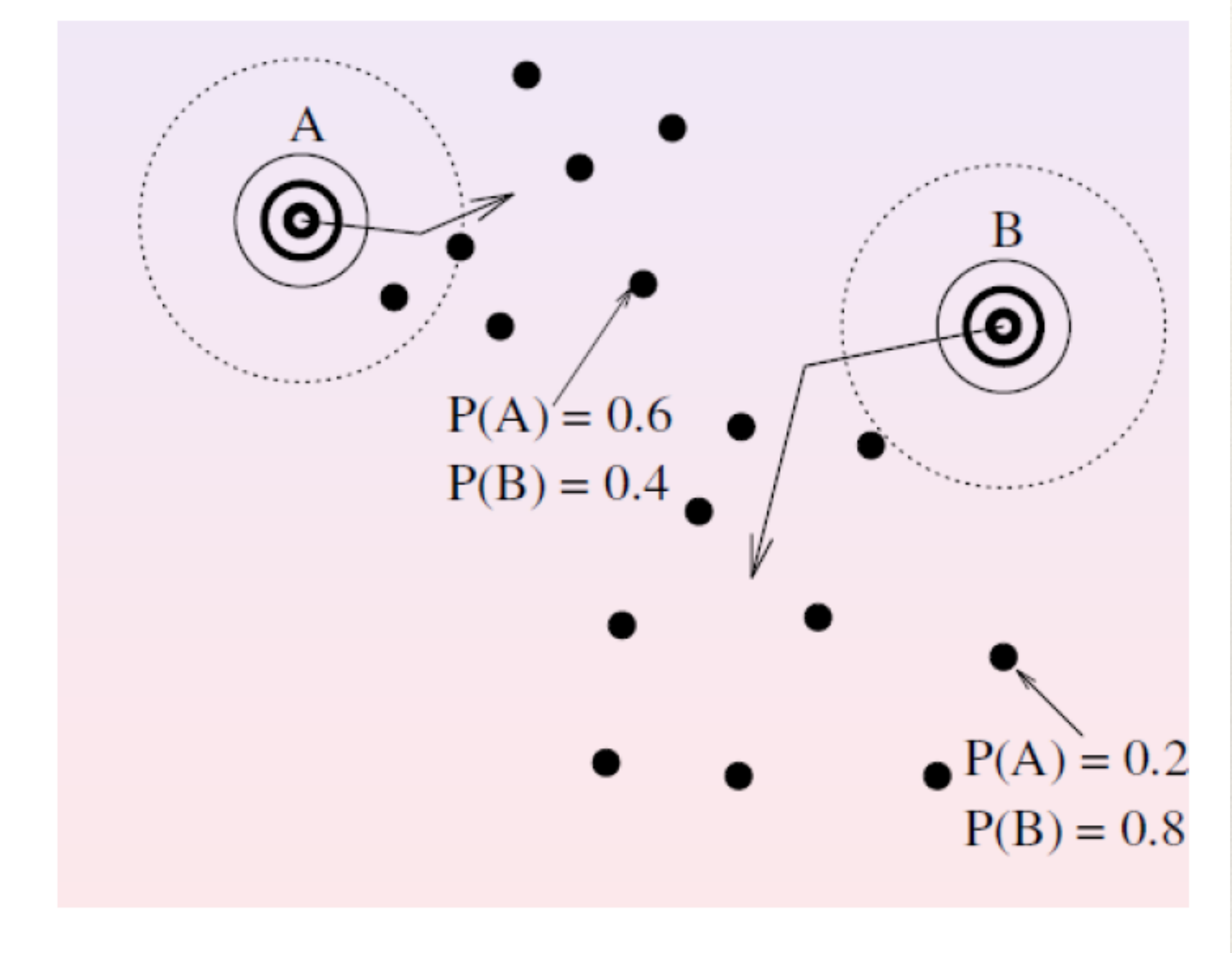
- The log-likelihood function over the entire data in this case will have a logarithm of a summation

$$\log L(\mathbf{\Theta}) = \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} \alpha_k p(\mathbf{x}_i|\boldsymbol{\theta}_k)\right)$$

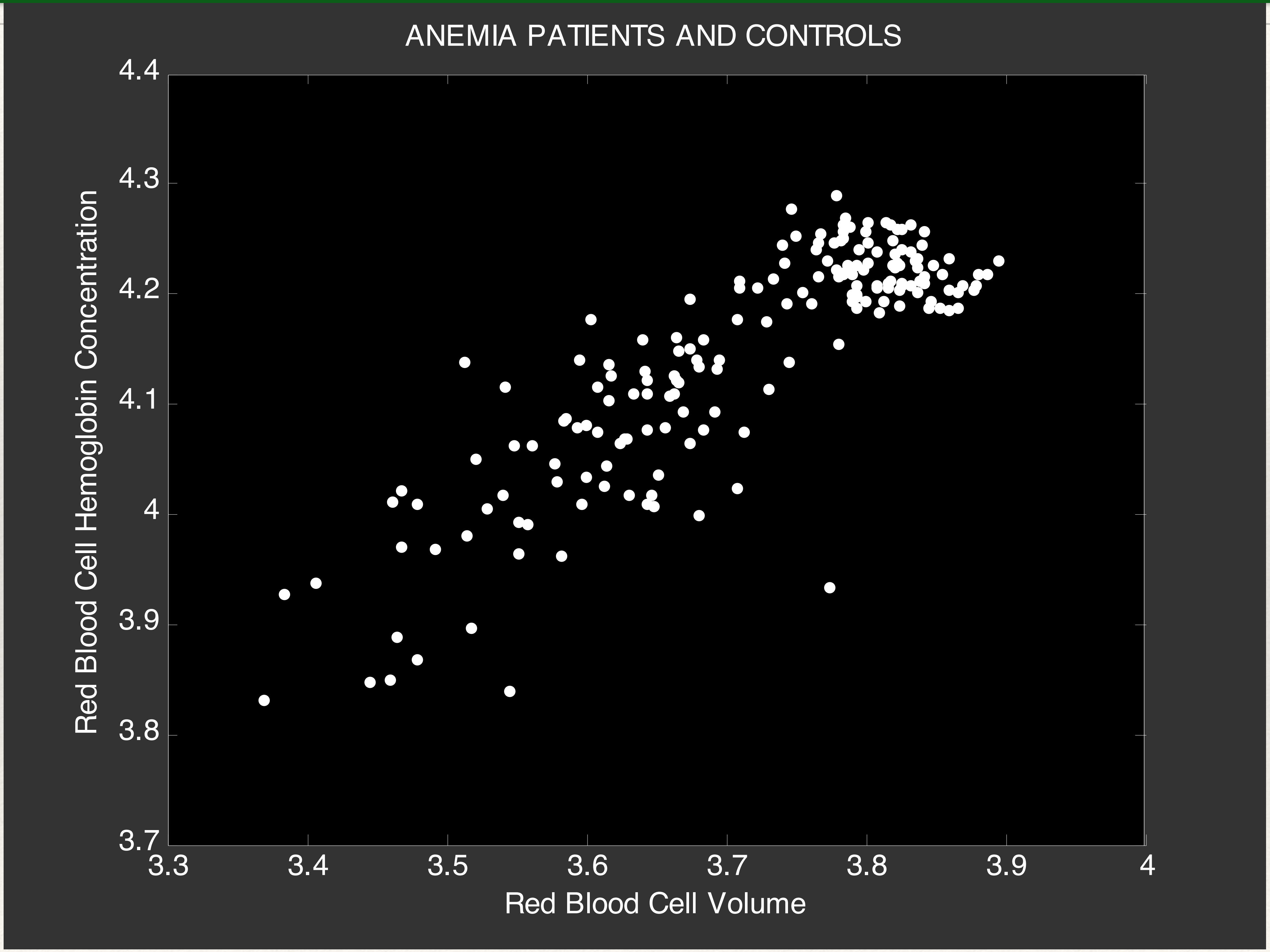
- Solving for the optimal parameters using MLE for GMM is not straight forward.
- Resort to the Expectation Maximization (EM) algorithm

