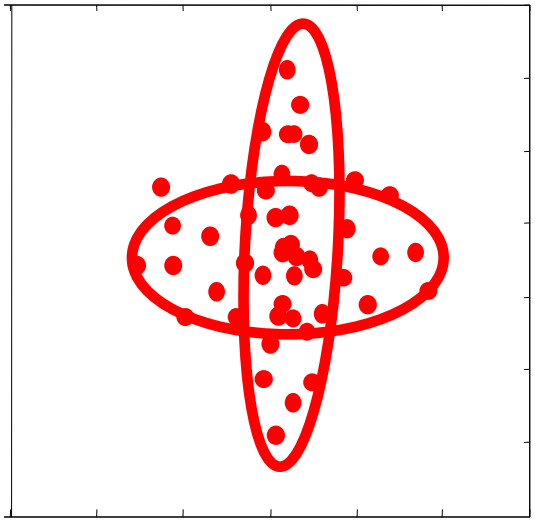
Expectation-Maximization (EM) Algorithm

Adopted from slides by Alexander Ihler

Probabilistic models in unsupervised learning

- K-means algorithm
  - Assigned each example to exactly one cluster
  - What if clusters are overlapping?
    - Hard to tell which cluster is right
    - Maybe we should try to remain uncertain
  - Used Euclidean distance
  - What if cluster has a non-circular shape?

- EM algorithm
  - Assign data to cluster with some probability
  - Gives probability model of x! (“generative”)



Expectation-Maximization (EM) Algorithm

- Learning algorithm for latent variable models
- Observed features x: $x^{(1)}, x^{(2)}, \cdots, x^{(m)}$
- Latent features z: $z^{(1)}, z^{(2)}, \cdots, z^{(m)}$

- Assume a probabilistic model over x, z

$$P_\theta(x, z) = P_{\theta'}(x|z)P_{\theta''}(z)$$

- Learning most likely parameters θ based on the observed data

$$\arg\max_\theta P_\theta(X) = \sum_Z P_\theta(X, Z)$$

Expectation-Maximization (EM) Algorithm

- Iteratively update θ and z

- Initially assume random parameters θ
- Iterate following two steps until convergence:
  - Expectation (E-step): Compute $P_\theta\left(z^{(i)}\middle|x^{(i)}\right)$ for each example i based on the current parameters θ
  - Maximization (M-step): Re-estimate the most likely parameters θ based on the expected data x, z

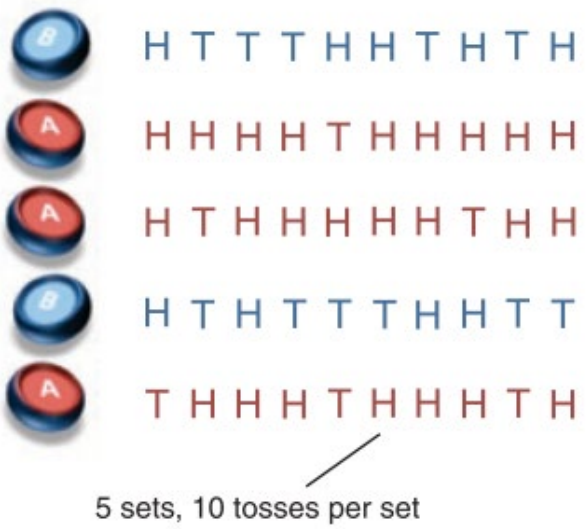
$$\arg\max_\theta P_\theta(X) = \sum_Z P_\theta(X, Z)$$

Coin tossing example

$$\hat{\theta}_A = \frac{\#\text{ of heads using coin } A}{\text{total }\#\text{ of flips using coin } A}$$

$$\hat{\theta}_B = \frac{\#\text{ of heads using coin } B}{\text{total }\#\text{ of flips using coin } B}$$

- Two coins A and B with unknown biases θ_A and θ_B
- Repeat following procedure 5 times:
  - Randomly choose one of the two coins
  - Perform 10 independent coin tosses with selected coin



Coin A	Coin B
	5 H, 5 T
9 H, 1 T	
8 H, 2 T	
	4 H, 6 T
7 H, 3 T	
24 H, 6 T	9 H, 11 T

$$\hat{\theta}_A = \frac{24}{24+6} = 0.80$$

$$\hat{\theta}_B = \frac{9}{9+11} = 0.45$$

EM Algorithm

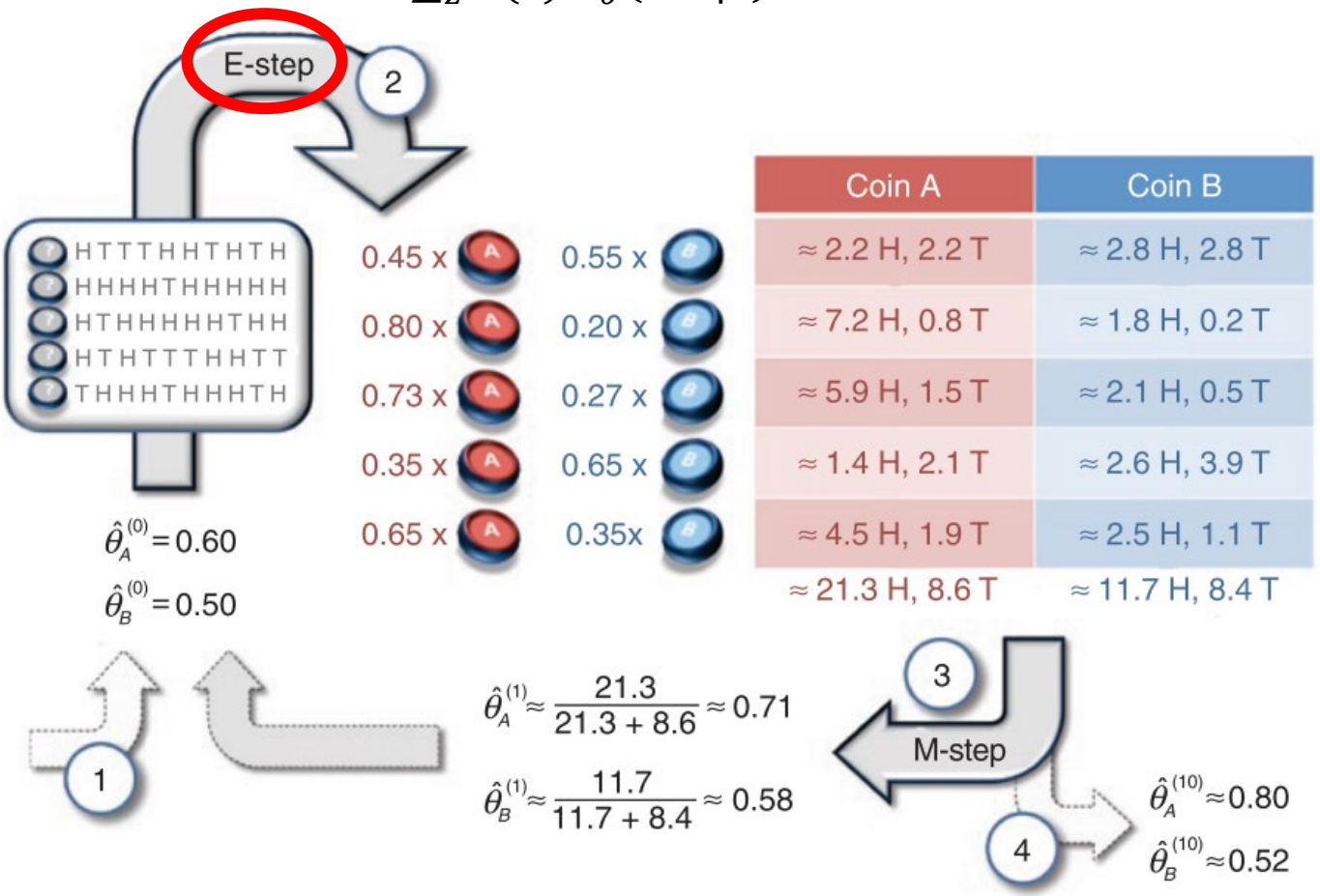
- Observed feature $x \in \{0,1,\cdots,10\}$: # of heads
  - Observed data:
    $x^{(1)} = 5, x^{(2)} = 9, x^{(3)} = 8, x^{(4)} = 4, x^{(5)} = 7$
- Latent feature $z \in \{A, B\}$: identity of the coin
  - $z^{(1)}, z^{(2)}, z^{(3)}, z^{(4)}, z^{(5)}$
  - Assume prior probability $P(z) = 0.5$

