Expectation Maximization

Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

April 10th, 2006

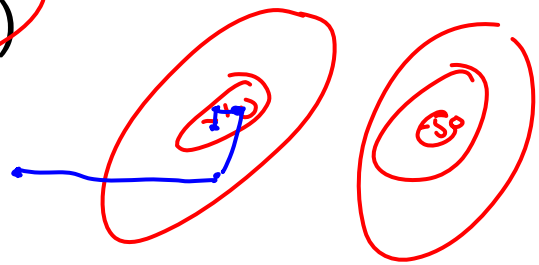
Announcements

- Reminder: Project milestone due Wednesday beginning of class

Coordinate descent algorithms

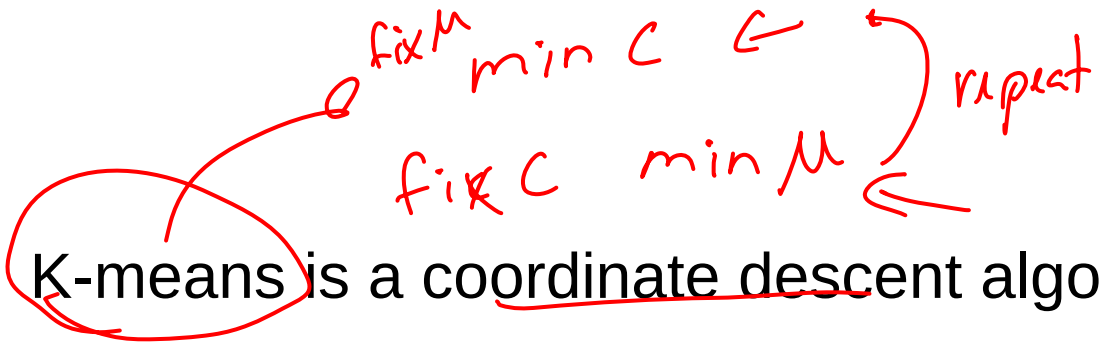
$$\min_{\mu} \min_{C} F(\mu, C) = \min_{\mu} \min_{C} \sum_{i=1}^{k} \sum_{j:C(j)=i} ||\mu_i - x_j||^2$$

- Want: $\min_a \min_b F(a,b)$
- Coordinate descent:
  - fix a, minimize b
  - fix b, minimize a
  - repeat

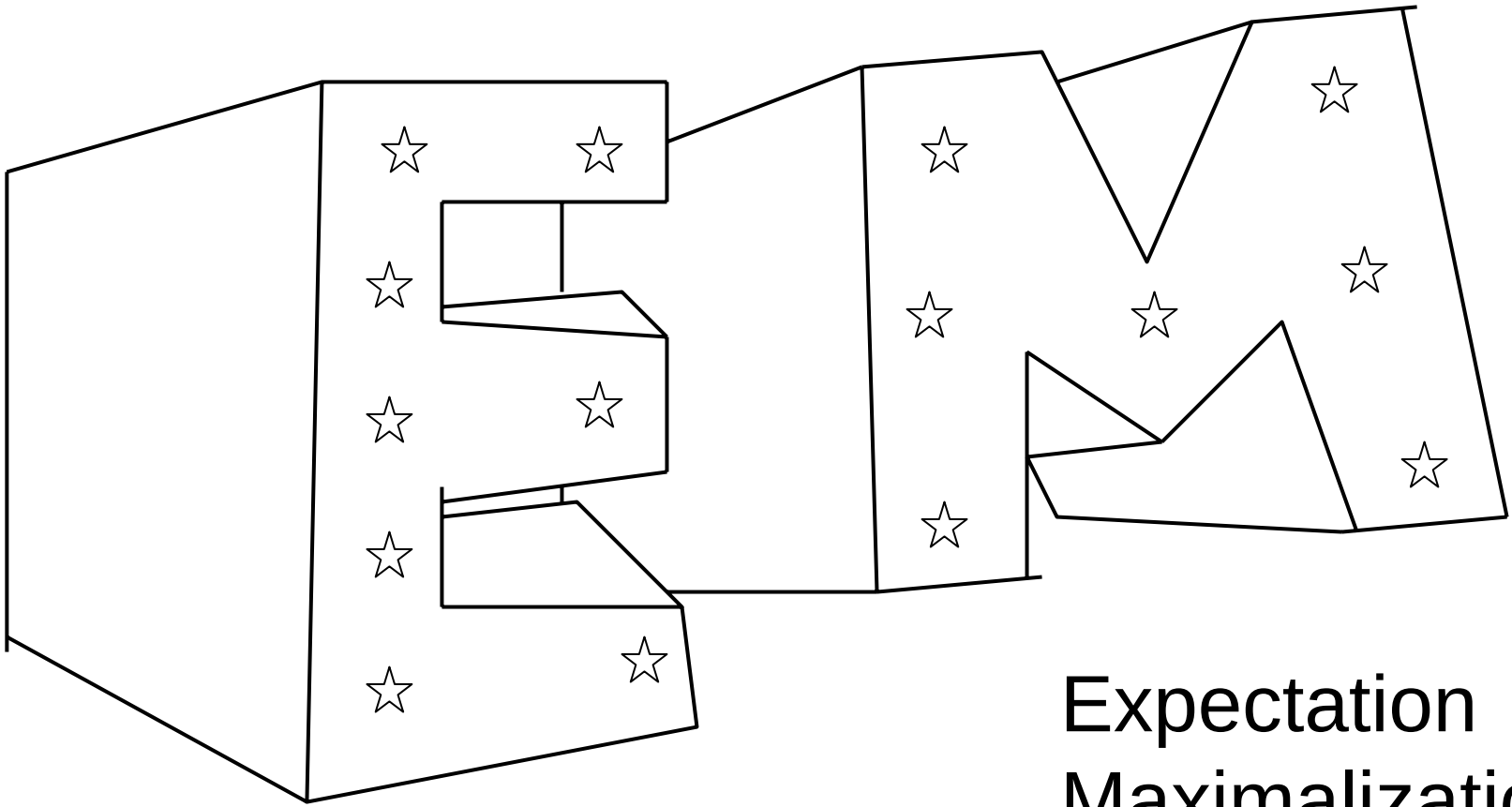


- Converges!!!
  - if F is bounded
  - to a (often good) local optimum
    - as we saw in applet (play with it!)

$\exists K : \forall a, b \quad f(a,b) \geq K$



- K-means is a coordinate descent algorithm!



Expectation
Maximalization

Back to Unsupervised Learning of GMMs – a simple case

Remember:

We have unlabeled data $\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots\ \boldsymbol{x}_m$

We know there are k classes

We know $P(y_1)\ P(y_2)\ P(y_3)\ \ldots\ P(y_k)$

We don't know $\boldsymbol{\mu}_1\ \boldsymbol{\mu}_2\ ..\ \boldsymbol{\mu}_k$

We can write P(data | $\boldsymbol{\mu}_1 \ldots\ \boldsymbol{\mu}_k$)

$$= p(x_1 \ldots x_m | \mu_1 \ldots \mu_k)$$

$$= \prod_{j=1}^{m} p(x_j | \mu_1 \ldots \mu_k)$$

$$= \prod_{j=1}^{m} \sum_{i=1}^{k} p(x_j | \mu_i) P(y = i)$$

$$\propto \prod_{j=1}^{m} \sum_{i=1}^{k} \exp\left(-\frac{1}{2\sigma^2} \| x_j - \mu_i \|^2 \right) P(y = i)$$

EM for simple case of GMMs: The E-step

- If we know $\mu_1,\ldots,\mu_k$ $\rightarrow$ easily compute prob. point x_j belongs to class y=i

prob. x_j comes from $y=i$

$$p(y=i|x_j,\mu_1\ldots\mu_k) \propto \exp\left(-\frac{1}{2\sigma^2}\|x_j-\mu_i\|^2\right)P(y=i)$$

[same as classification]

$p(y=1|x_j,\mu) \propto 0.3$

$p(y=0|x_j,\mu) \propto 0.2$

normalize $\Rightarrow = 0.6$, $= 0.4$

repeat for each x_j.

EM for simple case of GMMs: The M-step

- If we know prob. point x_j belongs to class y=i

  $\rightarrow$ MLE for μ_i is weighted average

  - imagine k copies of each x_j, each with weight $P(y=i|x_j)$:

$$\mu_i = \frac{\sum_{j=1}^{m} P(y=i|x_j)\, x_j}{\sum_{j=1}^{m} P(y=i|x_j)}$$

weighted version

x	P(y=0|x)
1	0.8
2	.0.2
3	0.1

$$\mu_0 = \frac{0.8 \times 1 + 0.2 \times 2 + 0.1 \times 3}{0.8 + .2 + .1}$$

if we knew x_j comes from $C(j)$

$$\mu_i = \frac{\sum_{j:C(j)=i} x_j}{\sum_{j:C(j)=i} 1}$$

E.M. for GMMs

E-step

Compute "expected" classes of all datapoints for each class

$$p(y=i|x_j,\mu_1...\mu_k) \propto \exp\left(-\frac{1}{2\sigma^2}\|x_j-\mu_i\|^2\right)P(y=i)$$

↙ prior

Just evaluate a Gaussian at x_j

M-step

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i = \frac{\sum_{j=1}^{m} P(y=i|x_j)\,x_j}{\sum_{j=1}^{m} P(y=i|x_j)}$$

weighted data using P(y|x)

E.M. for General GMMs

Iterate. On the *t*'th iteration let our estimates be

$$\lambda_t = \{ \mu_1^{(t)}, \mu_2^{(t)} \ldots \mu_k^{(t)}, \Sigma_1^{(t)}, \Sigma_2^{(t)} \ldots \Sigma_k^{(t)}, p_1^{(t)}, p_2^{(t)} \ldots p_k^{(t)} \}$$

means covariance priors

$p_i^{(t)}$ is shorthand for estimate of $P(y=i)$ on t'th iteration

solution depends on starting λ_0 use random restarts

E-step

Compute "expected" classes of all datapoints for each class

$$P(y = i | x_j, \lambda_t) \propto p_i^{(t)} p(x_j | \mu_i^{(t)}, \Sigma_i^{(t)})$$