MLE for GMM

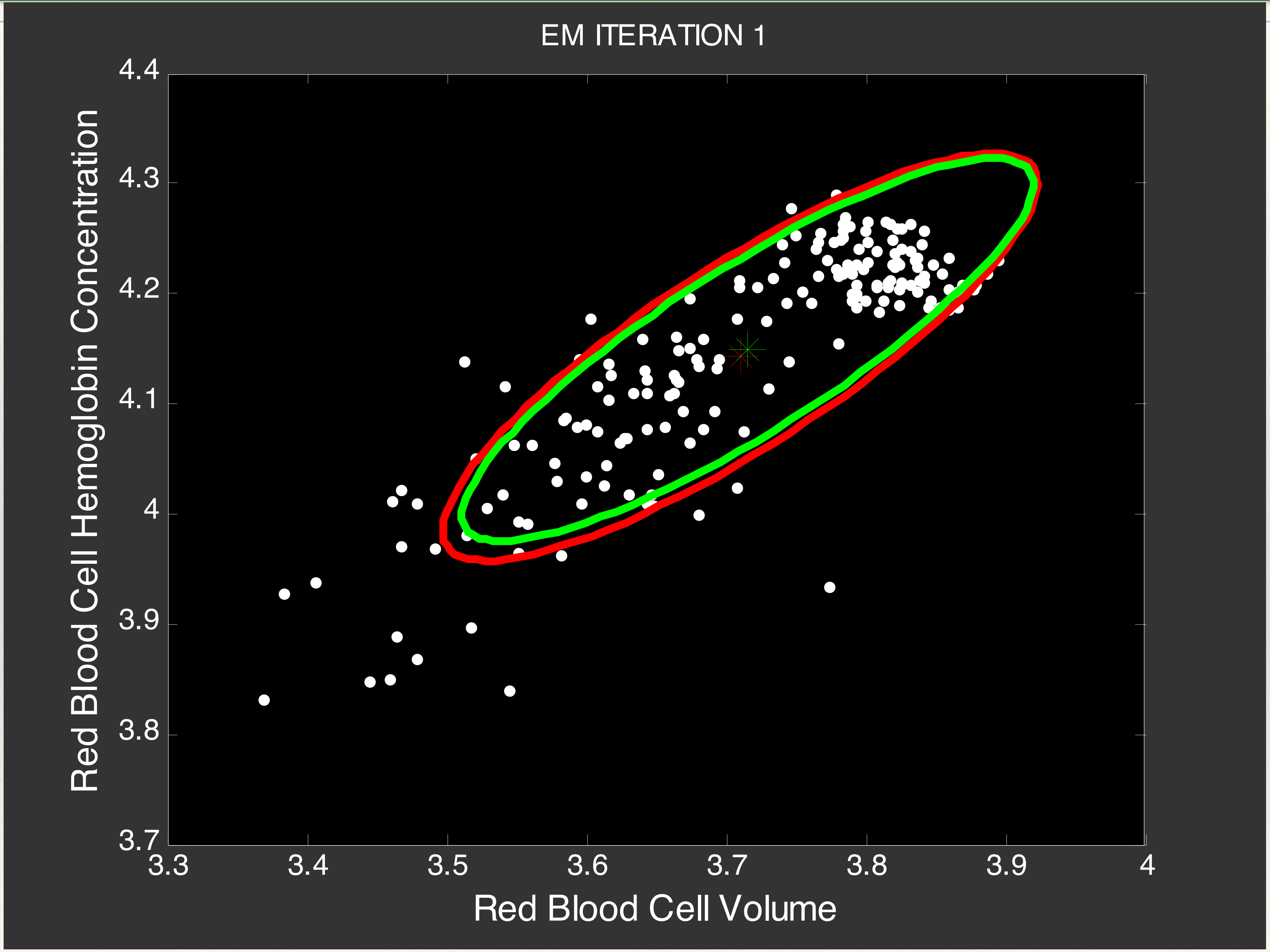




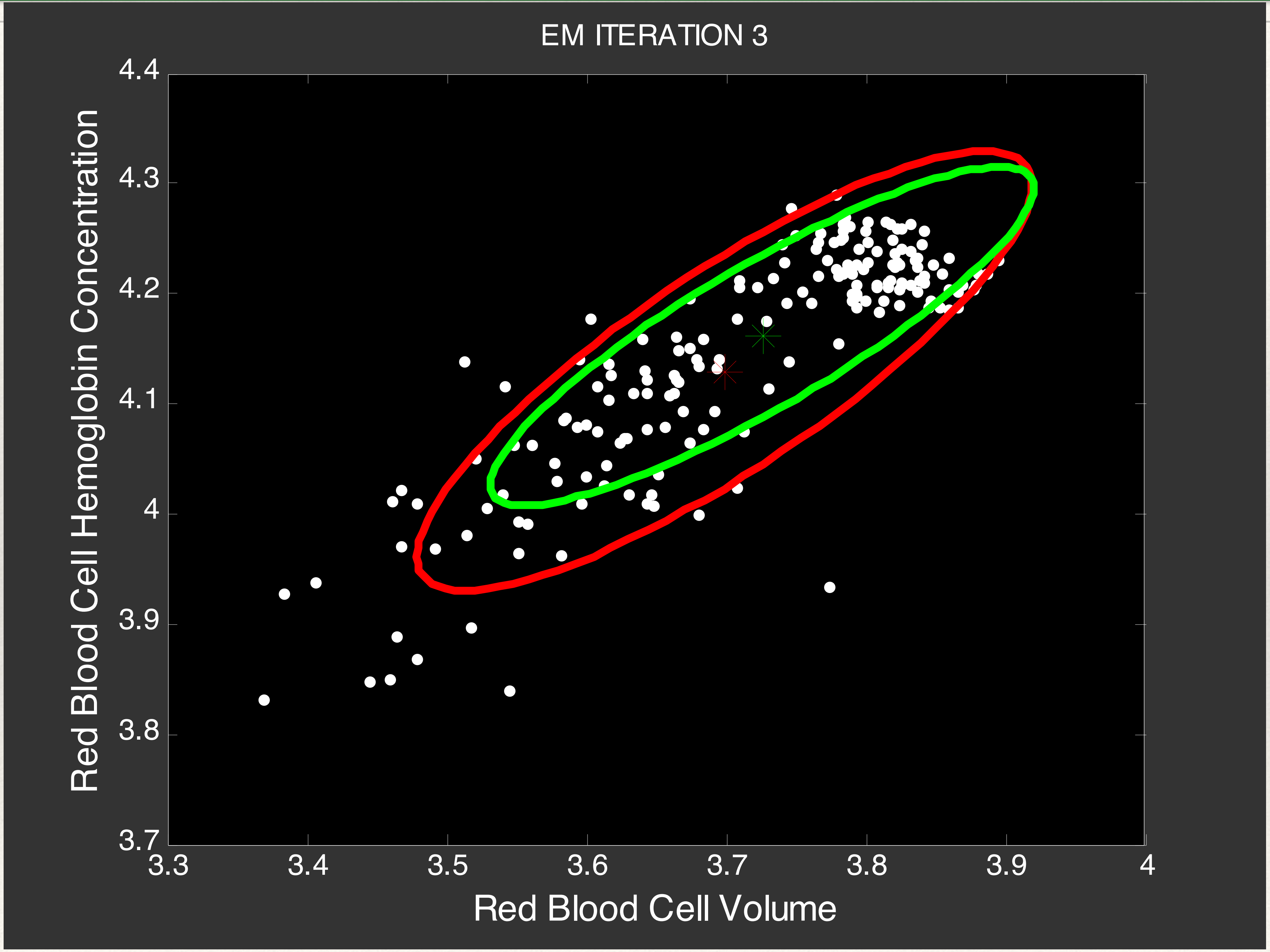
EM Algorithm for GMM



EM Algorithm for GMM

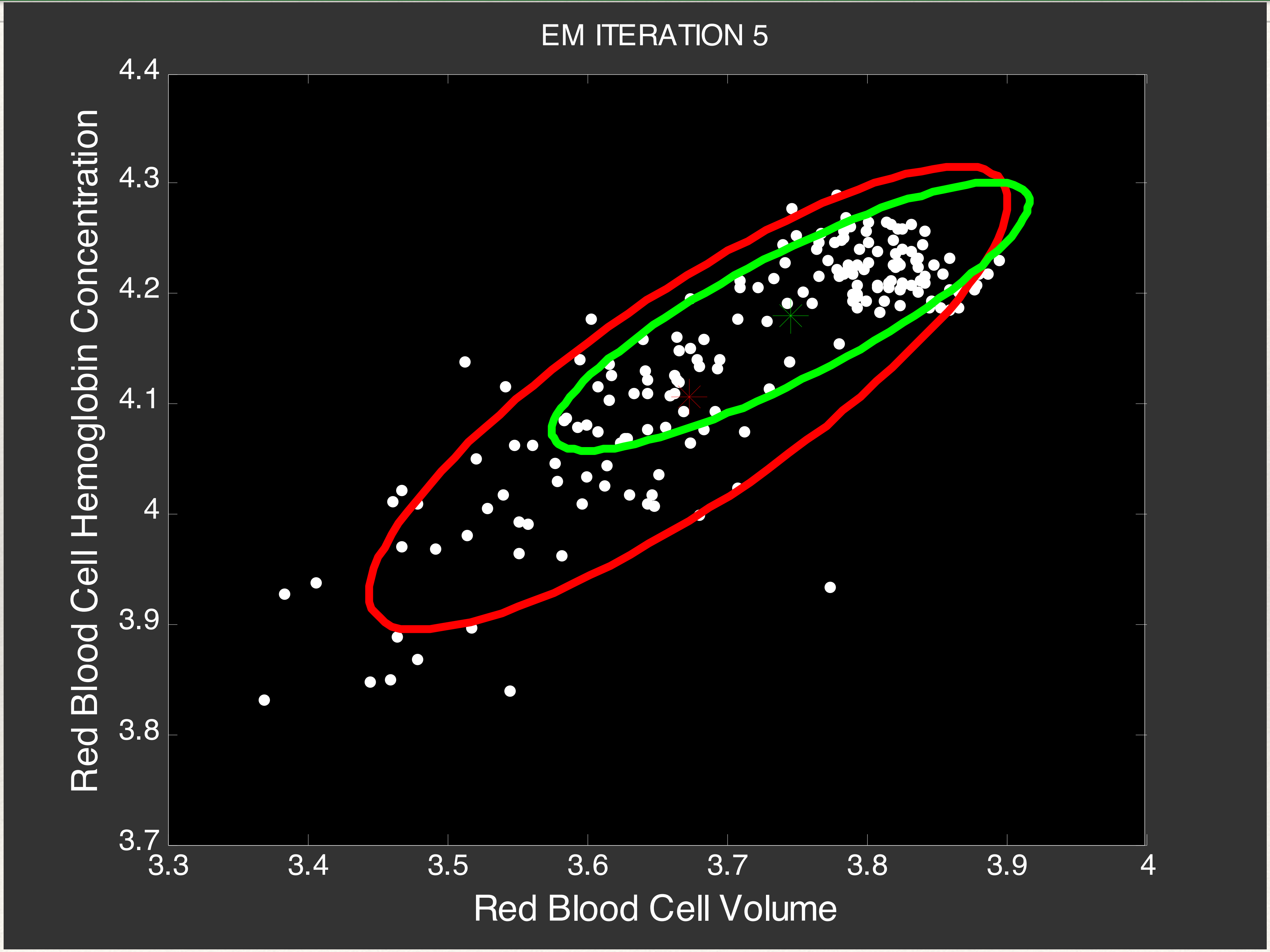


EM Algorithm for GMM

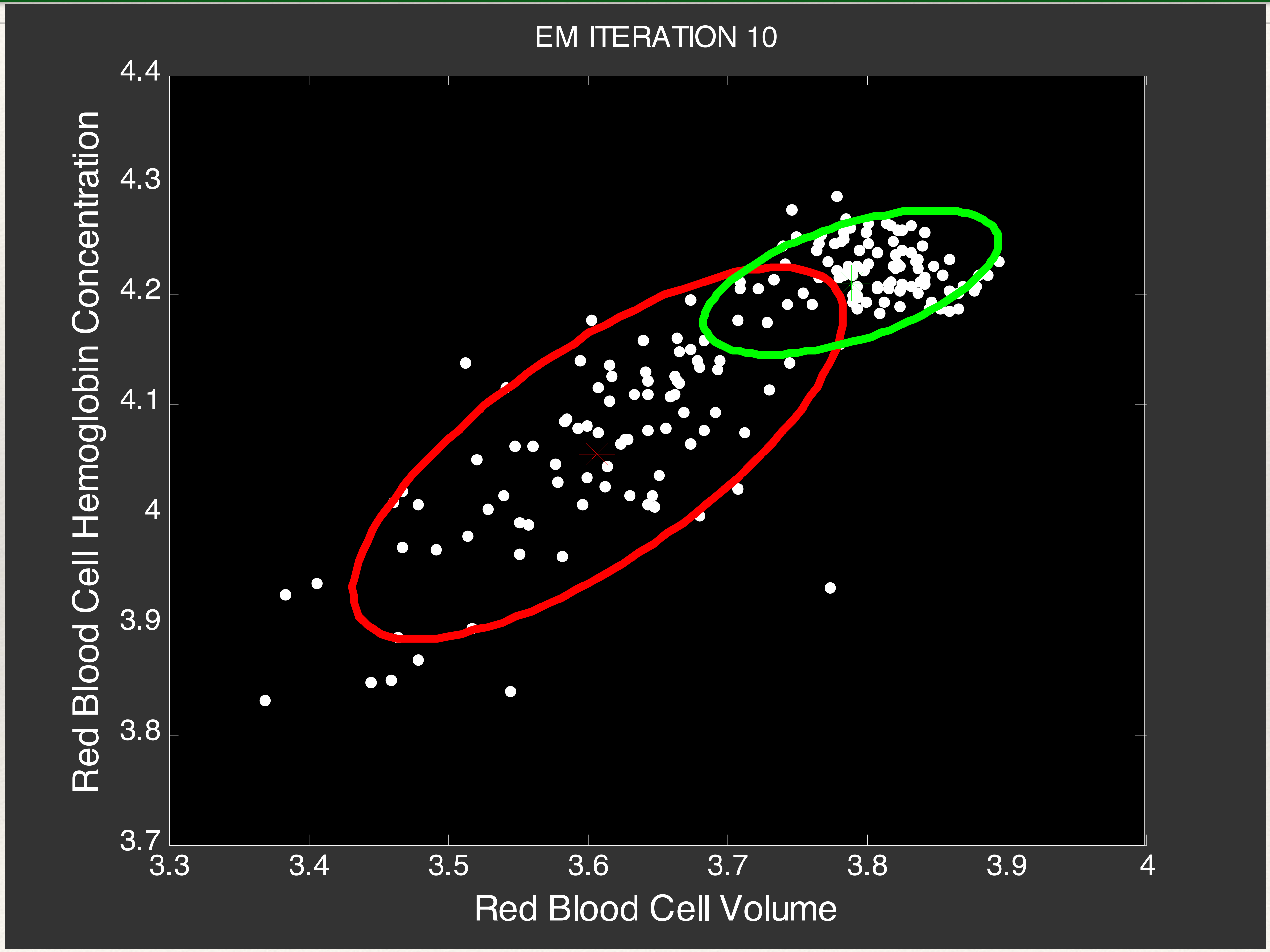


Established 1911

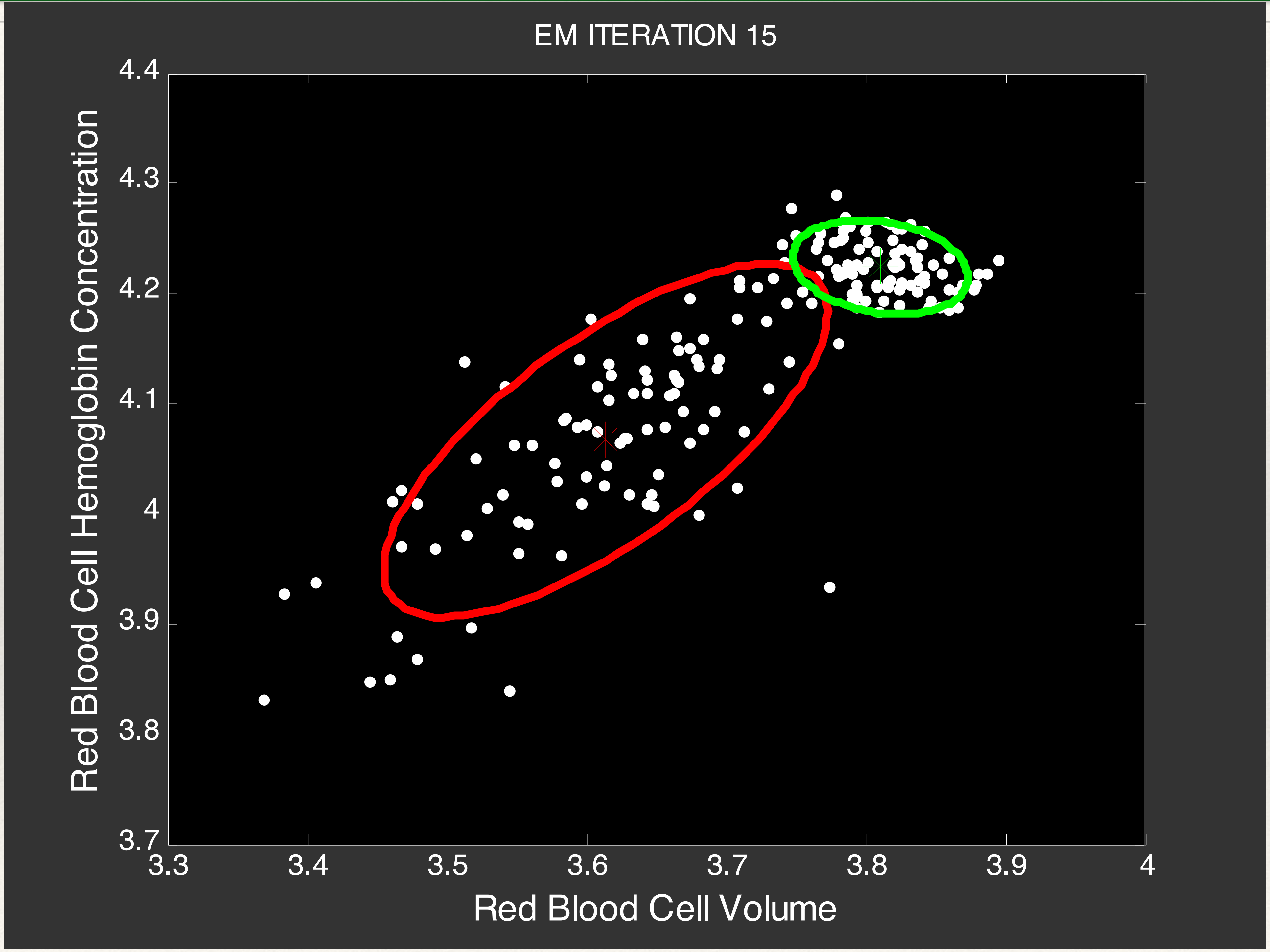
EM Algorithm for GMM



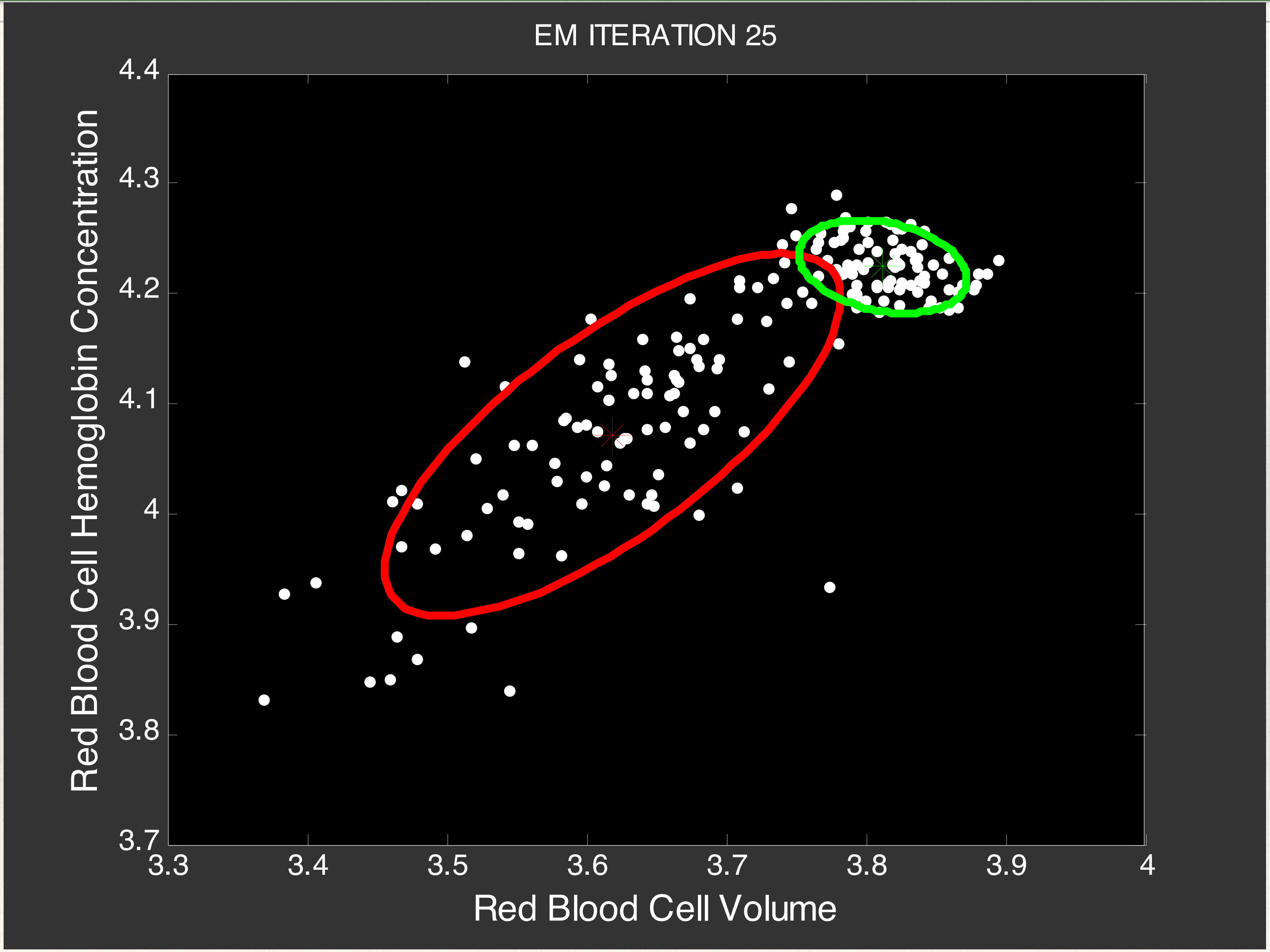