- Model
  - Parameters $\theta = \{\theta_A, \theta_B\}$

$$P_\theta(x|z) = \begin{cases} C_{10}^x \cdot (\theta_A)^x \cdot (1-\theta_A)^{10-x} & \text{if } z = 'A' \\ C_{10}^x \cdot (\theta_B)^x \cdot (1-\theta_B)^{10-x} & \text{if } z = 'B' \end{cases}$$

EM Algorithm

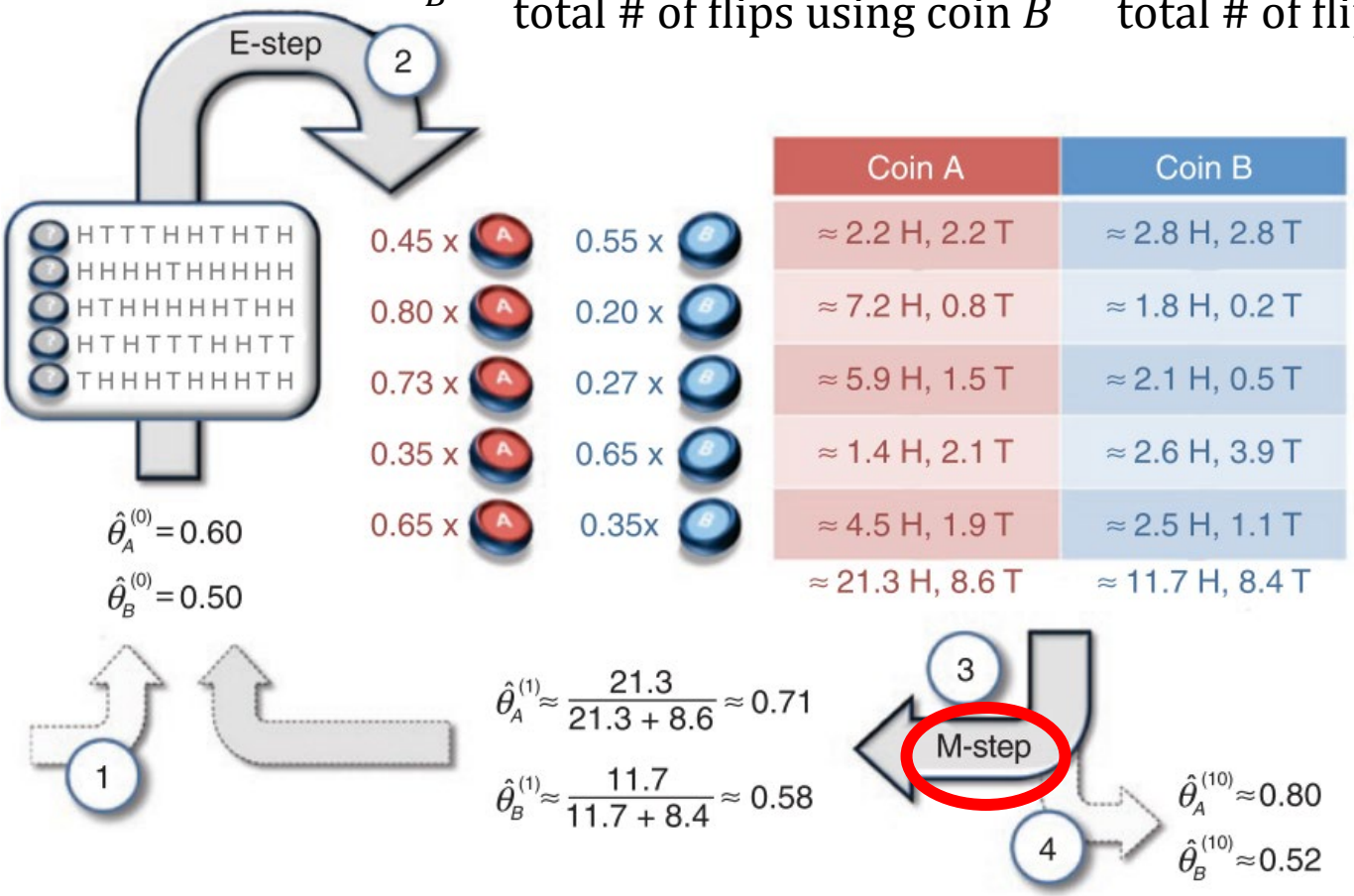
$$P_\theta\left(z^{(i)}\middle|x^{(i)}\right) = \frac{P\left(z^{(i)}\right)P_\theta\left(x^{(i)}\middle|z^{(i)}\right)}{\sum_z P(z)\,P_\theta(x^{(i)}|z)}$$



EM Algorithm

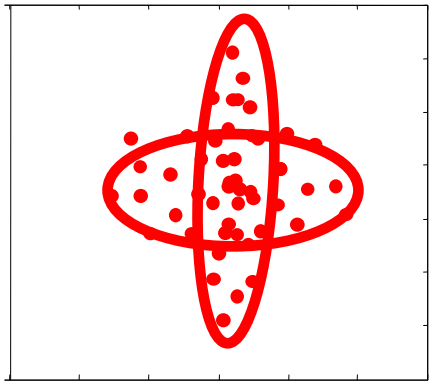
$$\hat{\theta}_A = \frac{\#\text{ of heads using coin } A}{\text{total }\#\text{ of flips using coin } A} = \frac{\#\text{ of heads} \times \text{pro of } A}{\text{total }\#\text{ of flips} \times \text{pro of } A}$$