Just evaluate a Gaussian at x_j

M-step

Compute Max. like **μ** given our data's class membership distributions

$$\mu_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t) x_j}{\sum_j P(y = i | x_j, \lambda_t)}$$

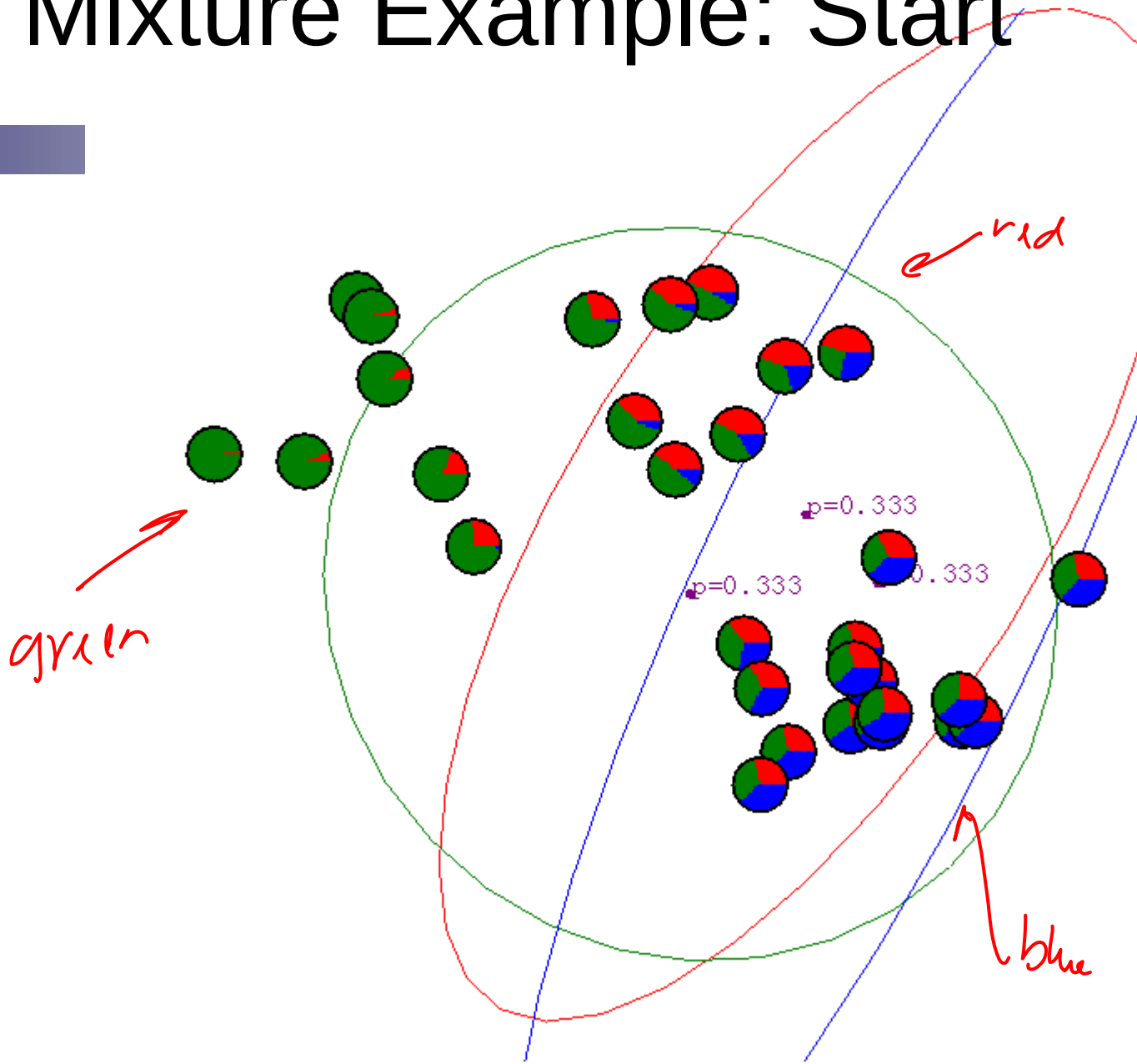
$$\Sigma_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t) [x_j - \mu_i^{(t+1)}][x_j - \mu_i^{(t+1)}]^T}{\sum_j P(y = i | x_j, \lambda_t)}$$

weighted data

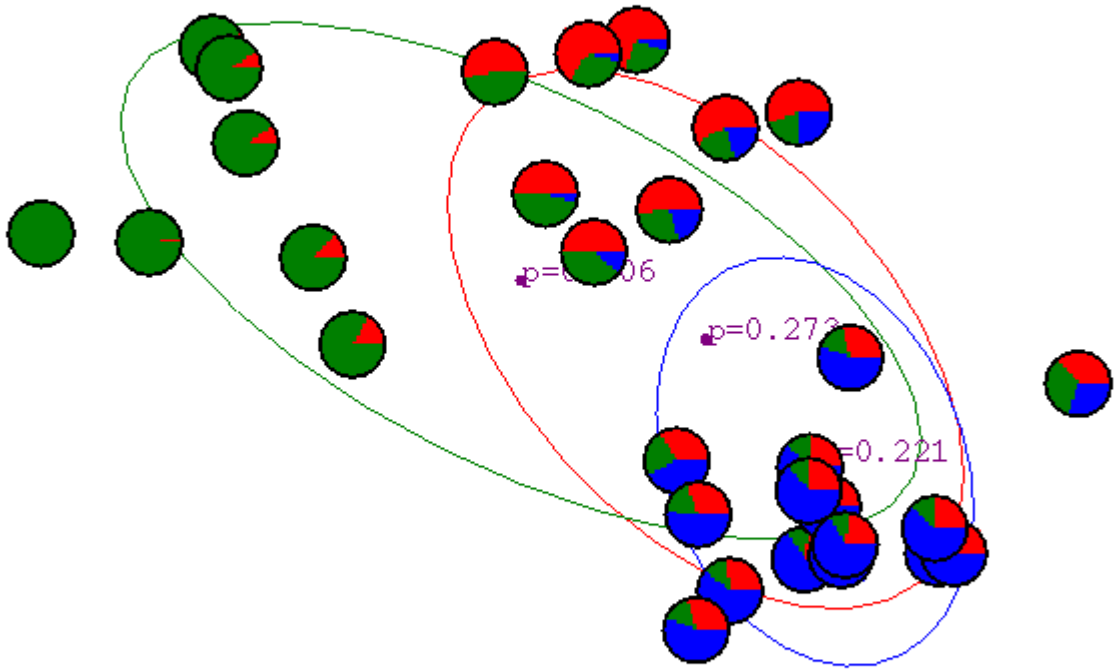
$$p_i^{(t+1)} = \frac{\sum_j P(y = i | x_j, \lambda_t)}{m}$$