EM Algorithm for GMM



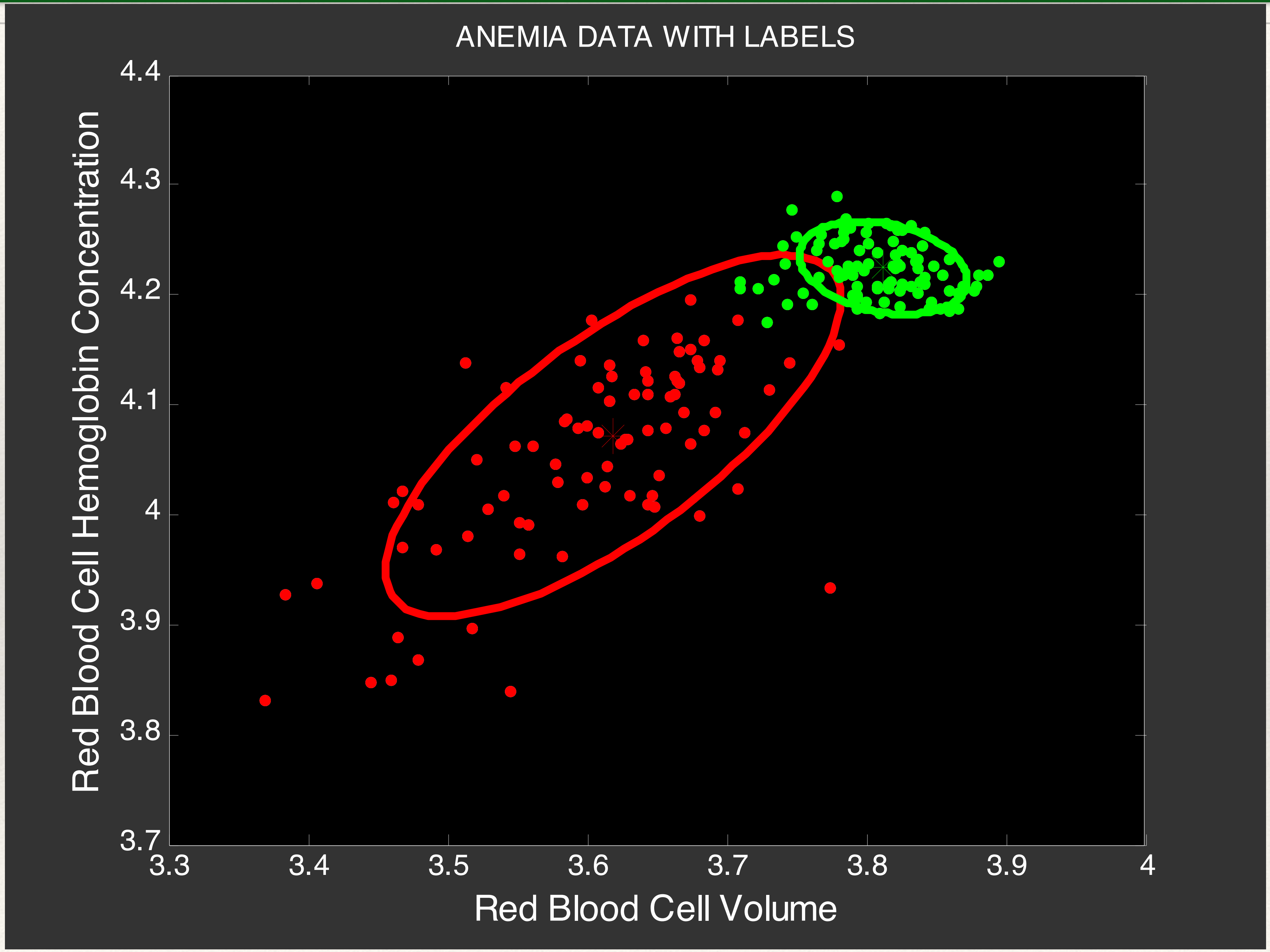
EM Algorithm for GMM



EM Algorithm for GMM

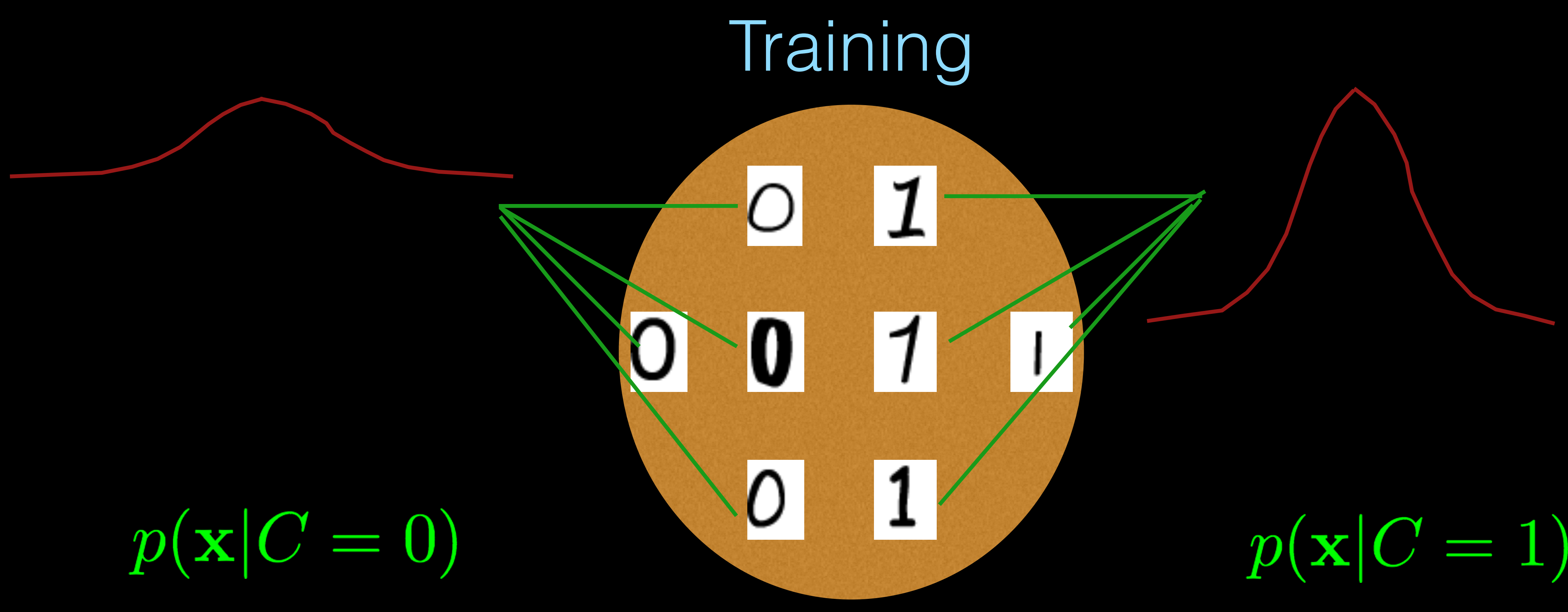


EM Algorithm for GMM



Generative classifier

- Model the two classes separately using probability distributions -
  - Make each sample $\mathbf{X}_i$ (28x28) as a vector $\mathbf{x}_i$ of size 784.