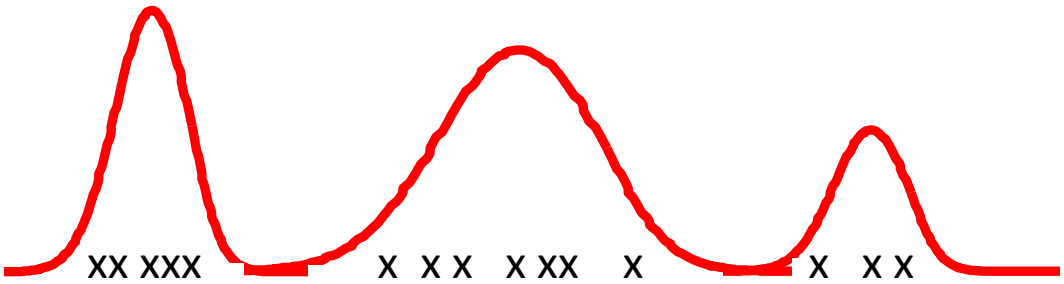
$$\hat{\theta}_B = \frac{\#\text{ of heads using coin } B}{\text{total }\#\text{ of flips using coin } B} = \frac{\#\text{ of heads} \times \text{pro of } B}{\text{total }\#\text{ of flips} \times \text{pro of } B}$$



EM for Clustering: Mixtures of Gaussians

- Start with parameters describing each cluster
- Mean μ_c, variance σ_c, "size" π_c
- Probability distribution: $$p(x) = \sum_c \pi_c \, \mathcal{N}(x \; ; \; \mu_c, \sigma_c)$$





Mixtures of Gaussians

- Start with parameters describing each cluster
- Mean μ_c, variance σ_c, "size" π_c
- Probability distribution: $$p(x) = \sum_c \pi_c \, \mathcal{N}(x \; ; \; \mu_c, \sigma_c)$$
- Equivalent "latent variable" form:

$$p(z = c) = \pi_c$$

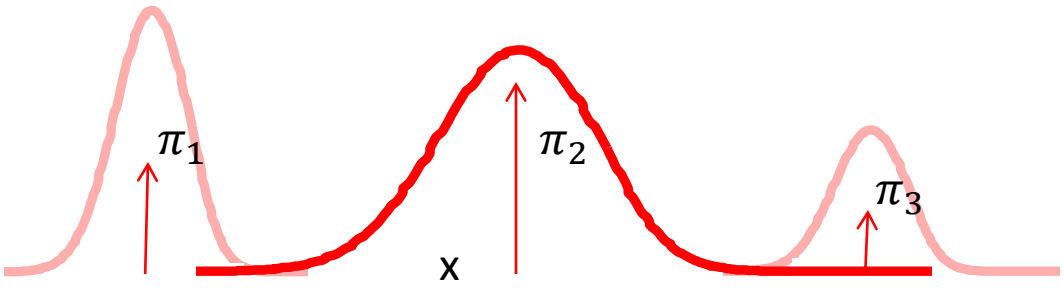
Select a mixture component with probability π

$$p(x|z = c) = \mathcal{N}(x \; ; \; \mu_c, \sigma_c)$$

Sample from that component's Gaussian

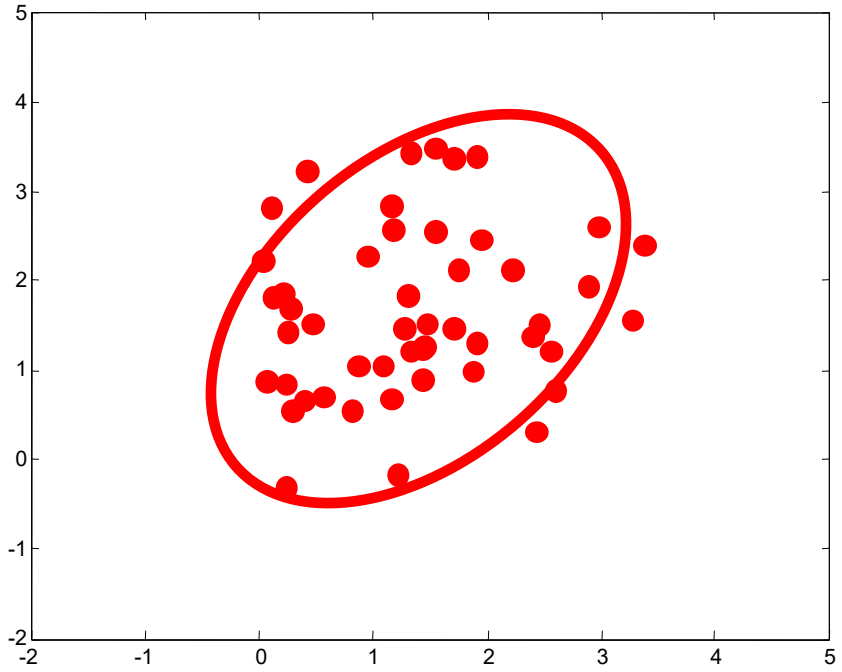
"Latent assignment" z:
we observe x, but z is hidden

p(x) = marginal over x



Multivariate Gaussian models

$$\mathcal{N}(\underline{x}\ ;\ \underline{\mu}, \Sigma) = \frac{1}{(2\pi)^{d/2}} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} (\underline{x} - \underline{\mu})^T \Sigma^{-1} (\underline{x} - \underline{\mu}) \right\}$$



Maximum Likelihood estimates

$$\hat{\mu} = \frac{1}{m} \sum_i x^{(i)}$$

$$\hat{\Sigma} = \frac{1}{m} \sum_i (x^{(i)} - \hat{\mu})^T (x^{(i)} - \hat{\mu})$$

We'll model each cluster using one of these Gaussian "bells"…

EM Algorithm

- Observed feature $x \in \mathbb{R}^d$
- Latent feature $z \in \{c_1, c_2, c_3\}$
- Model
  - Parameters: Mean μ_c, variance Σ_c, "size" π_c for each c

$$P(x|z = c) = \mathcal{N}(x; \mu_c, \Sigma_c)$$

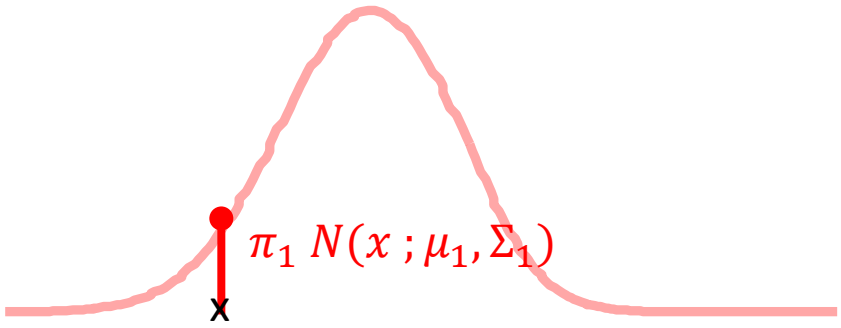
$$P(z = c) = \pi_c$$

$$P(x, z = c) = P(x|z = c)P(z = c) = \pi_c \cdot \mathcal{N}(x; \mu_c, \Sigma_c)$$