m = #records

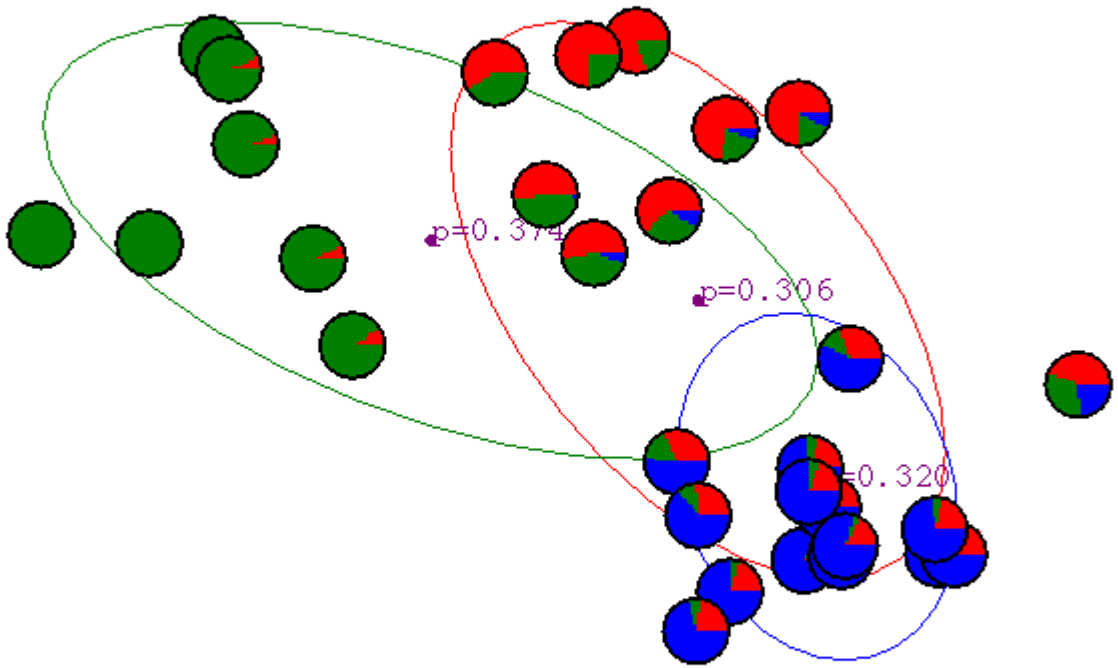
Gaussian Mixture Example: Start



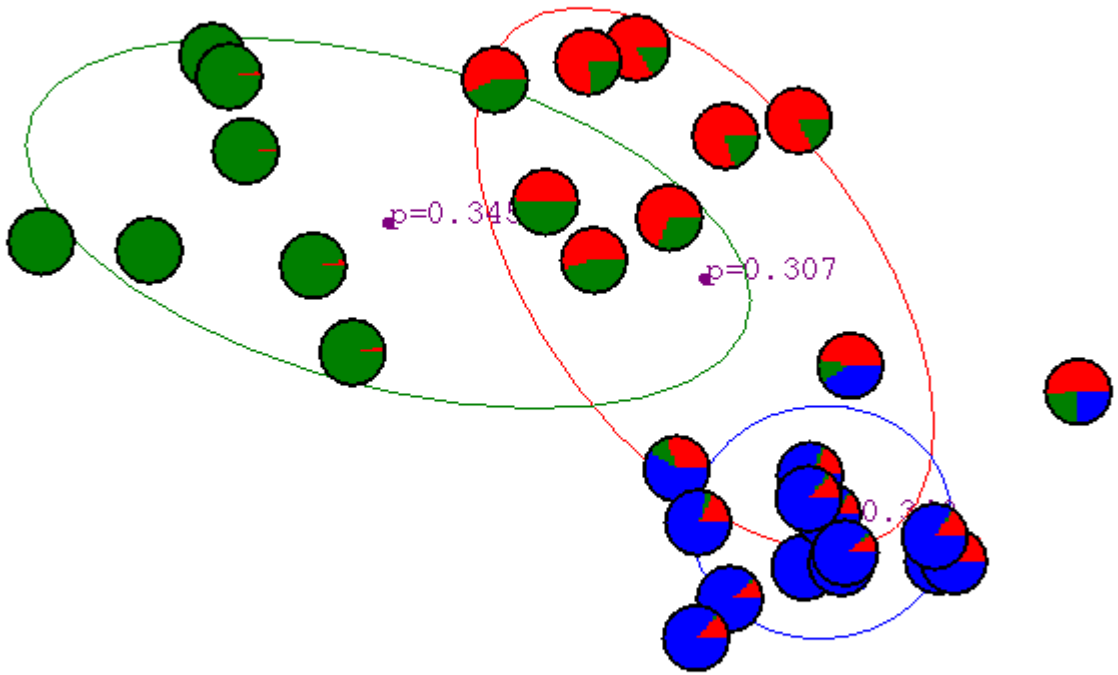
After first iteration



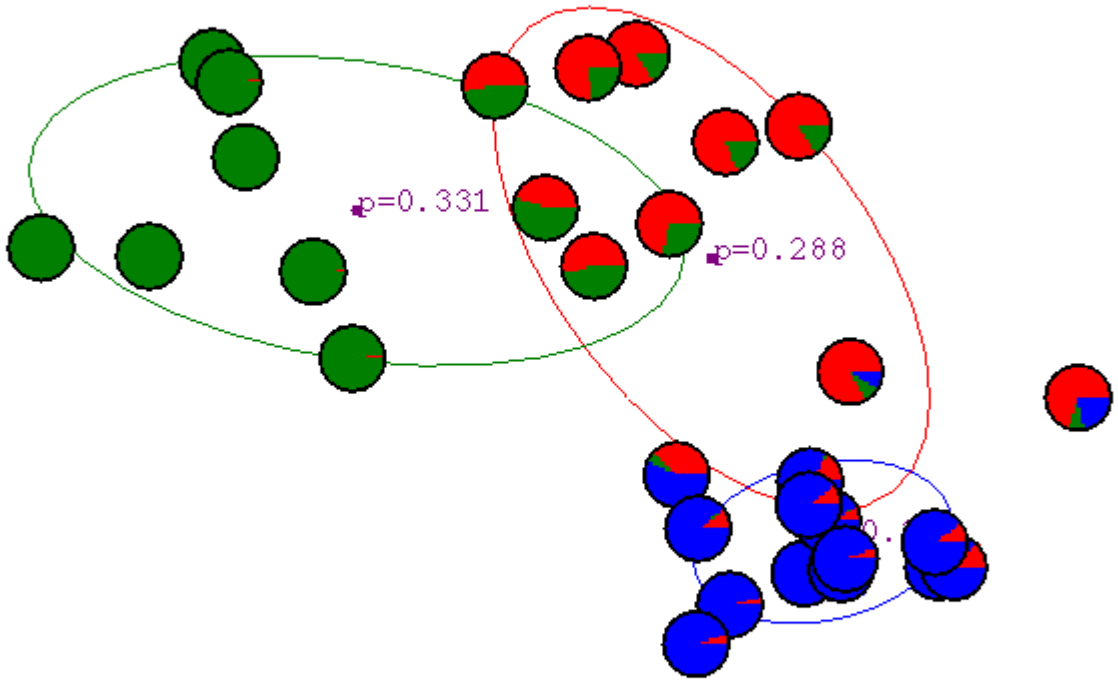
After 2nd iteration



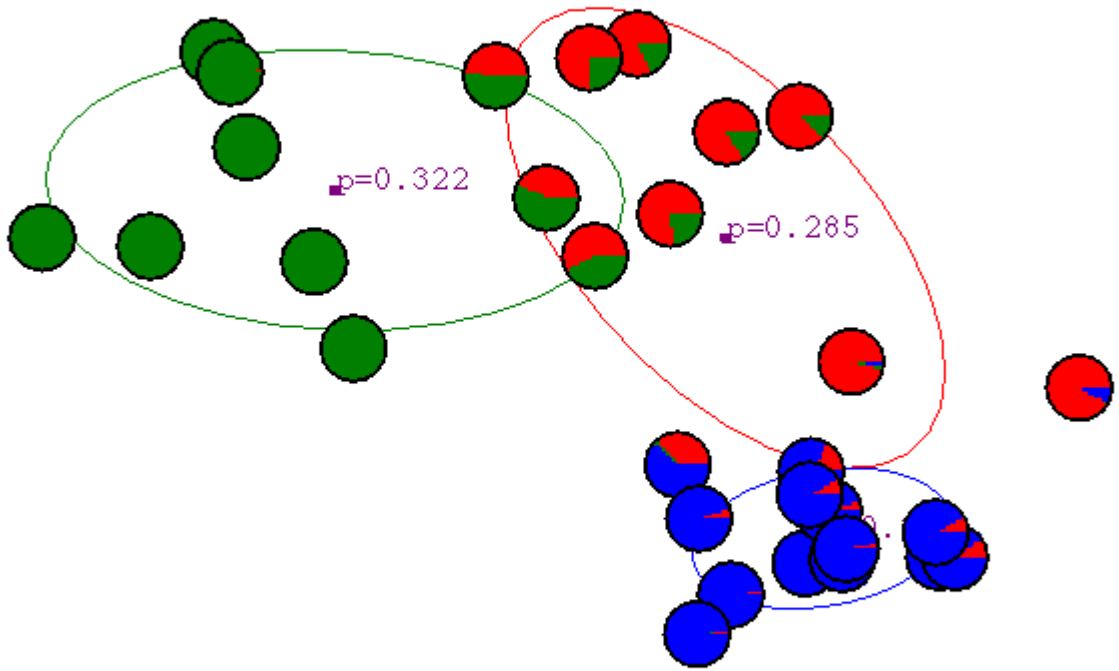
After 3rd iteration