  - Build class dependent probability $p(\mathbf{x}|C=0)$ & $p(\mathbf{x}|C=1)$

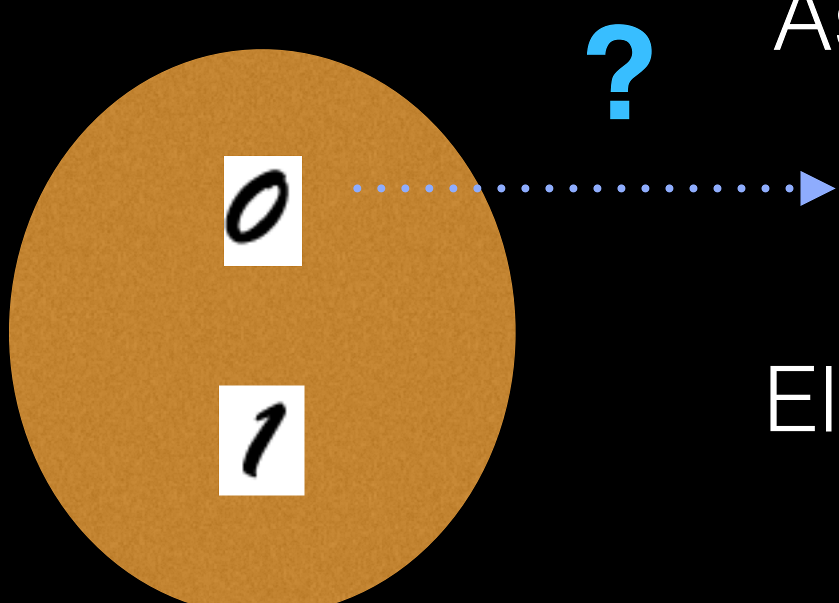


$p(\mathbf{x}|C=0)$ $p(\mathbf{x}|C=1)$

Generative classifier

- For the test sample
  - Make each sample $\mathbf{Y}$ (28x28) as a vector $\mathbf{y}$ of size 784.
  - Compute the probability of generating sample $\mathbf{y}$ for each class.