EM Algorithm: E-step

- Start with clusters: Mean μ_c, Covariance Σ_c, "size" π_c
- E-step ("Expectation")
  - For each datum (example) x_i,
  - Compute "r_{ic}", the probability that it belongs to cluster c
    - Compute its probability under model c
    - Normalize to sum to one (over clusters c)

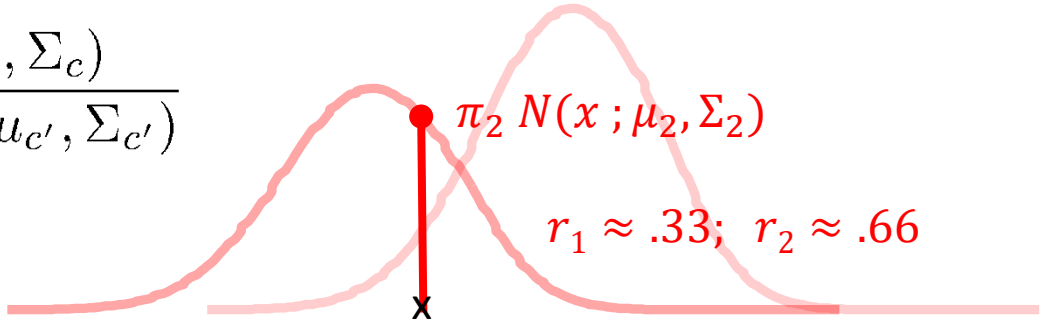
$$P\left(z^{(i)}\middle|x^{(i)}\right) = \; r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$



EM Algorithm: E-step

- Start with clusters: Mean μ_c, Covariance Σ_c, "size" π_c
- E-step ("Expectation")
  - For each datum (example) x_i,
  - Compute "r_{ic}", the probability that it belongs to cluster c
    - Compute its probability under model c
    - Normalize to sum to one (over clusters c)

$$P\left(z^{(i)}|x^{(i)}\right) = \; r_{ic} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$



- If x_i is very likely under the c^{th} Gaussian, it gets high weight
- Denominator just makes r's sum to one

EM Algorithm: M-step

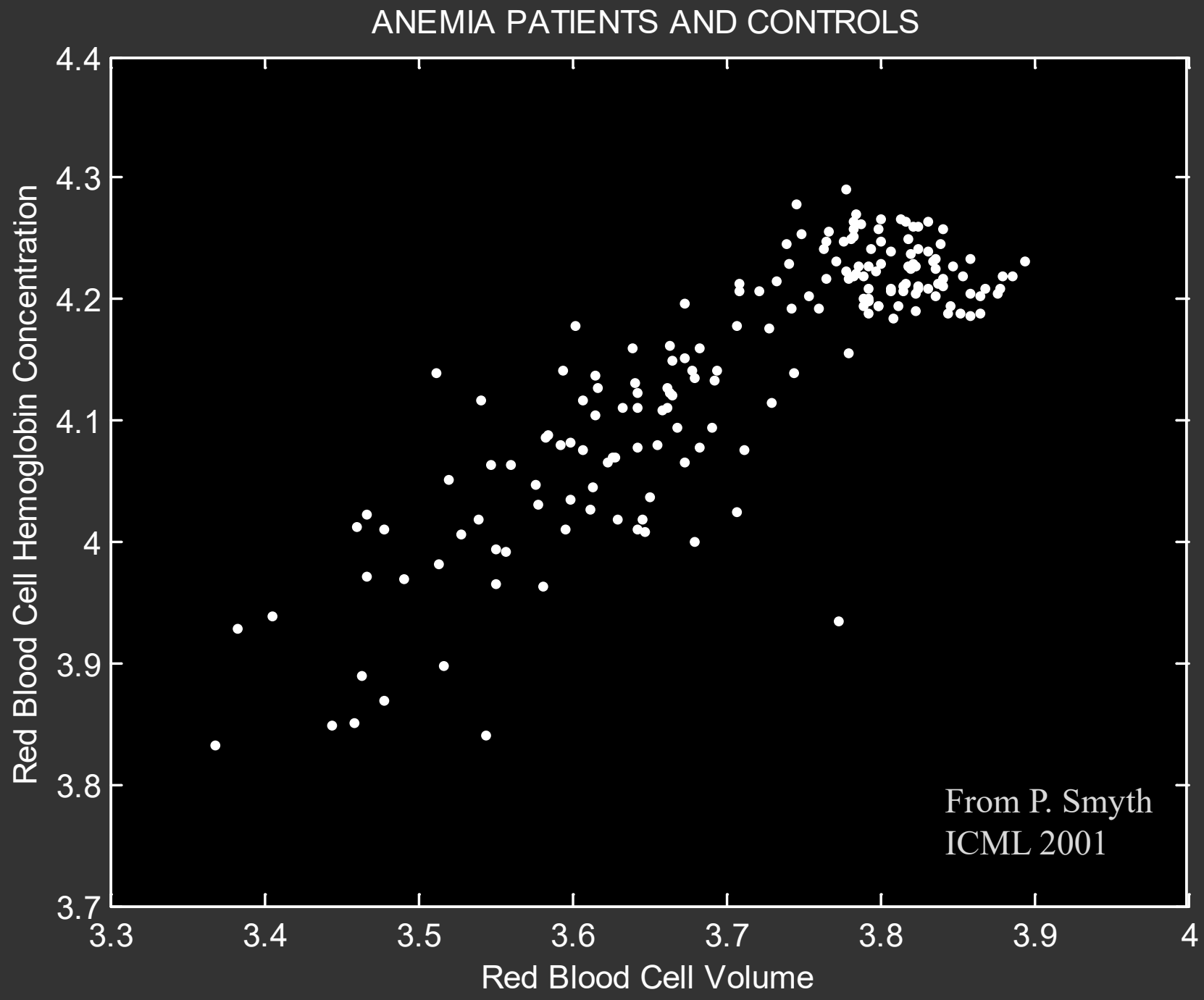
- Start with assignment probabilities r_{ic}
- Update parameters: Mean μ_c, Covariance Σ_c, "size" π_c
- M-step ("Maximization")
  - For each cluster (Gaussian) z = c,
  - Update its parameters using the (weighted) data points