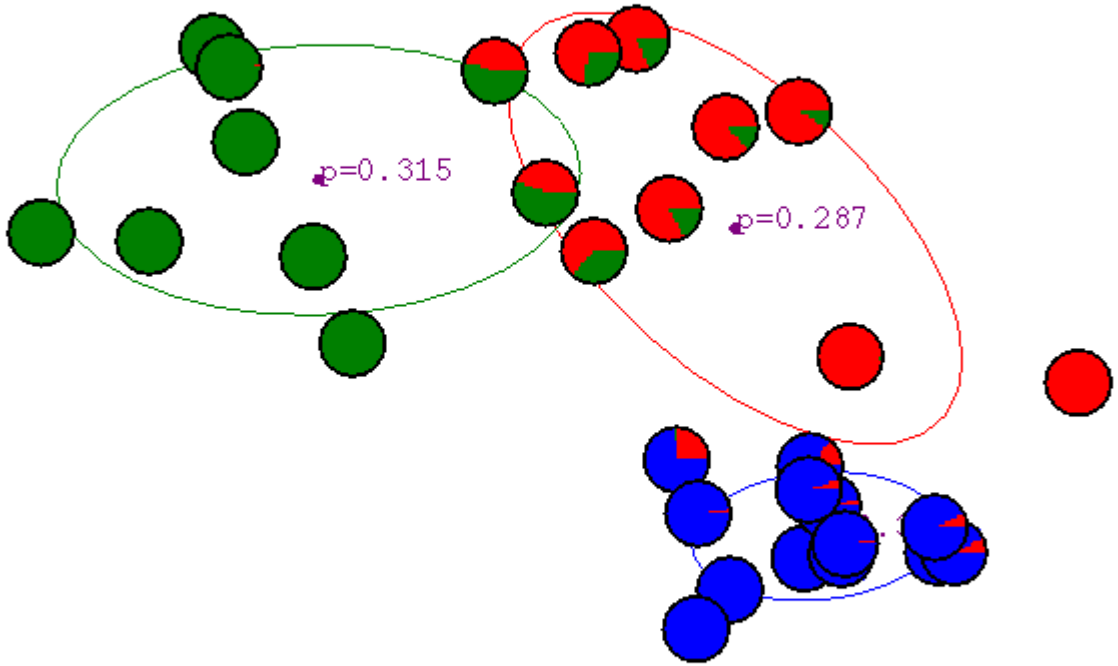
After 4th iteration



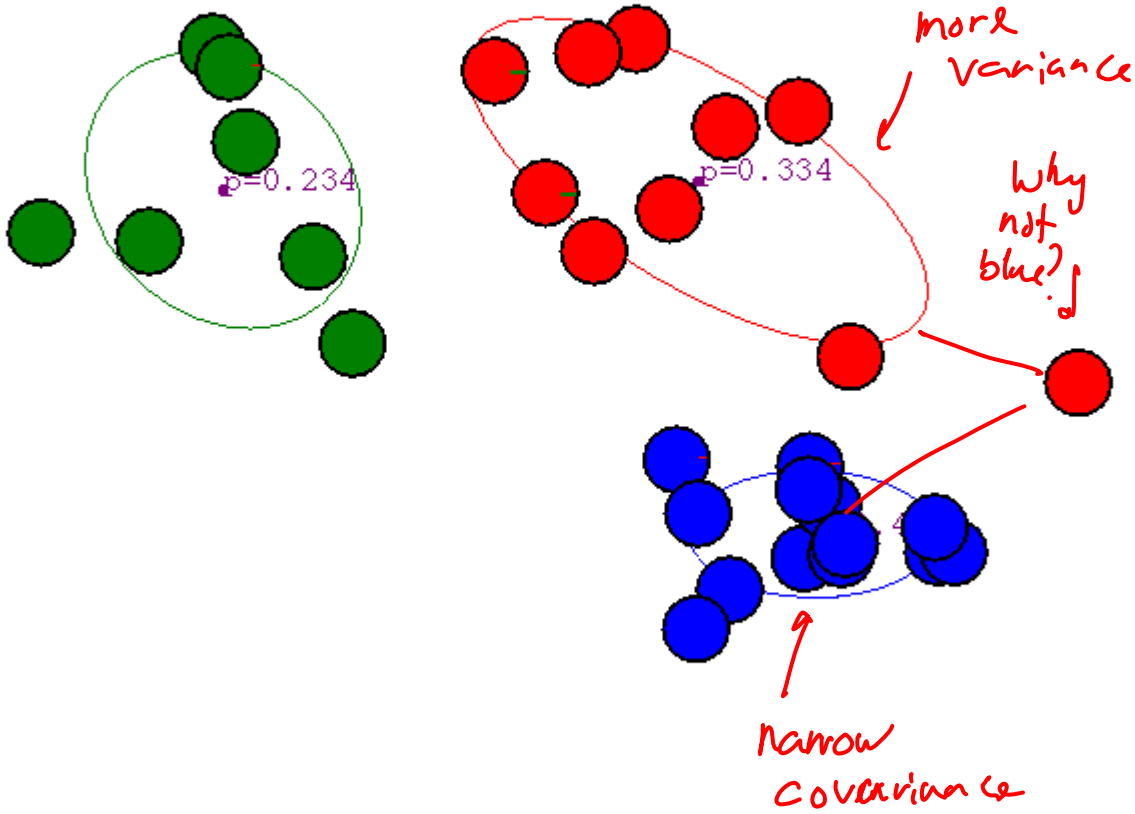
After 5th iteration



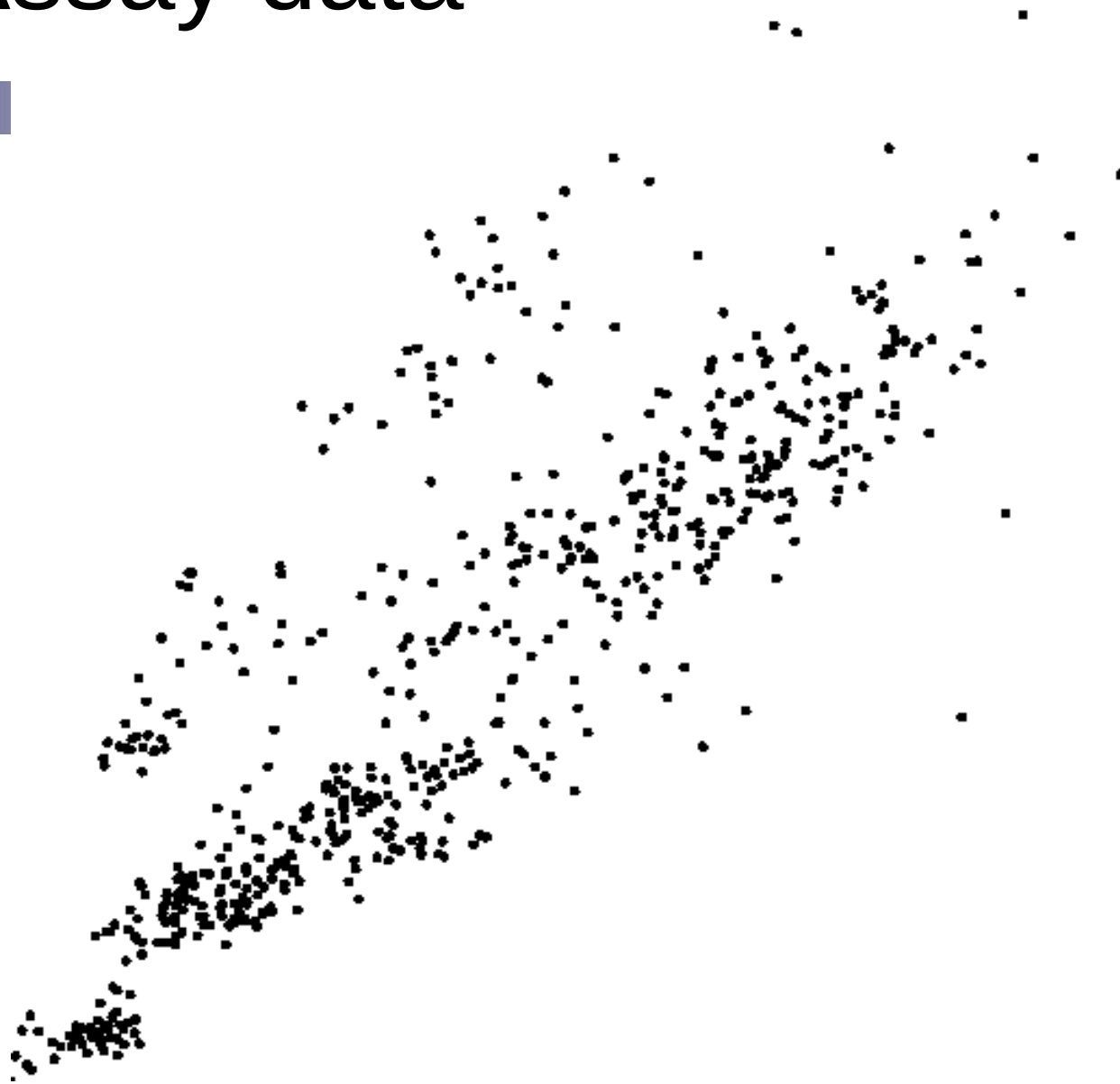
After 6th iteration



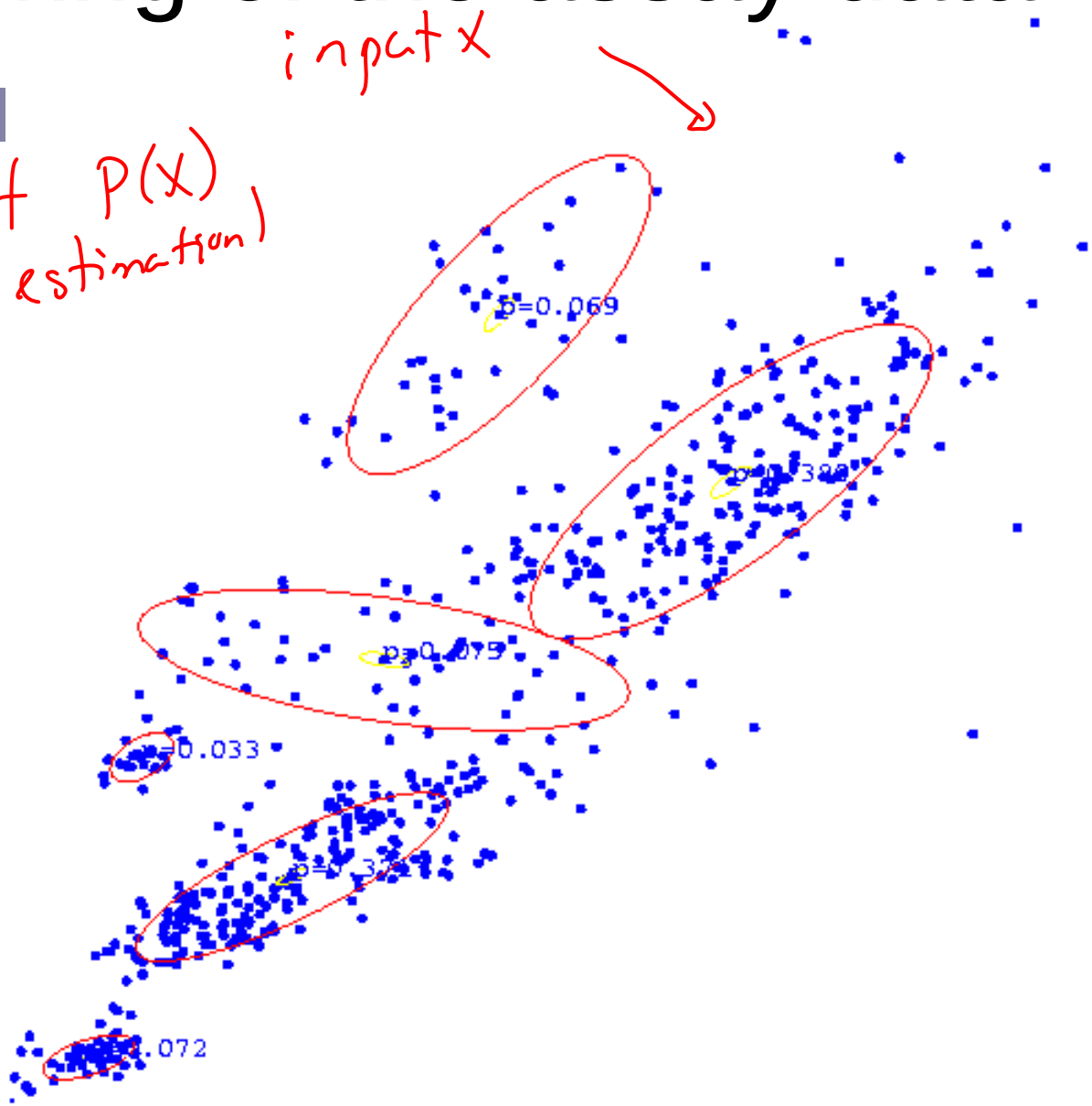
After 20th iteration