Test

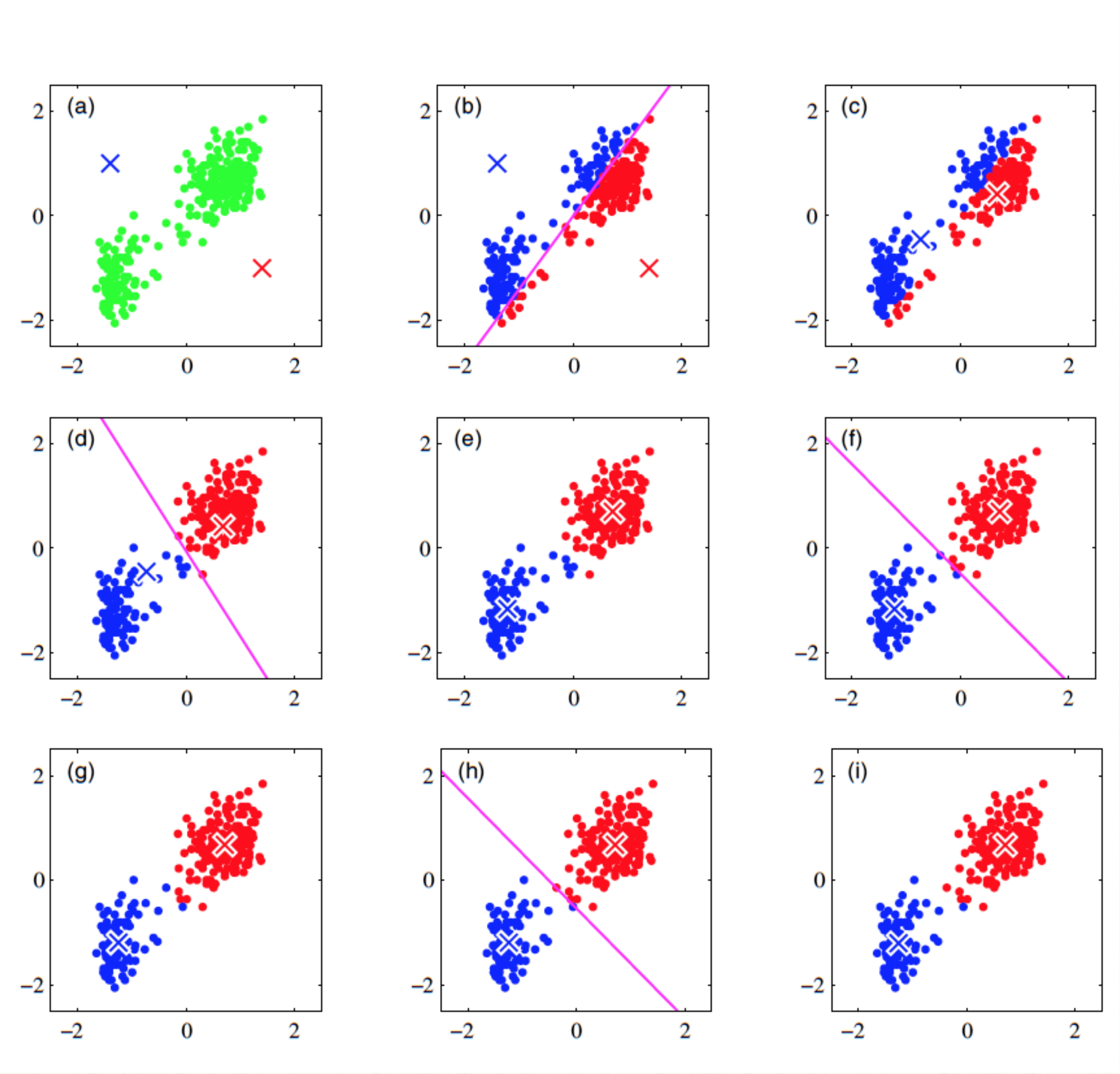


Assign to class label 0 if

$$p(\mathbf{y}|C=0) > p(\mathbf{y}|C=1)$$

Else assign to class 1

K-means Algorithm for Initialization



Discriminative models and Discriminant Functions

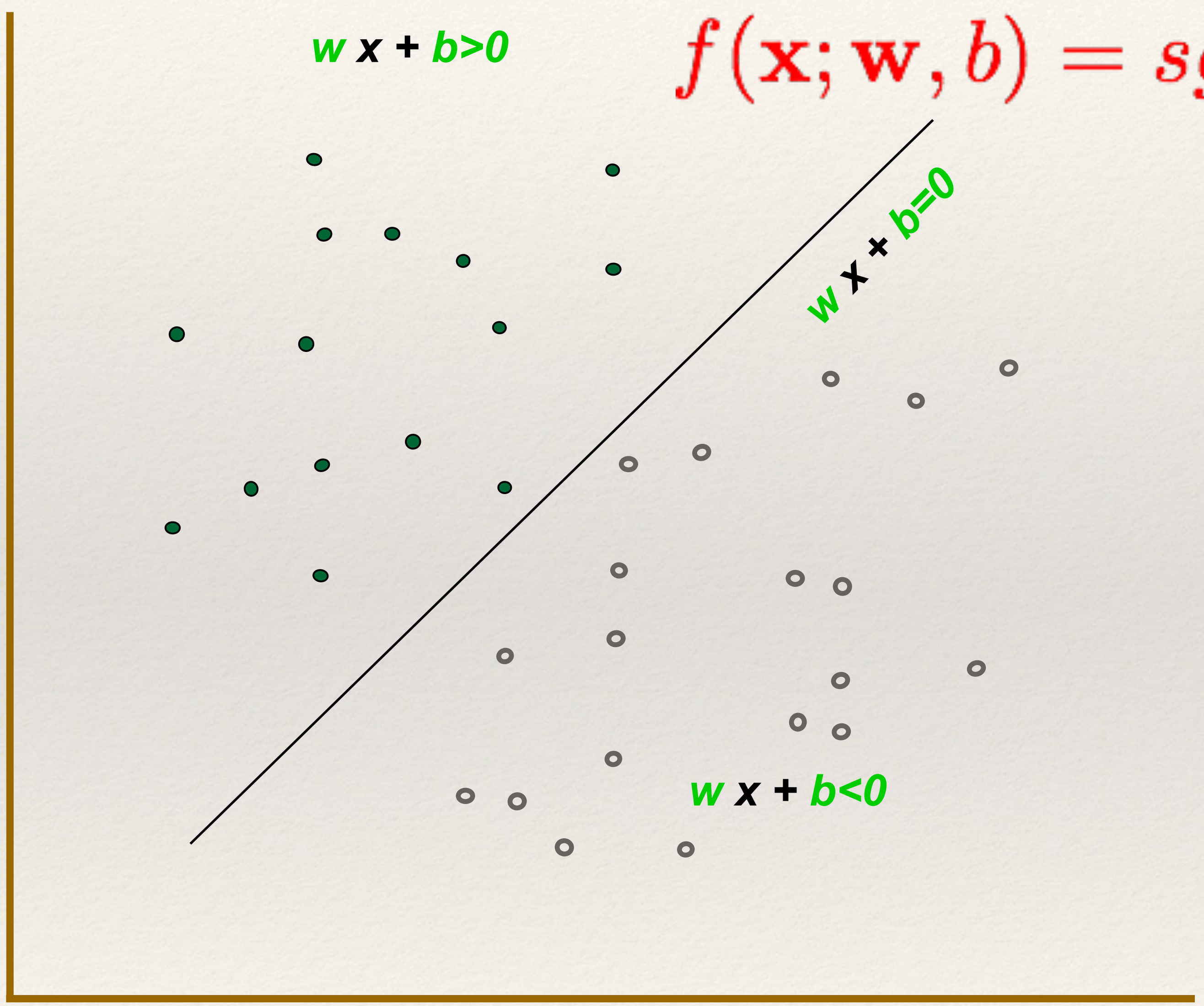
Support Vector Machines

Linear Classifiers

$$\mathbf{x} \longrightarrow \boxed{f} \longrightarrow y^{est}$$

$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$

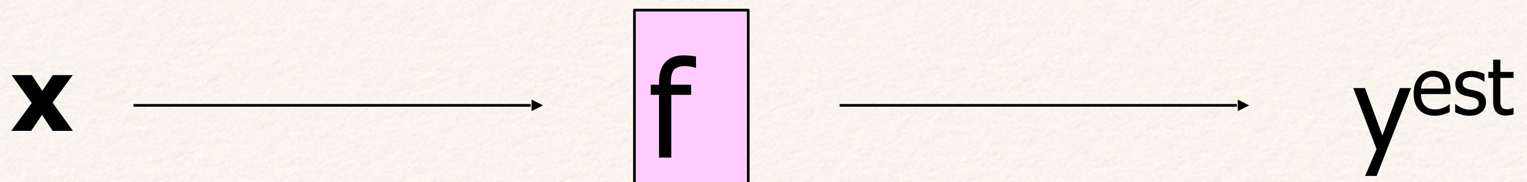
- denotes +1
- denotes -1



How would you classify this data?

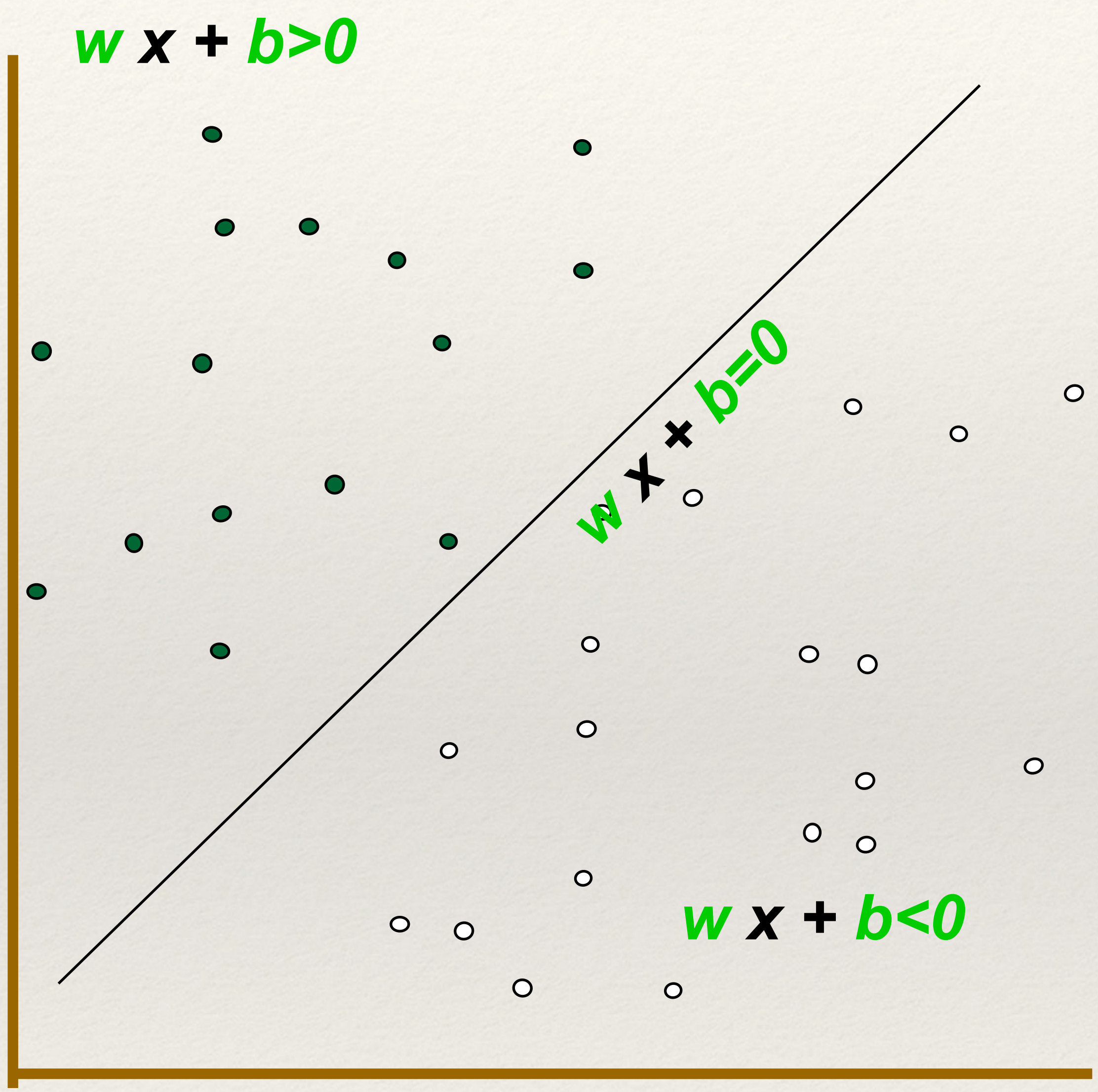
"SVM and applications", Mingyue Tan. Univ of British Columbia