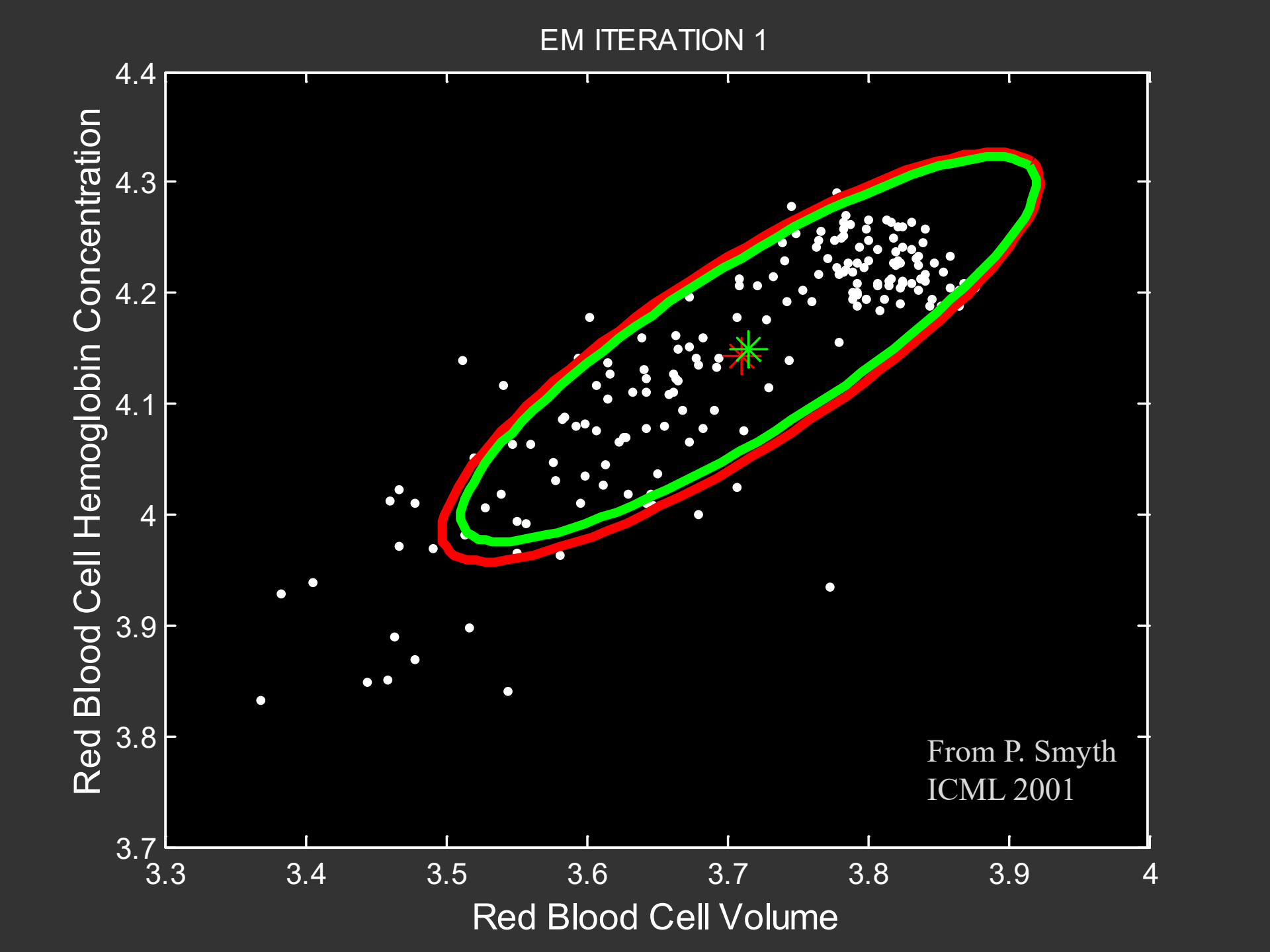
$$m_c = \sum_i r_{ic}$$ Total responsibility allocated to cluster c

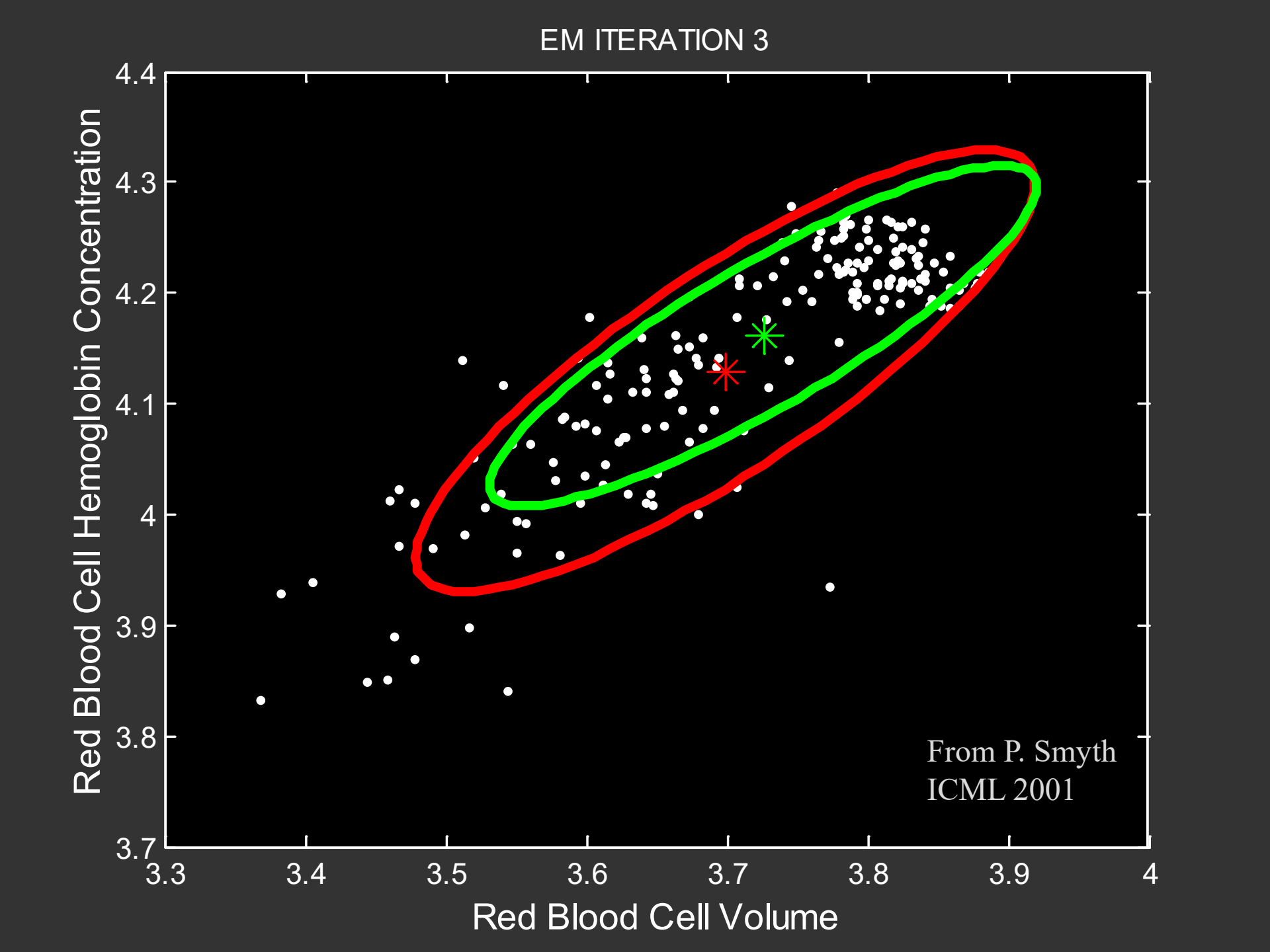
$$\pi_c = \frac{m_c}{m}$$ Fraction of total assigned to cluster c