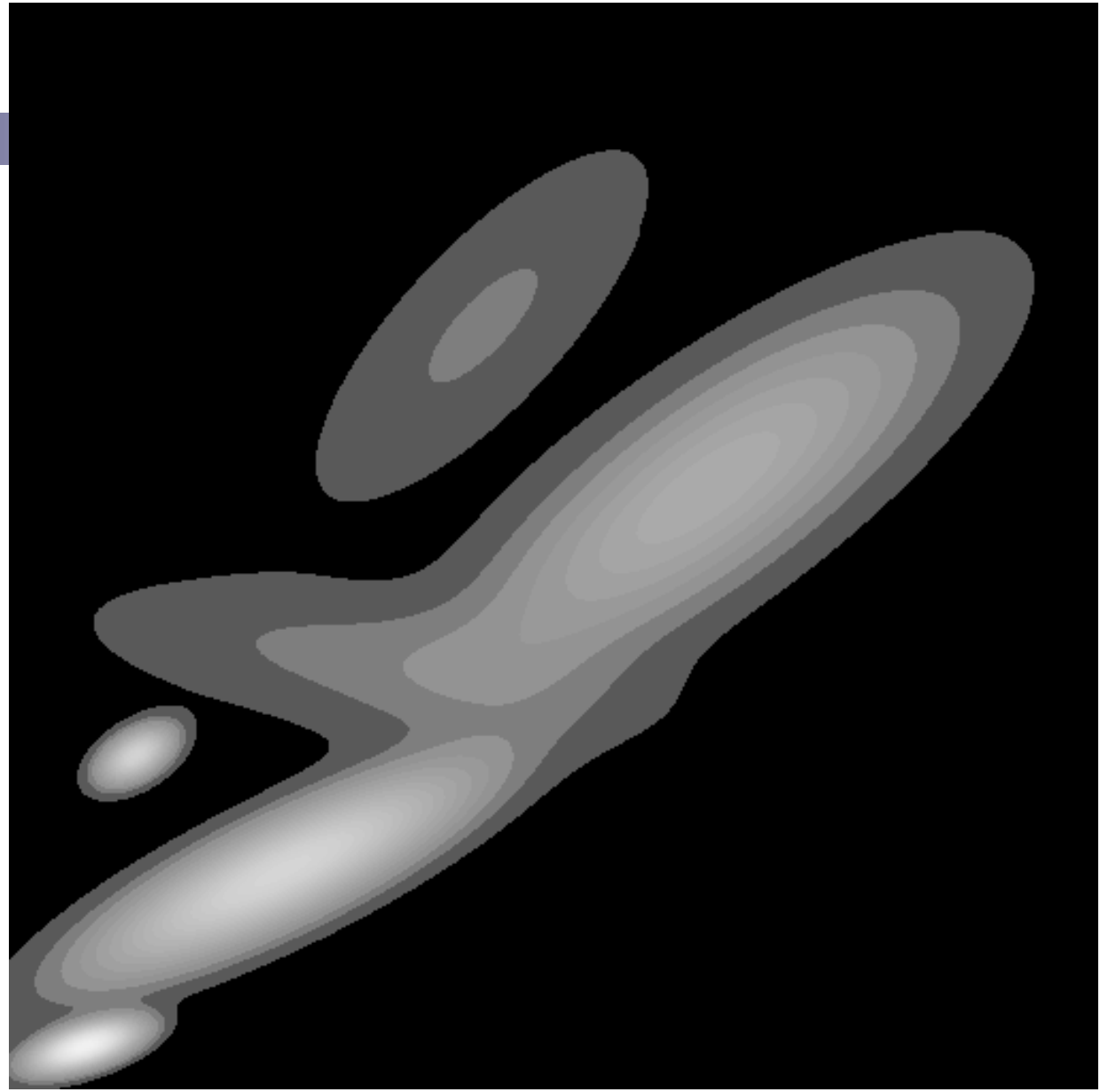
Some Bio Assay data



GMM clustering of the assay data

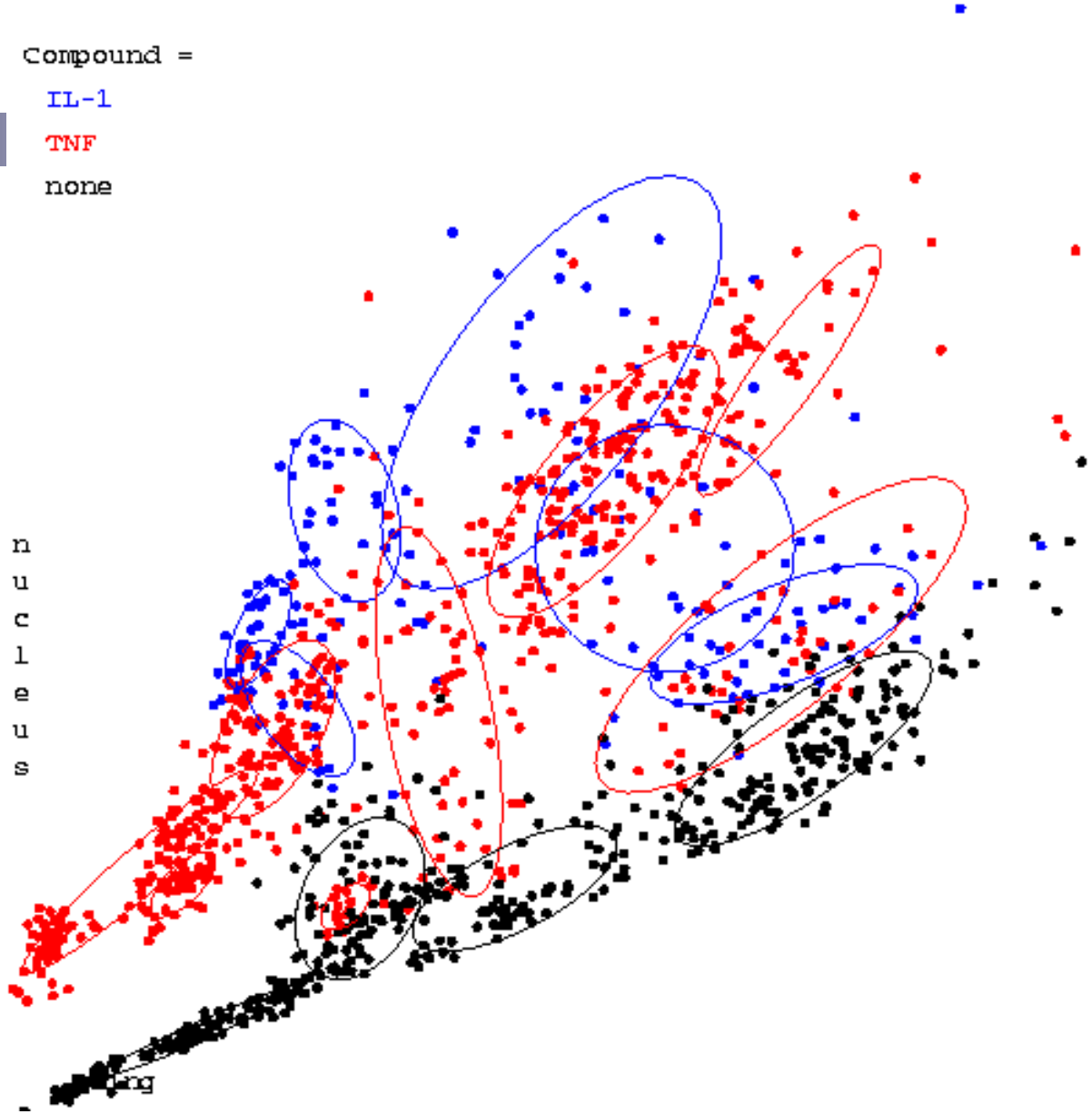


Resulting Density Estimator

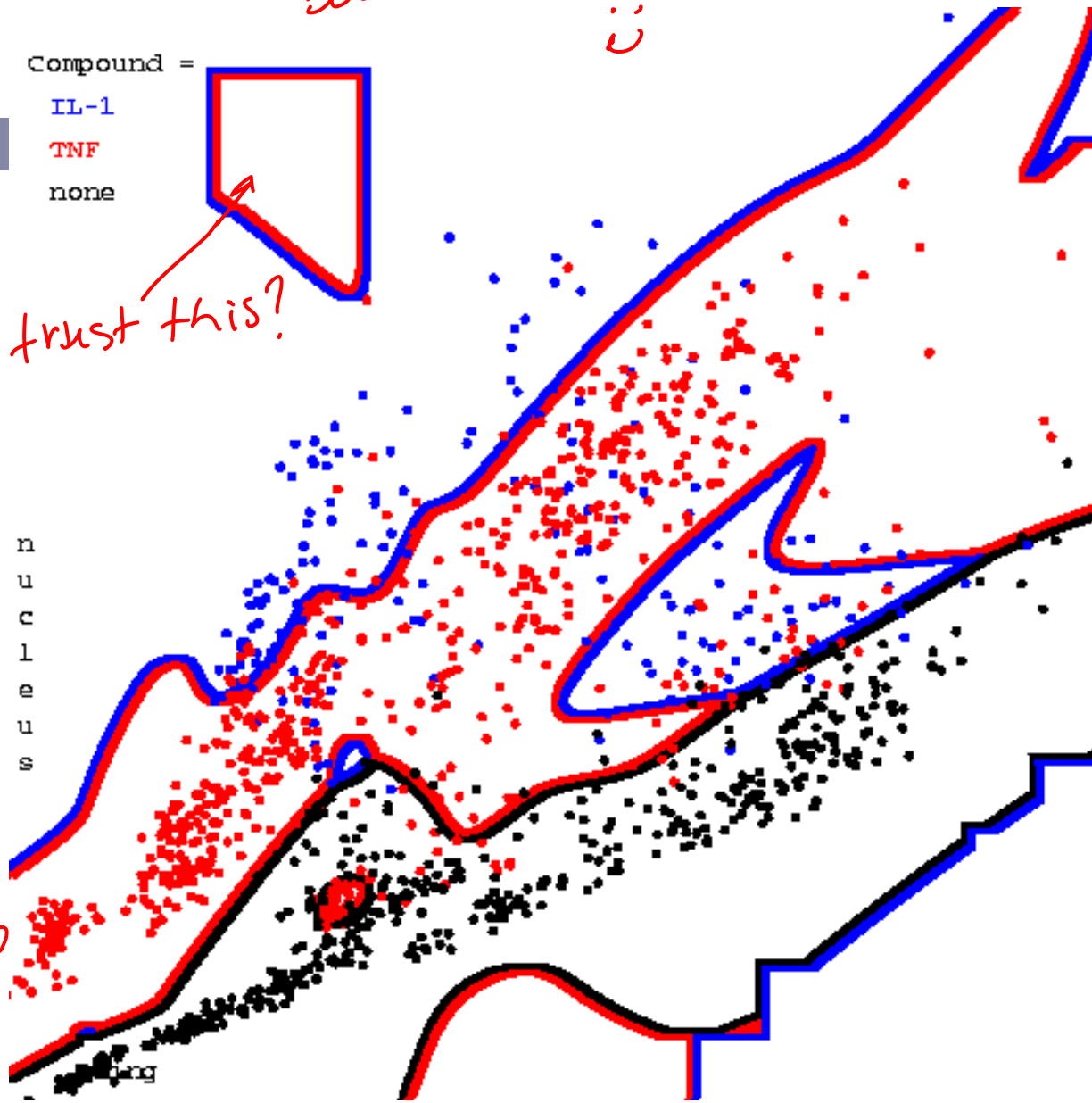


Three classes of assay

(each learned with it's own mixture model)