Linear Classifiers



$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \phi(\mathbf{x}) + b)$$

- ● denotes +1
- ○ denotes -1



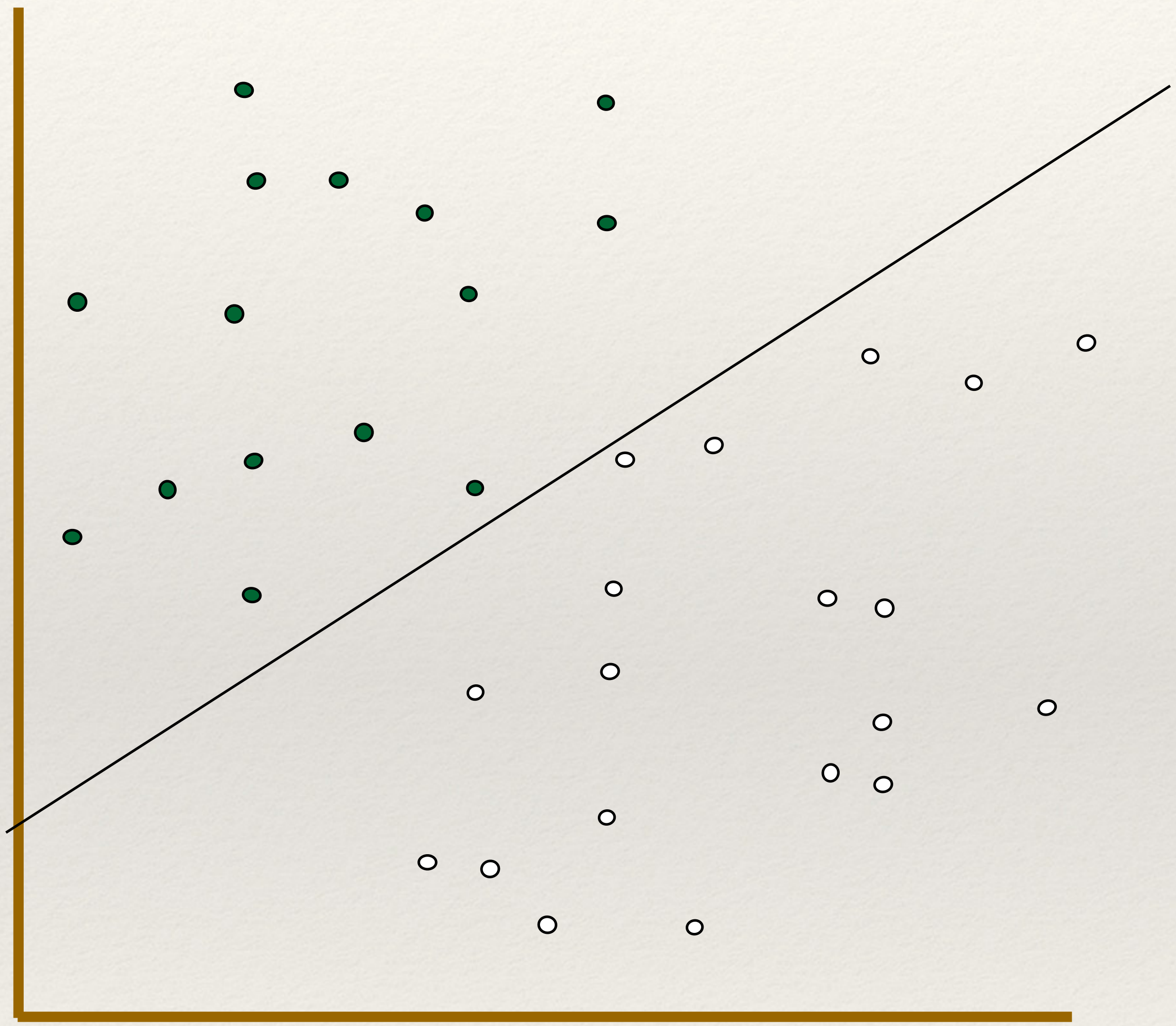
How would you classify this data?

Linear Classifiers

$$\mathbf{x} \longrightarrow \boxed{f} \longrightarrow y^{est}$$

$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b)$$

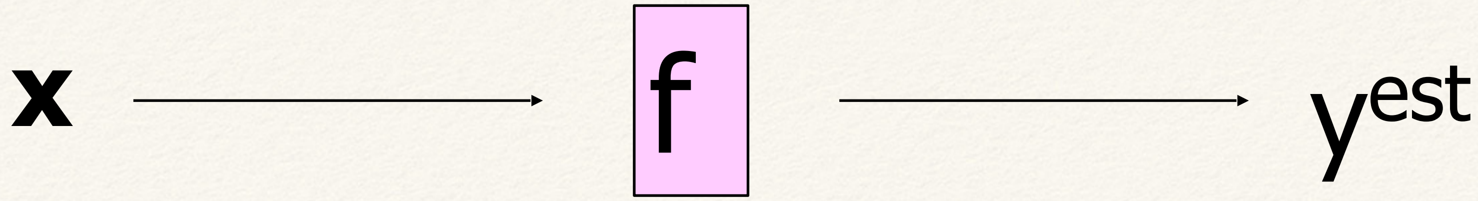
- denotes +1
- ◦ denotes -1



How would you classify this data?

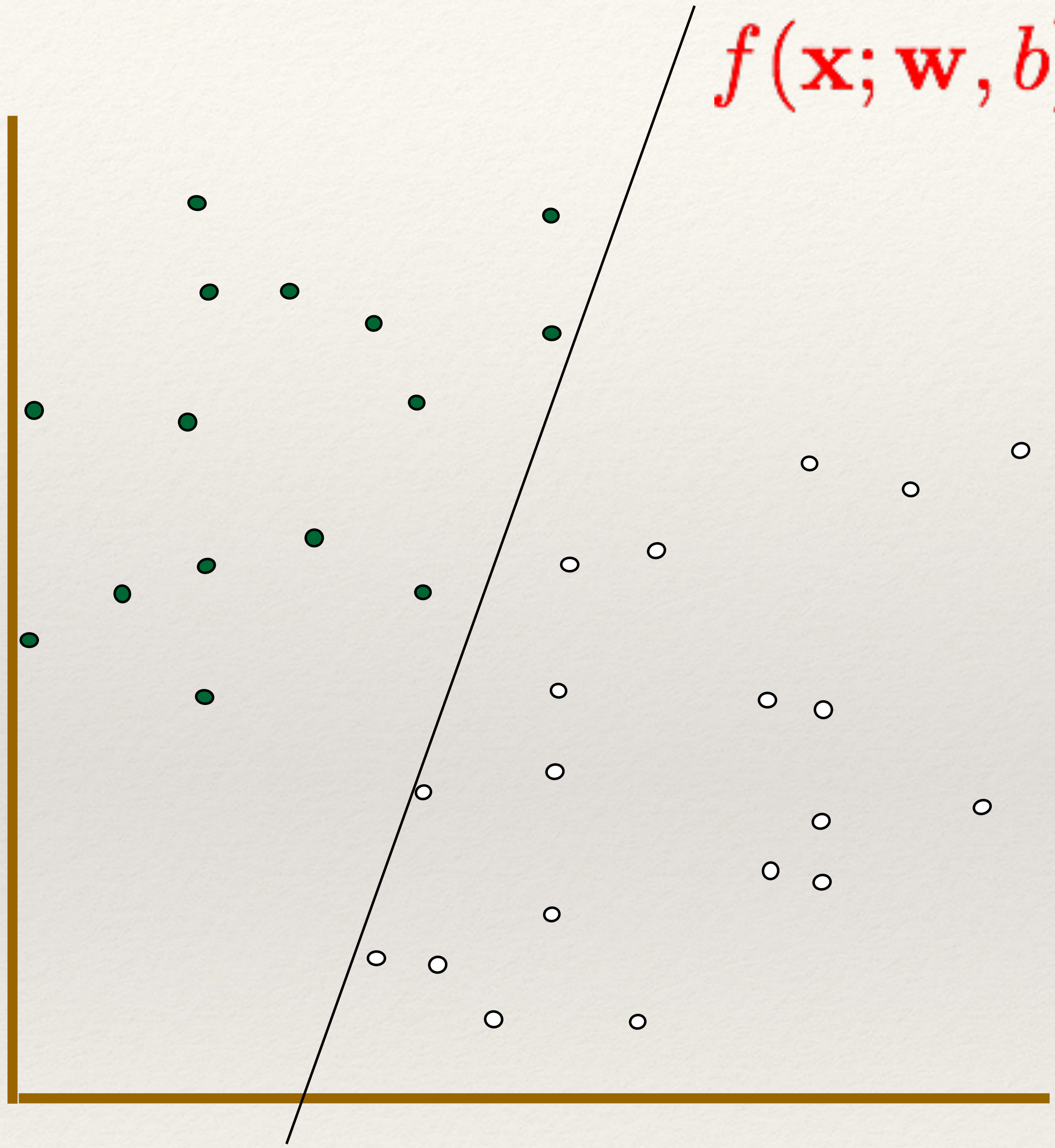
"SVM and applications", Mingyue Tan. Univ of British Columbia