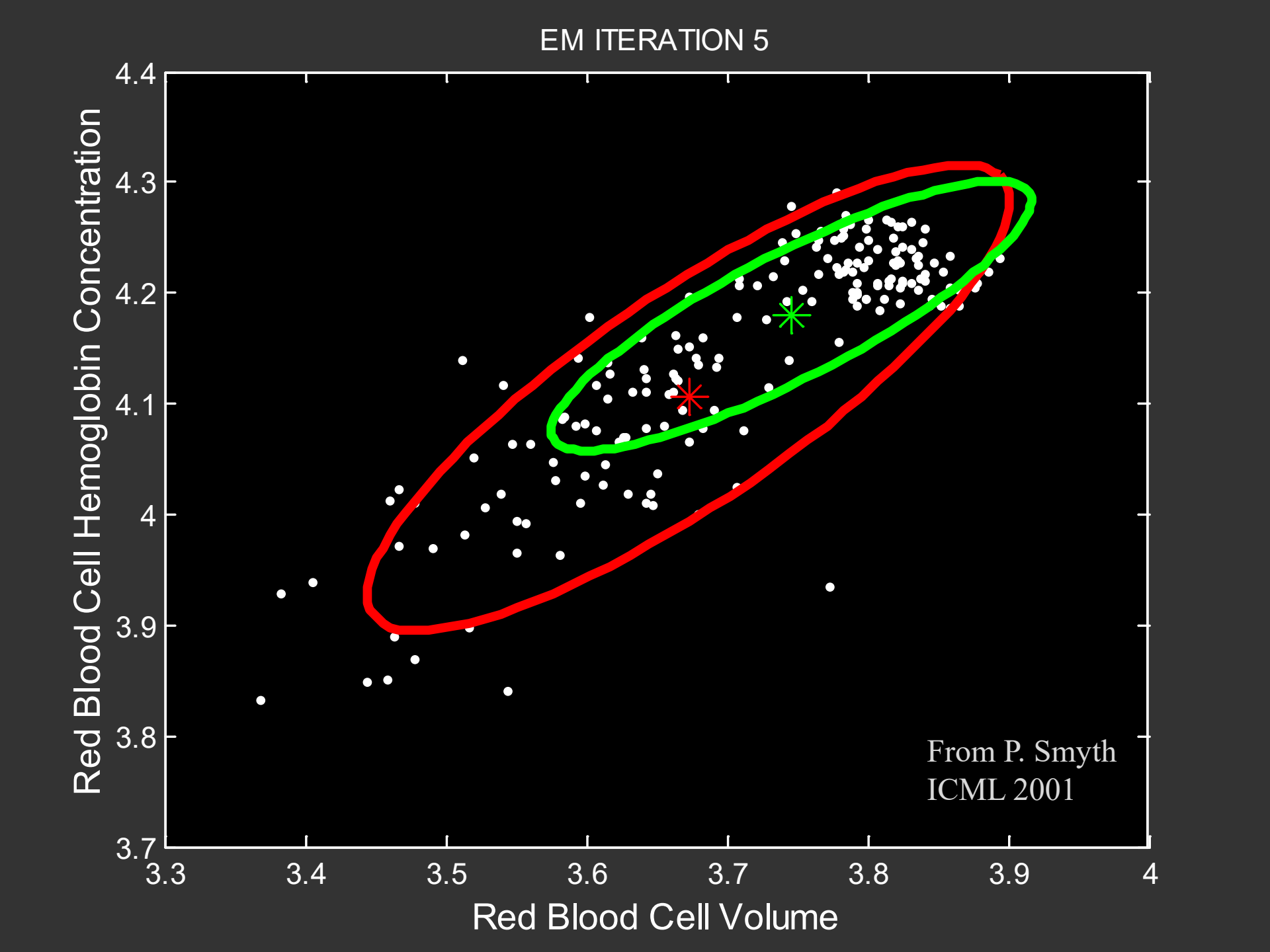
$$\mu_c = \frac{1}{m_c} \sum_i r_{ic} x^{(i)}$$ Weighted mean of assigned data

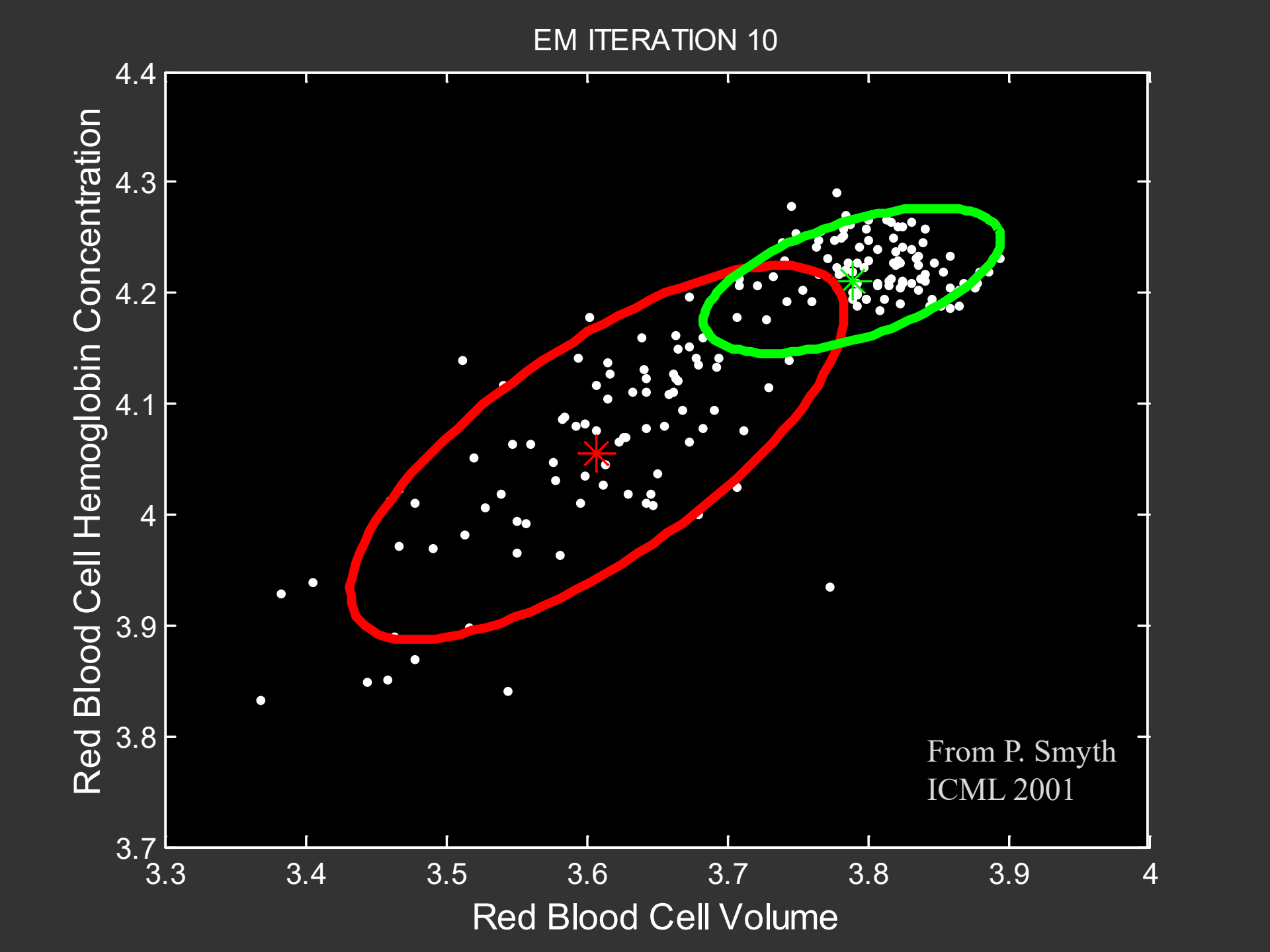
$$\Sigma_c = \frac{1}{m_c} \sum_i r_{ic} (x^{(i)} - \mu_c)^T (x^{(i)} - \mu_c)$$ Weighted covariance of assigned data (use new weighted means here)