Resulting Bayes Classifier

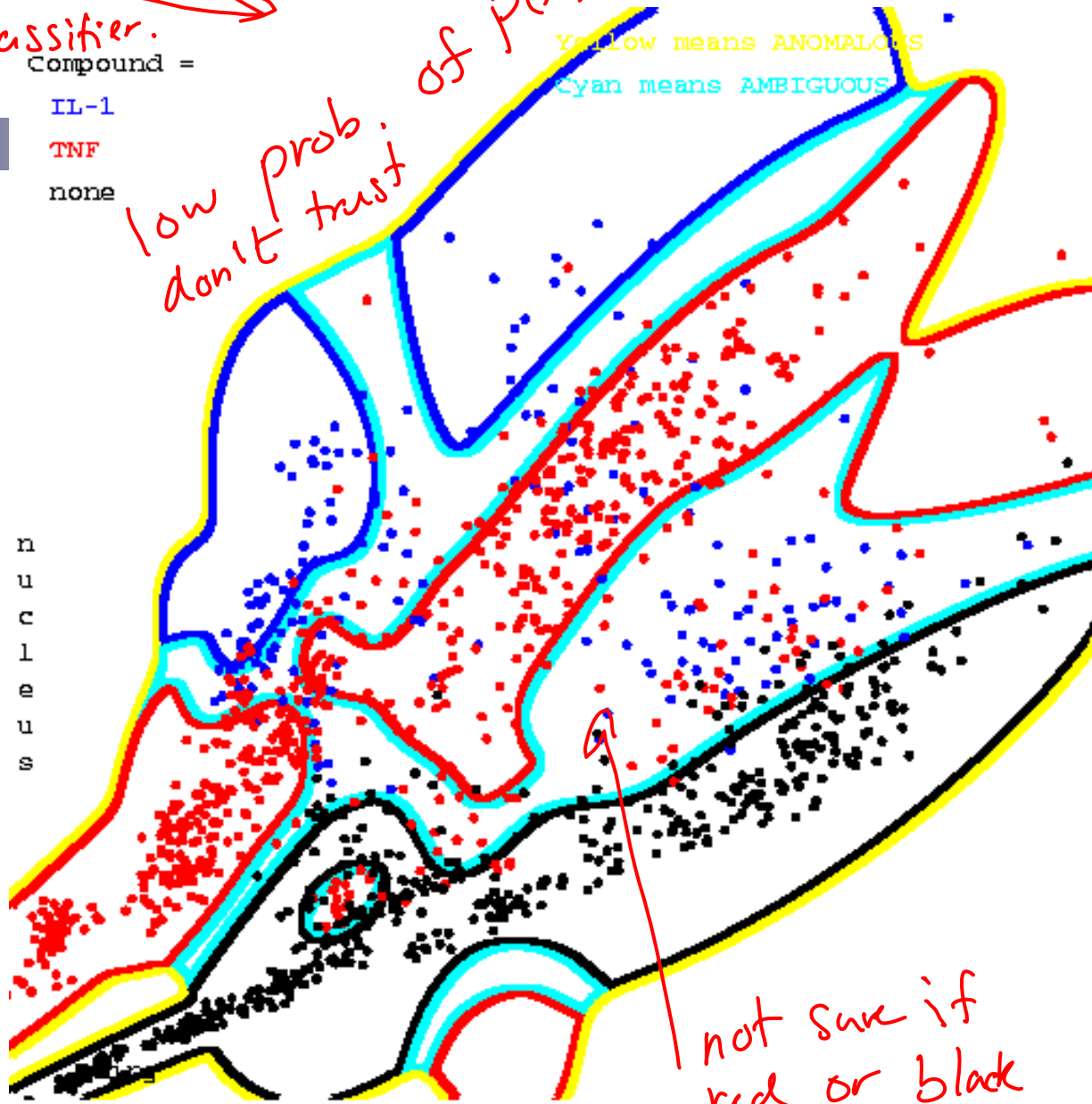


Can't do this easily with discriminative classifier.

Resulting Bayes Classifier, using posterior probabilities to alert about ambiguity and anomalousness

Yellow means anomalous

Cyan means ambiguous



The general learning problem with missing data

more general than GMMs

- Marginal likelihood – **x** is observed, **z** is missing:

$$\ell(\theta : \mathcal{D}) = \log \prod_{j=1}^{m} P(\mathbf{x}_j \mid \theta) \qquad \text{(i.i.d.)}$$

$$= \sum_{j=1}^{m} \log P(\mathbf{x}_j \mid \theta)$$

$$= \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{x}_j, \mathbf{z} \mid \theta)$$

ℓ = marginal likelihood of data

want to $\operatorname{argmax}_{\theta} \ell(\theta : \mathcal{D})$

E-step

- **x** is observed, **z** is missing
- Compute probability of missing data given current choice of θ
  - Q(**z**|$\mathbf{x}_j$) for each $\mathbf{x}_j$
    - e.g., probability computed during classification step
    - corresponds to "classification step" in K-means

$$Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) = P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})$$

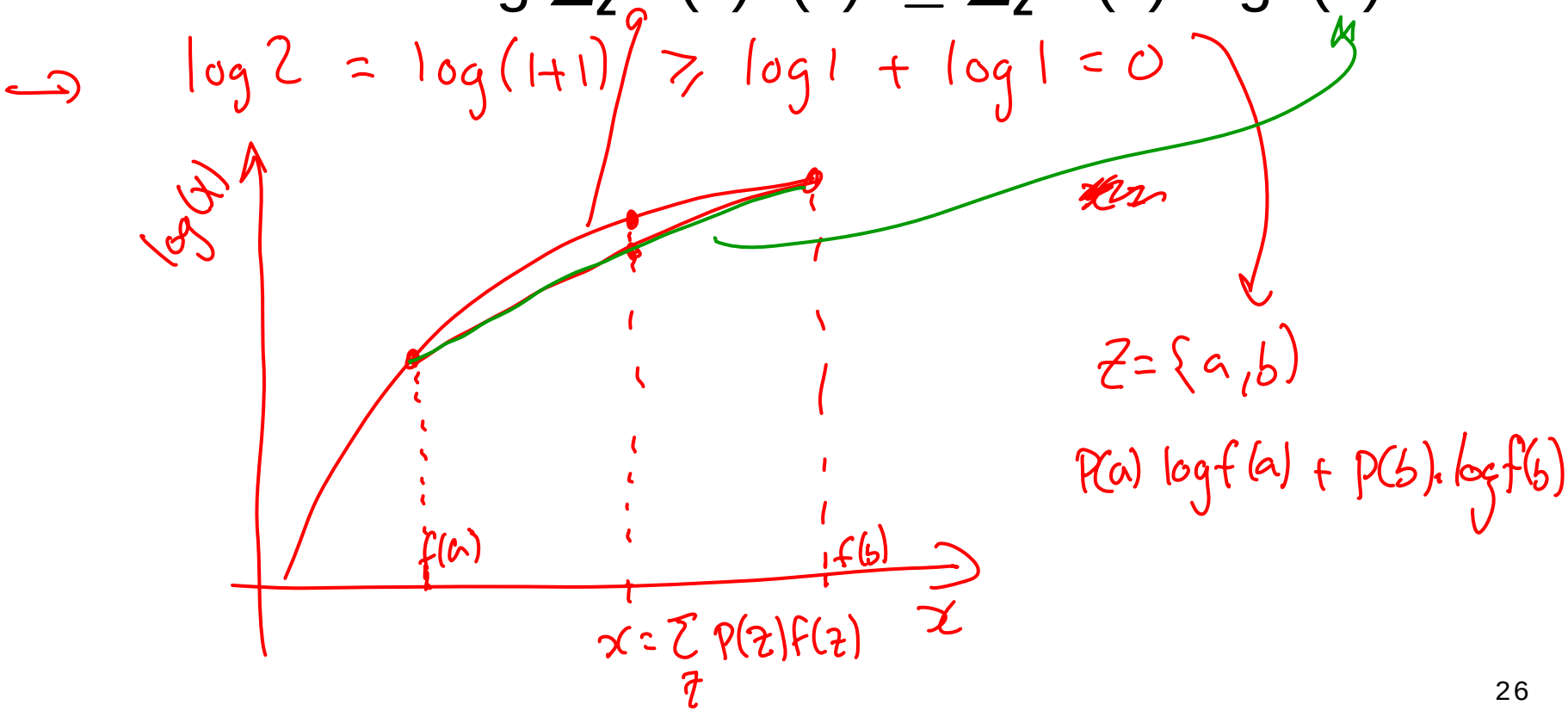
iteration t of EM

z is an assignment to hidden vars.
→ example: z = (y=1)

Jensen's inequality

$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log \sum_{\mathbf{z}} P(\mathbf{z} \mid \mathbf{x}_j) P(\mathbf{x}_j \mid \theta)$$

- **Theorem**: $\log \sum_{\mathbf{z}} P(\mathbf{z})\, f(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log f(\mathbf{z})$



Applying Jensen's inequality

- Use: $\log \sum_z P(z)\, f(z) \geq \sum_z P(z) \log f(z)$

$$\ell(\theta^{(t)} : \mathcal{D}) = \sum_{j=1}^{m} \log \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)})}{Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j)}$$

$$\geq \sum_{j=1}^{m} \sum_{z} Q^{(t+1)}(z \mid x_j) \cdot \log \frac{P(z, x_j \mid \theta^{(t)})}{Q^{(t+1)}(z \mid x_j)}$$

$$= \sum_{j=1}^{m} \sum_{z} Q^{(t+1)}(z \mid x_j) \log P(z, x_j \mid \theta^{(t)}) - \sum_{j=1}^{m} \sum_{z} Q^{(t+1)}(z \mid x_j) \log Q^{(t+1)}(z \mid x_j)$$

$$m \cdot \hat{H}_{Q^{(t+1)}}(Z \mid X) \qquad H(Q^{(t+1)})$$

$H(x) = -\sum_x p(x) \log p(x)$

$\ell(\theta : D) = \sum_{j=1}^{m} \log \sum_z P(z, x_j \mid \theta)$

$\log \frac{a}{b} = \log a - \log b$

The M-step maximizes lower bound on weighted data

- Lower bound from Jensen's:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)}) + H(Q^{(t+1)})$$

likelihood all data

doesn't depend on θ

fix Q, max over θ ⇒ MLE for weighted data

irrelevant

- Corresponds to weighted dataset:
  - $\langle \mathbf{x}_1, \mathbf{z}=1 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}_1)$
  - $\langle \mathbf{x}_1, \mathbf{z}=2 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}_1)$
  - $\langle \mathbf{x}_1, \mathbf{z}=3 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}_1)$
  - $\langle \mathbf{x}_2, \mathbf{z}=1 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}_2)$
  - $\langle \mathbf{x}_2, \mathbf{z}=2 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}_2)$
  - $\langle \mathbf{x}_2, \mathbf{z}=3 \rangle$ with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}_2)$
  - …