Linear Classifiers



$$f(\mathbf{x};\mathbf{w},b) = sgn(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x})+b)$$

- • denotes +1
- ◦ denotes -1



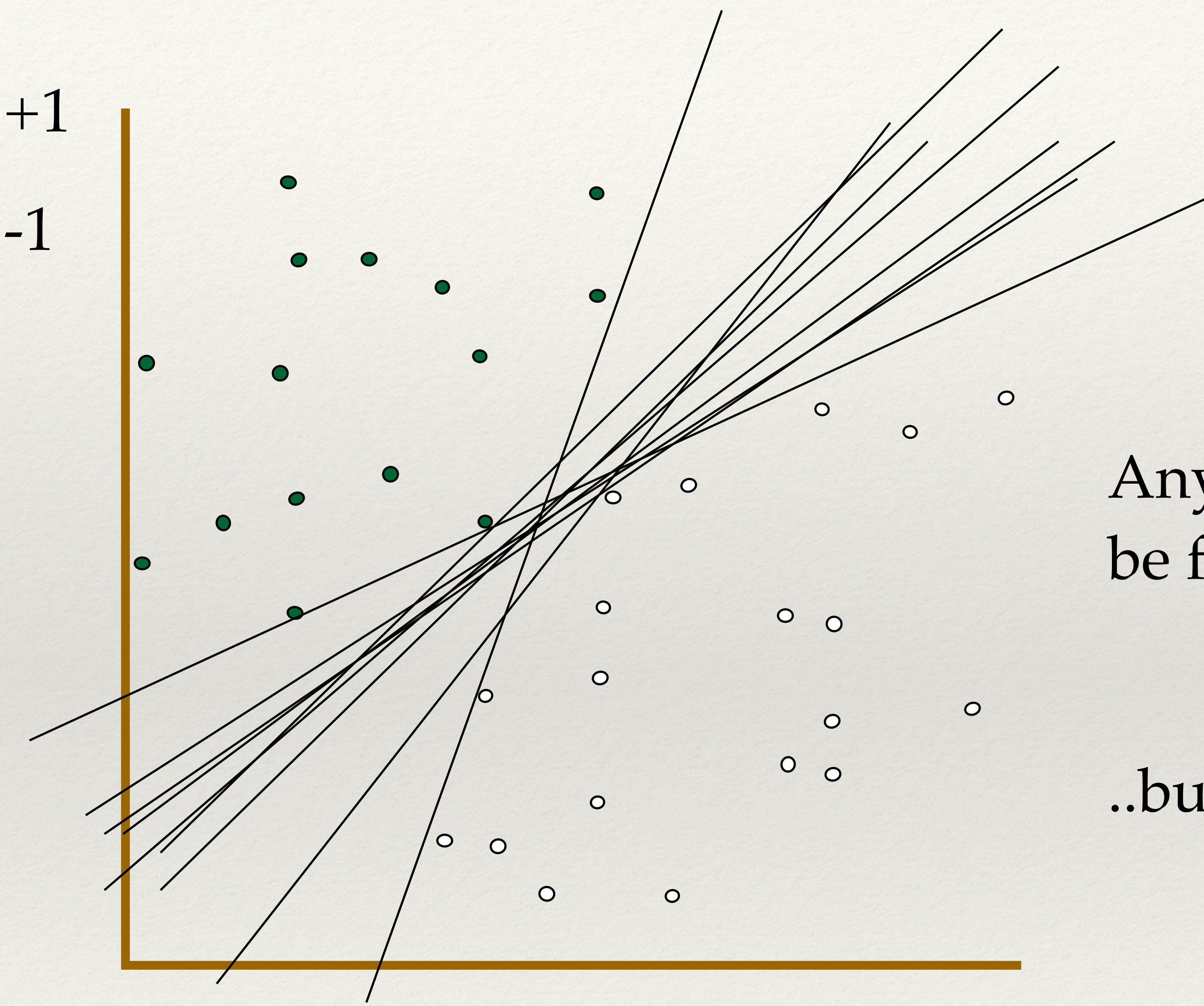
How would you classify this data?

"SVM and applications", Mingyue Tan. Univ of British Columbia

Linear Classifiers

$$\mathbf{x} \longrightarrow \boxed{f} \longrightarrow y^{est}$$

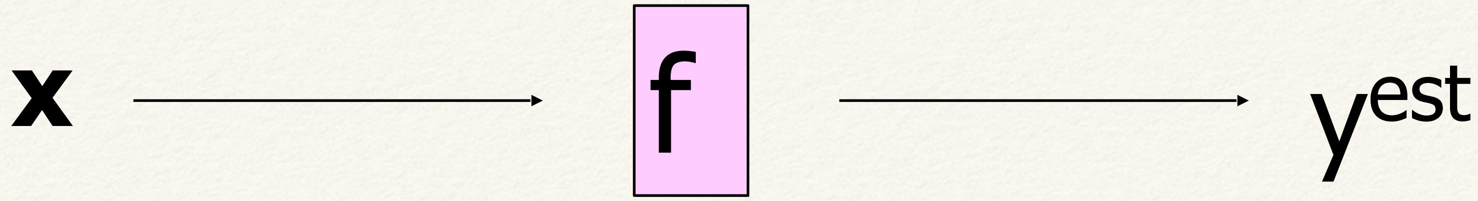
- • denotes +1
- ◦ denotes -1



Any of these would be fine..

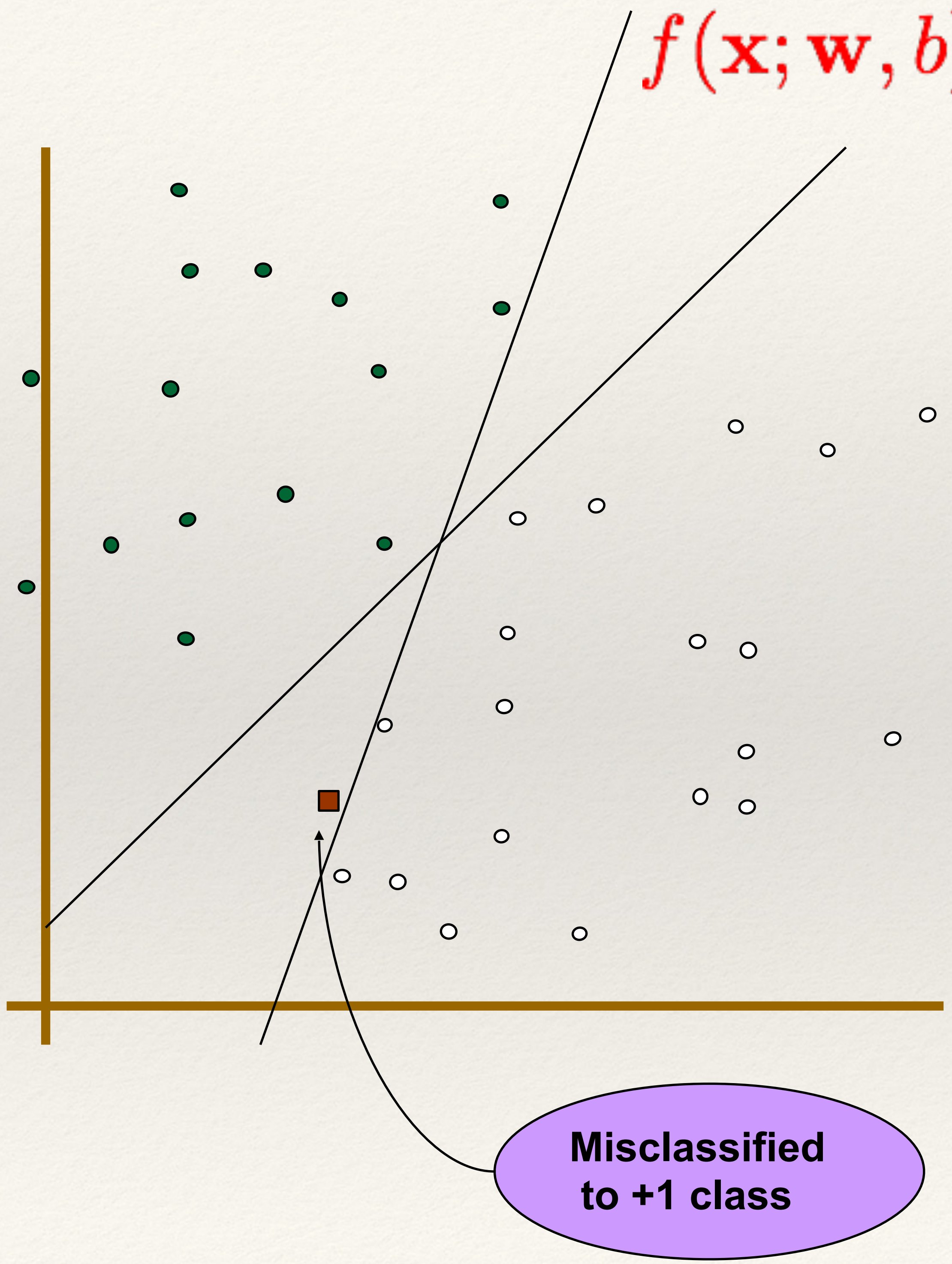
..but which is best?