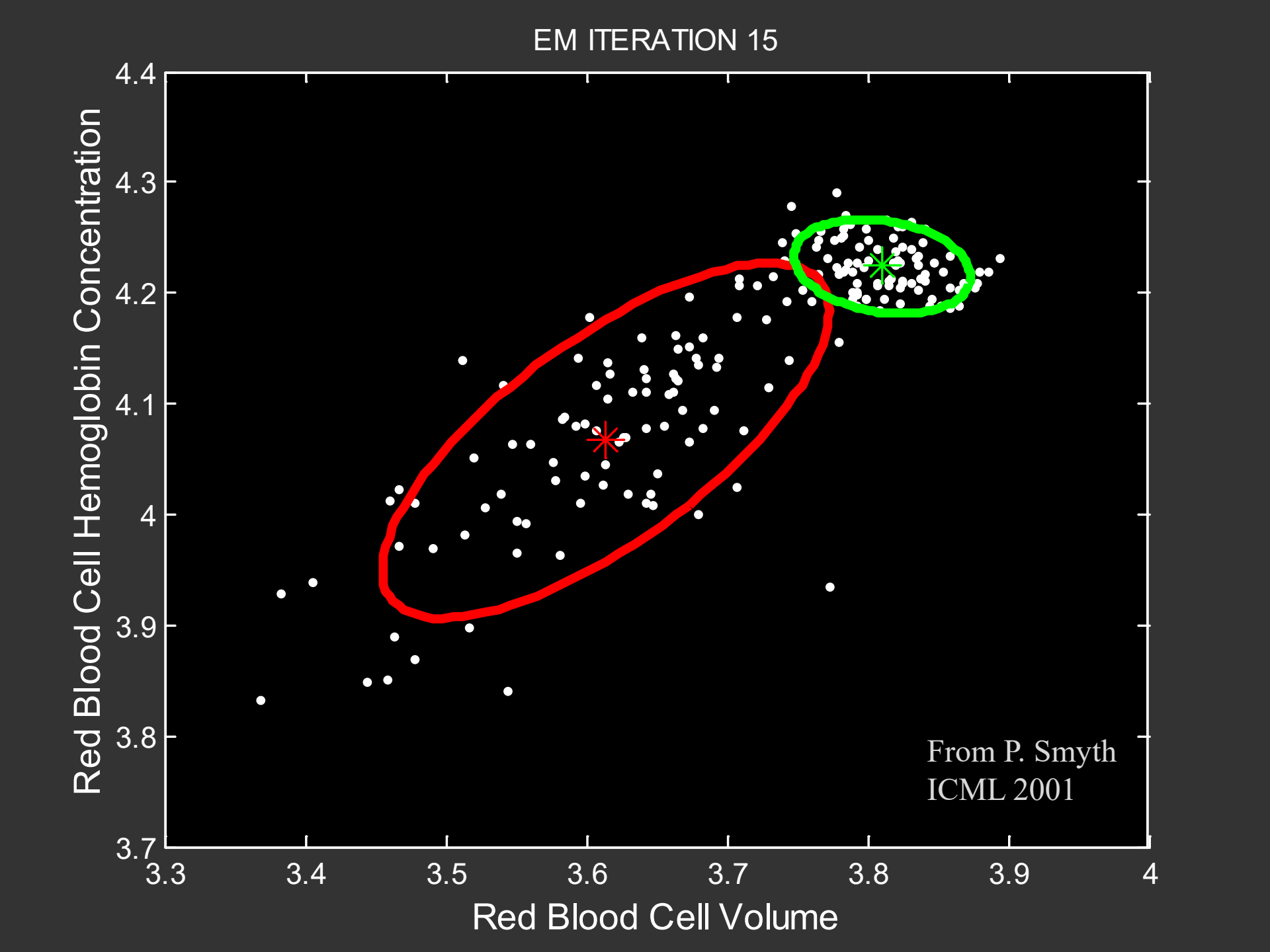

ANEMIA PATIENTS AND CONTROLS
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

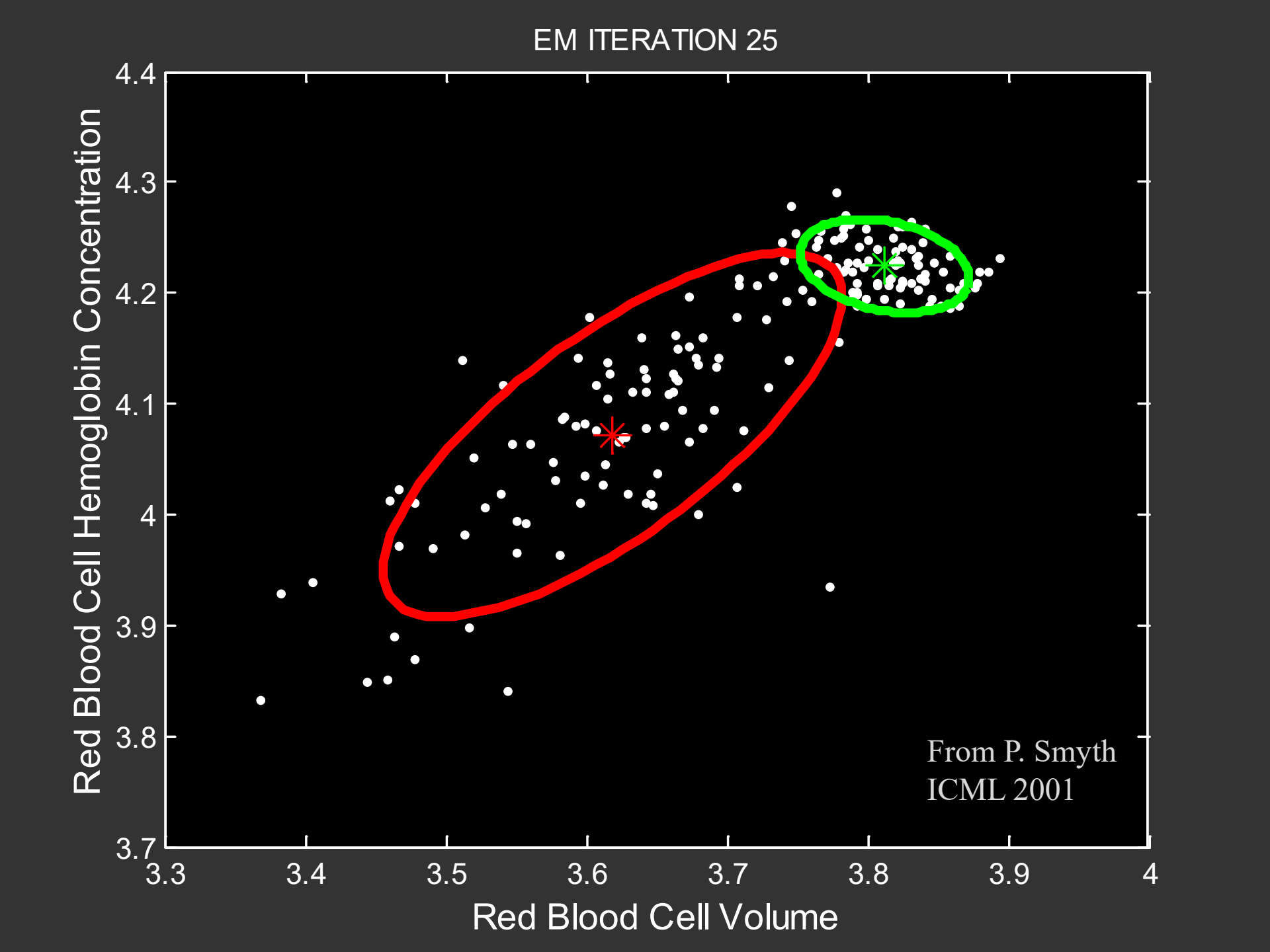

EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001

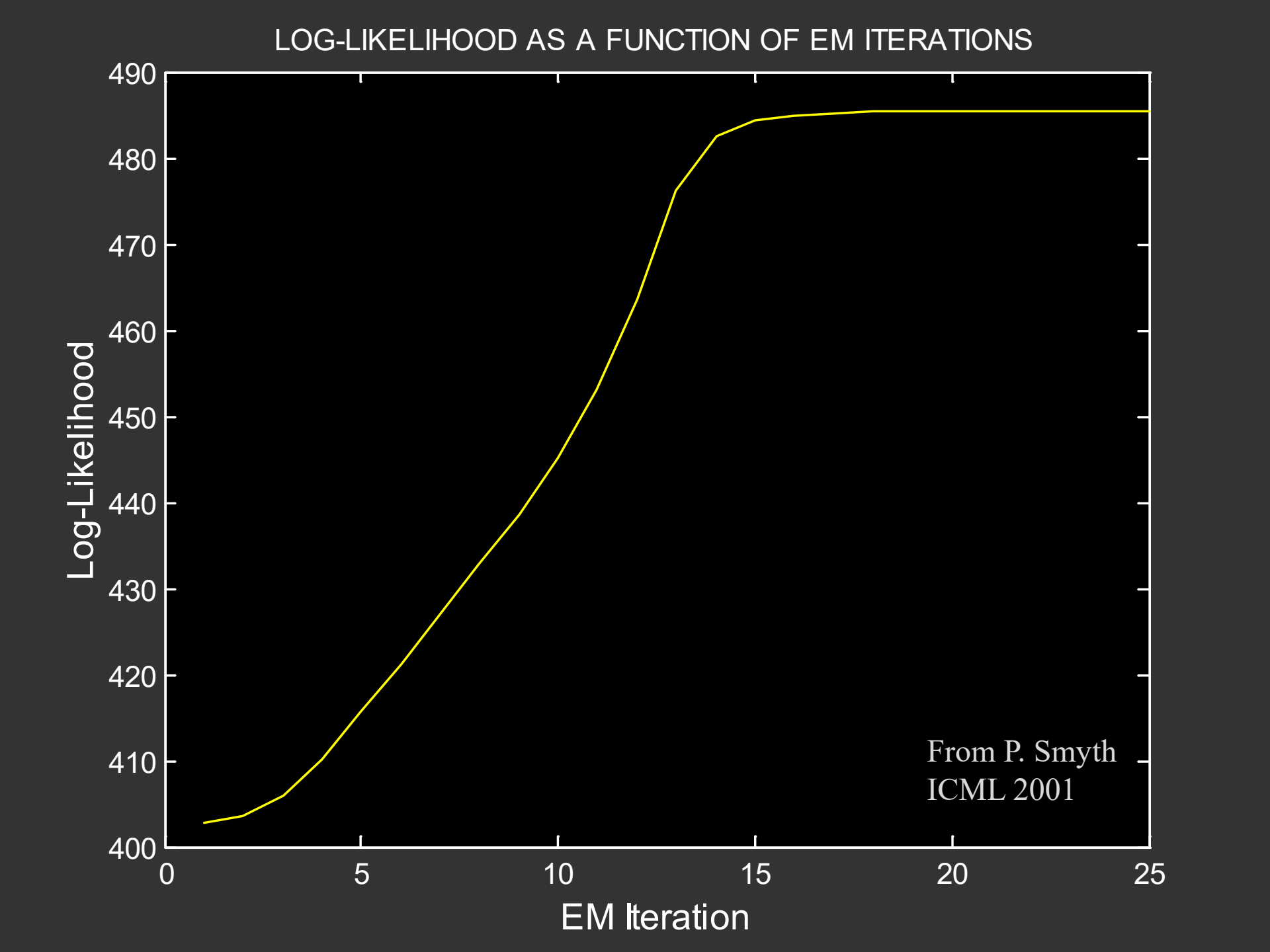

EM ITERATION 3
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 10
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


EM ITERATION 25
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
From P. Smyth
ICML 2001


LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25
From P. Smyth
ICML 2001

EM and missing data

- EM is a general framework for partially observed data
  - “Complete data” xi, zi – features and assignments
  - Assignments zi are missing (unobserved)

- EM corresponds to
  - Computing the distribution over all zi given the parameters
  - Maximizing the “expected complete” log likelihood
  - GMMs = plug in “soft assignments”, but not always so easy

- Alternatives: Stochastic EM, Hard EM
  - Instead of expectations, just sample the zi or choose best (often easier)
  - Called “imputing” the values of z
  - Hard EM: similar to EM, but less “smooth”, more local minima
  - Stochastic EM: similar to EM, but with extra randomness
    - Not obvious when it has converged