The M-step

doesn't play in M-step

$$\ell(\theta^{(t)} : \mathcal{D}) \geq \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)}) + H(Q^{(t+1)})$$

- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta)$$

same as fully observed data weighted by $Q^{(t+1)}$

- Use expected counts instead of counts:
  - If learning requires Count(**x**,**z**)
  - Use $E_{Q(t+1)}$[Count(**x**,**z**)] [weighted counts]

Convergence of EM

- Define potential function F(θ,Q):

$$\ell(\theta : \mathcal{D}) \;\geq\; F(\theta, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

- EM corresponds to coordinate ascent on F
  - Thus, maximizes lower bound on marginal log likelihood

fix Q → max over θ

fix θ → max over Q

will see soon

M-step is easy

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta)$$

- Using potential function

doesn't change in M step

$$F(\theta, Q^{(t+1)}) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) + H(Q^{(t+1)})$$

$\theta^{(t+1)}$ maximizes this part

$$\Rightarrow F(\theta^{(t+1)}, Q^{(t+1)}) \geq F(\theta^{(t)}, Q^{(t+1)})$$

Fixed $Q^{(t+1)}$, improved over θ

E-step also doesn't decrease potential function 1

- Fixing θ to $\theta^{(t)}$:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

$\log a \cdot b = \log a + \log b$

$P(z, x_j \mid \theta^{(t)}) = P(z \mid x_j, \theta^{(t)}) \cdot P(x_j \mid \theta^{(t)})$

$$= \sum_{j=1}^{m} \sum_{z} Q(z \mid x_j) \log \frac{P(z \mid x_j, \theta^{(t)}) \cdot P(x_j \mid \theta^{(t)})}{Q(z \mid x_j)}$$

$$= \sum_{j=1}^{m} \sum_{z} Q(z \mid x_j) \log \frac{P(z \mid x_j, \theta^{(t)})}{Q(z \mid x_j)} + \sum_{j=1}^{m} \sum_{z} Q(z \mid x_j) \log P(x_j \mid \theta^{(t)})$$

($P(x_j \mid \theta^{(t)})$ doesn't depend on z)

$$= \sum_{j=1}^{m} \log P(x_j \mid \theta^{(t)}) \cdot \sum_{z} Q(z \mid x_j)$$

($\sum_z Q(z \mid x_j) = 1$)

$$= \sum_{j=1}^{m} \log P(x_j \mid \theta^{(t)}) = \ell(\theta^{(t)} : \mathcal{D})$$

KL-divergence

$\log \frac{Q}{P} = -\log \frac{P}{Q}$

- Measures distance between distributions

$$KL(Q||P) = \sum_z Q(z) \log \frac{Q(z)}{P(z)}$$

if Q is "far" from P ⇒ KL(Q||P) large (very positive)

if Q is "close" to P ⇒ KL(Q||P) low

KL always ≥ 0

- KL=zero if and only if Q=P

E-step also doesn't decrease potential function 2

- Fixing θ to $\theta^{(t)}$:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) = \ell(\theta^{(t)} : \mathcal{D}) + \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

$$\rightarrow = \ell(\theta^{(t)} : \mathcal{D}) - \sum_{j=1}^{m} KL\left(Q(\mathbf{z} \mid \mathbf{x}_j) || P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\right)$$

doesn't depend on Q

want to maximize

$\Rightarrow$ set KL = 0

$\Rightarrow Q(z|x_j) = p(z|x_j, \theta^{(t)})$

E-step also doesn't decrease potential function 3

$$\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) \;=\; \ell(\theta^{(t)} : \mathcal{D}) - \sum_{j=1}^{m} KL\left(Q(\mathbf{z} \mid \mathbf{x}_j) || P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\right)$$

- Fixing θ to $\theta^{(t)}$
- Maximizing $F(\theta^{(t)}, Q)$ over Q $\rightarrow$ set Q to posterior probability:

$$Q^{(t+1)}(\mathbf{z} \mid \mathbf{x}_j) \leftarrow P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})$$

$\forall z$ / write a table (if z is discrete) : $\forall j$

z	$Q(z \mid x_j)$
true	0.8
false	0.2

$\Rightarrow$ M step weight

$\langle x_j, z=t \rangle$, 0.8

$\langle x_j, z=f \rangle$, 0.2

- Note that $F(\theta^{(t)}, Q^{(t+1)}) \;=\; \ell(\theta^{(t)} : \mathcal{D})$

EM is coordinate ascent

$$\ell(\theta : \mathcal{D}) \;\geq\; F(\theta, Q) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q(\mathbf{z} \mid \mathbf{x}_j) \log \frac{P(\mathbf{z}, \mathbf{x}_j \mid \theta)}{Q(\mathbf{z} \mid \mathbf{x}_j)}$$

- **M-step**: Fix Q, maximize F over θ (a lower bound on $\ell(\theta : \mathcal{D})$):

$$\ell(\theta : \mathcal{D}) \;\geq\; F(\theta, Q^{(t)}) = \sum_{j=1}^{m} \sum_{\mathbf{z}} Q^{(t)}(\mathbf{z} \mid \mathbf{x}_j) \log P(\mathbf{z}, \mathbf{x}_j \mid \theta) + H(Q^{(t)})$$

learn from weighted data

- **E-step**: Fix θ, maximize F over Q:

$$\ell(\theta^{(t)} : \mathcal{D}) \geq F(\theta^{(t)}, Q) \;=\; \ell(\theta^{(t)} : \mathcal{D}) - \sum_{j=1}^{m} KL\left(Q(\mathbf{z} \mid \mathbf{x}_j) || P(\mathbf{z} \mid \mathbf{x}_j, \theta^{(t)})\right)$$

doesn't depend on Q

setting to ∅

  - "Realigns" F with likelihood:

$$F(\theta^{(t)}, Q^{(t+1)}) \;=\; \ell(\theta^{(t)} : \mathcal{D})$$

What you should know

- K-means for clustering:
  - algorithm
  - converges because it's coordinate ascent
- EM for mixture of Gaussians:
  - How to "learn" maximum likelihood parameters (locally max. like.) in the case of unlabeled data
- Be happy with this kind of probabilistic analysis
- Remember, E.M. can get stuck in local minima, and empirically it DOES
- EM is coordinate ascent
- General case for EM

[random restarts]

Acknowledgements

- K-means & Gaussian mixture models presentation contains material from excellent tutorial by Andrew Moore:
  - http://www.autonlab.org/tutorials/
- K-means Applet:
  - http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html
- Gaussian mixture models Applet:
  - http://www.neurosci.aist.go.jp/%7Eakaho/MixtureEM.html

EM for HMMs a.k.a. The Baum-Welch Algorithm

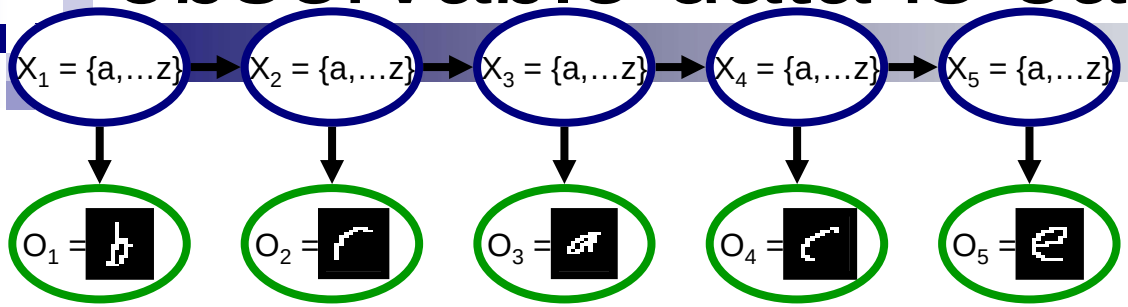
Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

April 10th, 2006

Learning HMMs from fully observable data is easy



Learn 3 distributions:

$$P(X_1 = a) = \frac{\text{Count}(\#\text{ first letter was } a)}{\#\text{= dataset size}}$$

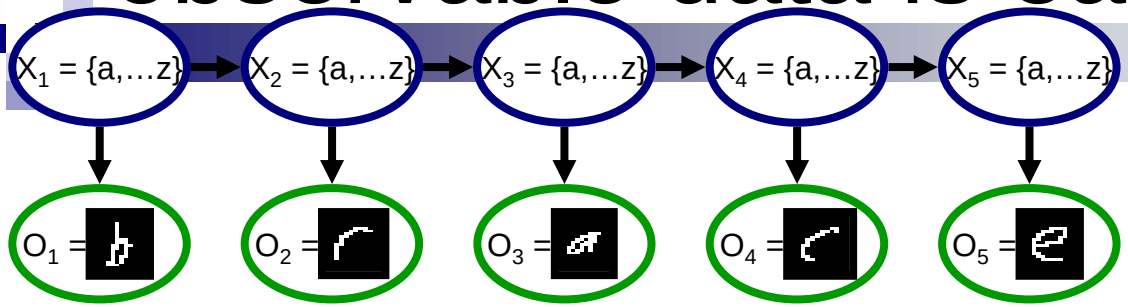
select training data where letter was a

$$P(O_i = \text{pixel 17 is white} \mid X_i = a) = \frac{\text{Count}(\text{pixel 17 was white}, X_i = a)}{\text{Count}(X_i = a)}$$

Count any position in word

$$P(X_i = a \mid X_{i-1} = b) = \frac{\text{Count}(a \text{ appears after } b)}{\text{Count}(\#\text{ of } b\text{'s that are not at the end of the word})}$$

Learning HMMs from fully observable data is easy



Learn 3 distributions:

$$P(X_1 = a) = \frac{\text{count}(\#\text{ first letter was } a)}{\#\text{ = dataset size}}$$

select training data where letter was a

$$P(O_i \mid X_i = a) = \frac{\text{count}(\text{pixel 17 was white}, X_i = a)}{\text{count}(\text{any } X_i = a)}$$

$$P(X_i = a \mid X_{i-1} = b)$$

What if **O** is observed, but **X** is hidden

Log likelihood for HMMs with hidden **X**

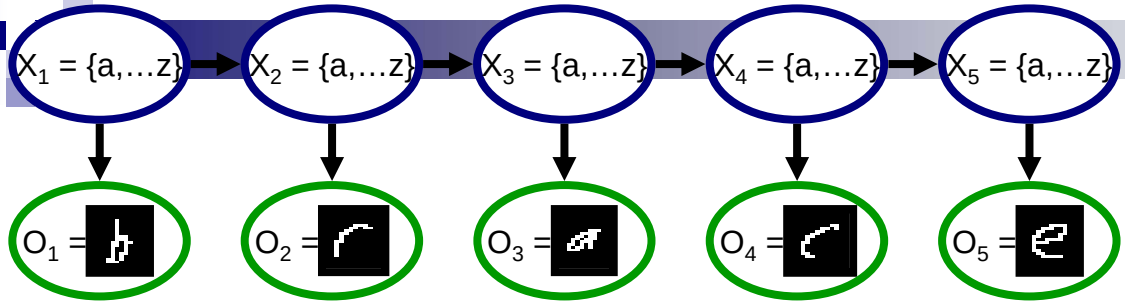
- Marginal likelihood – **O** is observed, **X** is missing
  - For simplicity of notation, we'll consider training data consists of only one sequence:

$$\begin{aligned} \ell(\theta : \mathcal{D}) &= \log P(\mathbf{o} \mid \theta) \\ &= \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o} \mid \theta) \end{aligned}$$

  - If there were m sequences:

$$\ell(\theta : \mathcal{D}) = \sum_{j=1}^{m} \log \sum_{\mathbf{x}} P(\mathbf{x}, \mathbf{o}^{(j)} \mid \theta)$$

E-step

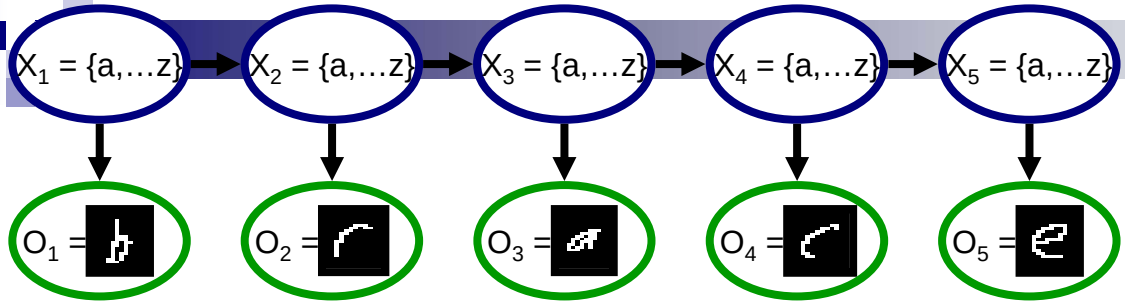


- E-step computes probability of hidden vars **x** given **o**

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \;=\; P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$

- Will correspond to inference
  - use forward-backward algorithm!

The M-step



- Maximization step:

$$\theta^{(t+1)} \leftarrow \arg\max_{\theta} \sum_{\mathbf{z}} Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) \log P(\mathbf{x}, \mathbf{o} \mid \theta)$$

- Use expected counts instead of counts:
  - If learning requires Count(**x**,**o**)
  - Use $E_{Q(t+1)}$[Count(**x**,**o**)]

Starting state probability $P(X_1)$

- Using expected counts
  - $P(X_1=a) = \theta_{X1=a}$

$$\theta_{X_1=a} = \frac{\sum_{j=1}^{m} Q(X_1 = a \mid \mathbf{o}^{(j)})}{m}$$

Transition probability $P(X_{t+1}|X_t)$

- Using expected counts
  - $P(X_{t+1}=a|X_t=b) = \theta_{Xt+1=a|Xt=b}$

$$\theta_{X_{t+1}=a|X_t=b} = \frac{\sum_{j=1}^{m}\sum_{t=1}^{n-1} Q(X_{t+1}=a, X_t=b \mid \mathbf{o}^{(j)})}{\sum_{j=1}^{m}\sum_{t=1}^{n-1}\sum_{i=1}^{k} Q(X_{t+1}=i, X_t=b \mid \mathbf{o}^{(j)})}$$

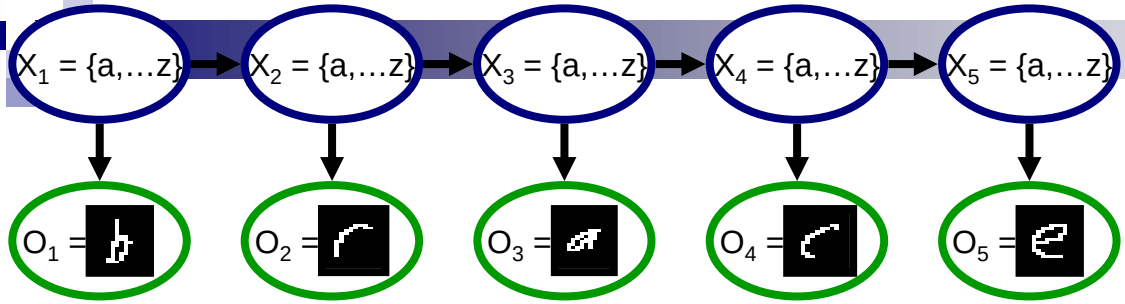
Observation probability $P(O_t|X_t)$

- Using expected counts
  - $P(O_t=a|X_t=b) = \theta_{Ot=a|Xt=b}$

$$\theta_{O_t=a|X_t=b} = \frac{\sum_{j=1}^{m}\sum_{t=1}^{n}\delta(\mathbf{o}_t^{(j)}=a)Q(X_t=b\mid\mathbf{o}^{(j)})}{\sum_{j=1}^{m}\sum_{t=1}^{n}Q(X_t=b\mid\mathbf{o}^{(j)})}$$

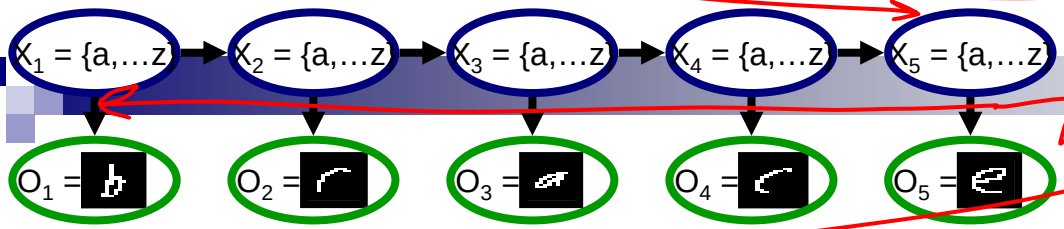
E-step revisited

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) = P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$



- E-step computes probability of hidden vars **x** given **o**
- Must compute:
  - Q(x_t=a|**o**) – marginal probability of each position
  - Q(x_{t+1}=a,x_t=b|**o**) – joint distribution between pairs of positions

The forwards-backwards algorithm



$$P(X_i \mid o_{1..n})$$

- Initialization: $\alpha_1(X_1) = P(X_1)P(o_1 \mid X_1)$
- For i = 2 to n
  - Generate a forwards factor by eliminating X_{i-1}

$$\alpha_i(X_i) = \sum_{x_{i-1}} P(o_i \mid X_i)P(X_i \mid X_{i-1} = x_{i-1})\alpha_{i-1}(x_{i-1})$$

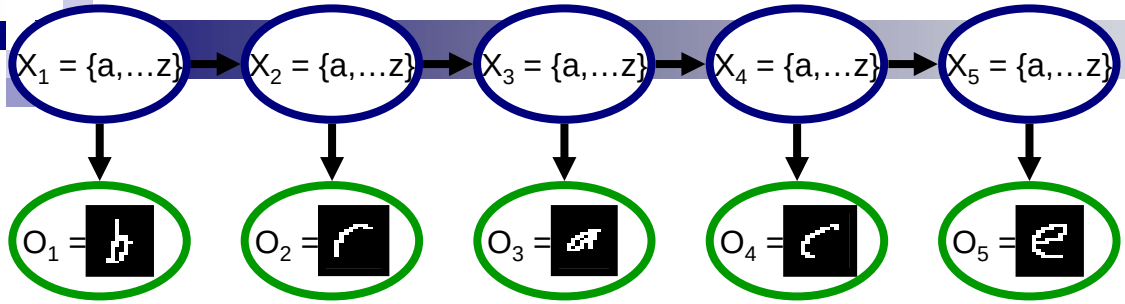
- Initialization: $\beta_n(X_n) = 1$
- For i = n-1 to 1
  - Generate a backwards factor by eliminating X_{i+1}

$$\beta_i(X_i) = \sum_{x_{i+1}} P(o_{i+1} \mid x_{i+1})P(x_{i+1} \mid X_i)\beta_{i+1}(x_{i+1})$$

- ∀ i, probability is: $P(X_i \mid o_{1..n}) \propto \alpha_i(X_i)\beta_i(X_i)$

E-step revisited

$$Q^{(t+1)}(\mathbf{x} \mid \mathbf{o}) = P(\mathbf{x} \mid \mathbf{o}, \theta^{(t)})$$



- E-step computes probability of hidden vars **x** given **o**
- Must compute:
  - $Q(x_t=a|\mathbf{o})$ – marginal probability of each position
    - Just forwards-backwards!
  - $Q(x_{t+1}=a,x_t=b|\mathbf{o})$ – joint distribution between pairs of positions
    - Homework! ☺