Linear Classifiers



$$f(\mathbf{x}; \mathbf{w}, b) = sgn(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b)$$

- denotes +1
- denotes -1



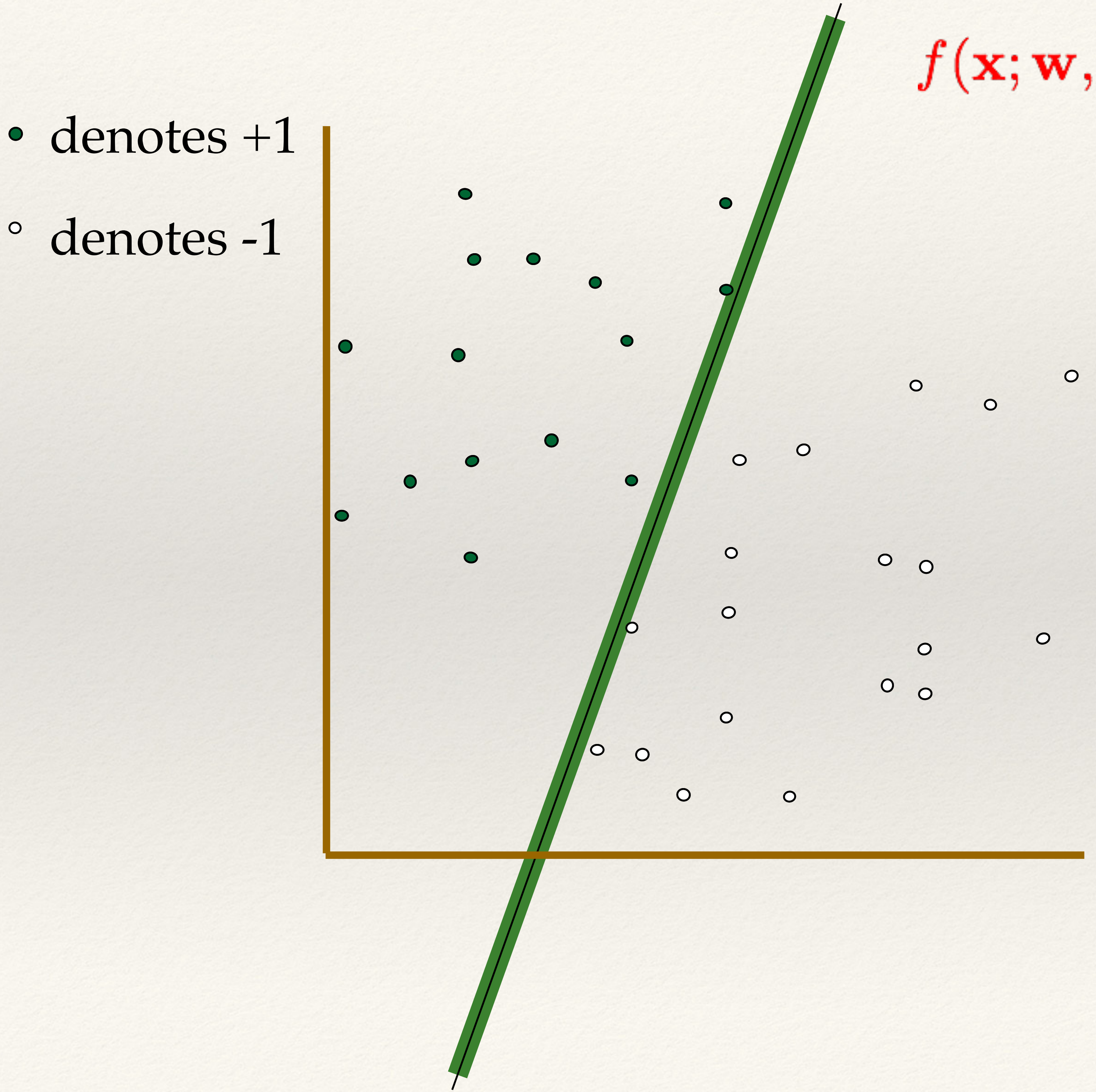
How would you classify this data?

"SVM and applications", Mingyue Tan. Univ of British Columbia

Linear Classifiers

$\mathbf{x}$ → f → y^{est}

$$f(\mathbf{x};\mathbf{w},b) = sgn(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}) + b)$$



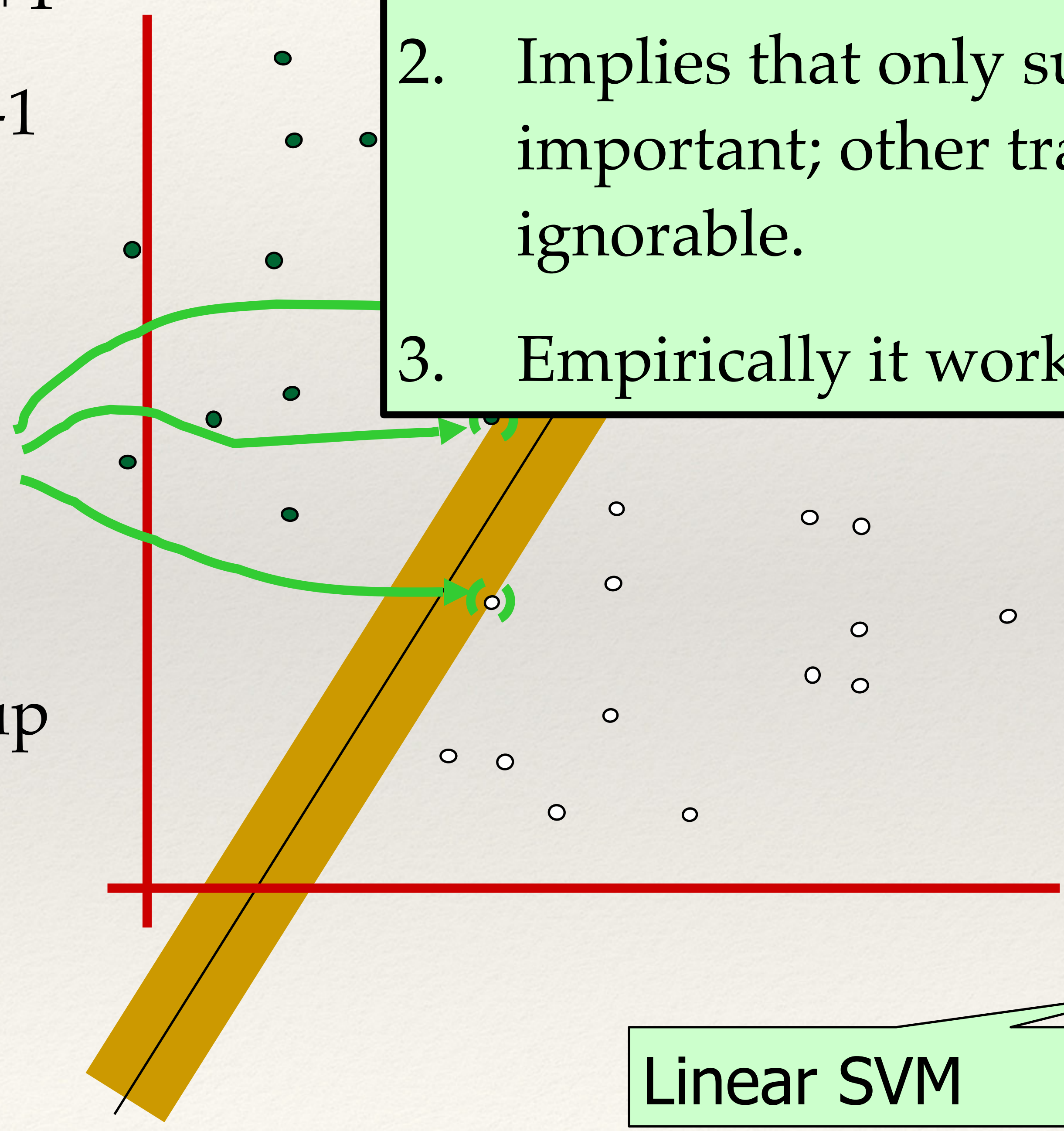
Define the **margin** of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin

1. Maximizing the margin is good according to intuition
2. Implies that only support vectors are important; other training examples are ignorable.
3. Empirically it works very very well.

• denotes +1

° denotes -1



Support Vectors are those data points that the margin pushes up against

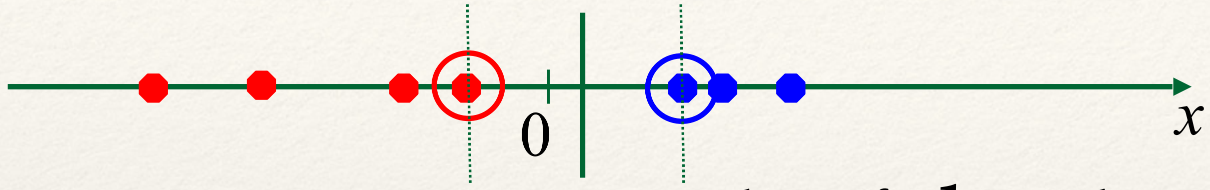
linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear SVM

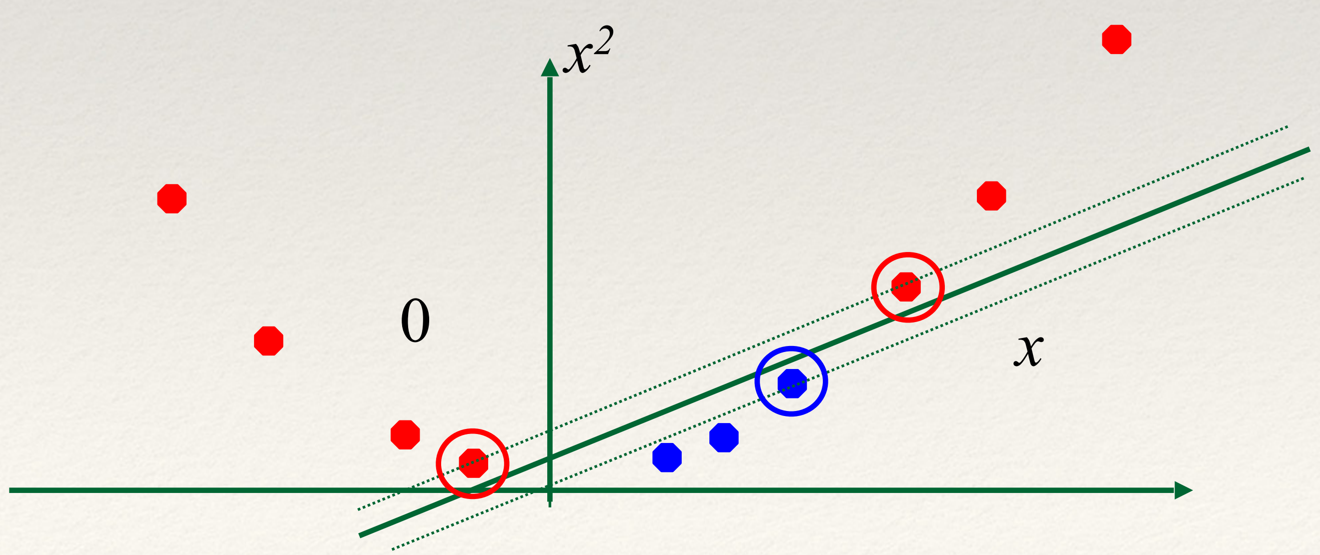
Non-linear SVMs

- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable: