EM (cont.)

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

November 26th, 2007

Silly Example

Let events be "grades in a class"

w_1 = Gets an A P(A) = ½
w_2 = Gets a B P(B) = μ
w_3 = Gets a C P(C) = 2μ
w_4 = Gets a D P(D) = ½-3μ

(Note $0 \leq \mu \leq 1/6$)

Assume we want to estimate μ from data. In a given class there were

a A's
b B's
c C's
d D's

What's the maximum likelihood estimate of μ given a,b,c,d ?

$\hat{\mu}_{MLE}$

Trivial Statistics

P(A) = ½ P(B) = μ P(C) = 2μ P(D) = ½-3μ

P(*a,b,c,d* | μ) = K(½)^a(μ)^b(2μ)^c(½-3μ)^d

log P(*a,b,c,d* | μ) = log K + *a*log ½ + *b*log μ + *c*log 2μ + *d*log (½-3μ)

FOR MAX LIKE μ, SET $\frac{\partial \text{LogP}}{\partial \mu} = 0$

$$\frac{\partial \text{LogP}}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0$$

Gives max like $\mu = \frac{b+c}{6(b+c+d)}$

So if class got

A	B	C	D
14	6	9	10

Max like $\mu = \frac{1}{10}$

Boring, but true!

Same Problem with Hidden Information

Someone tells us that

Number of High grades (A's + B's) = *h*

Number of C's = *c*

Number of D's = *d*

What is the max. like estimate of μ now?

We can answer this question circularly:

EXPECTATION

If we know the value of μ we could compute the expected value of *a* and *b*

Since the ratio a:b should be the same as the ratio ½ : μ

$$\tilde{a} = \frac{1/2}{1/2 + \mu} h \qquad \tilde{b} = \frac{\mu}{1/2 + \mu} h$$

MAXIMIZATION

If we know the expected values of $\tilde{a}$ and $\tilde{b}$ we could compute the maximum likelihood value of μ

$$\hat{\mu} = \frac{\tilde{b} + c}{6(\tilde{b} + c + d)}$$

REMEMBER

P(A) = ½

P(B) = μ

P(C) = 2μ

P(D) = ½-3μ

E.M. for our Trivial Problem

REMEMBER
P(A) = ½
P(B) = μ
P(C) = 2μ
P(D) = ½-3μ

We begin with a guess for μ
We iterate between EXPECTATION and MAXIMALIZATION to improve our estimates of μ and *a* and *b*.

Define $\mu^{(t)}$ the estimate of μ on the t'th iteration
$b^{(t)}$ the estimate of *b* on t'th iteration

$$\mu^{(0)} = \text{initial guess}$$

$$b^{(t)} = \frac{\mu^{(t)} h}{1/2 + \mu^{(t)}} = \mathrm{E}\left[b \mid \mu^{(t)}\right]$$

E-step

$$\mu^{(t+1)} = \frac{b^{(t)} + c}{6\left(b^{(t)} + c + d\right)}$$

$$= \text{max like est. of } \mu \text{ given } b^{(t)}$$

M-step

Continue iterating until converged.
Good news: Converging to local optimum is assured.
Bad news: I said "local" optimum.

E.M. Convergence

- Convergence proof based on fact that Prob(data | μ) must increase or remain same between each iteration [NOT OBVIOUS]
- But it can never exceed 1 [OBVIOUS]

So it must therefore converge [OBVIOUS]

In our example, suppose we had

h = 20
c = 10
d = 10
$\mu^{(0)}$ = 0

t	$\mu^{(t)}$	$b^{(t)}$
0	0	0
1	0.0833	2.857
2	0.0937	3.158
3	0.0947	3.185
4	0.0948	3.187
5	0.0948	3.187
6	0.0948	3.187

Convergence is generally linear: error decreases by a constant factor each time step.

Back to Unsupervised Learning of GMMs – a simple case

A simple case:

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 $\boldsymbol{\mu}_1\ \boldsymbol{\mu}_2\ \ldots\ \boldsymbol{\mu}_k$

maximize

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

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

iid

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

summing over possible classes

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

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

EM for simple case of GMMs: The E-step

Expected value of hidden vars

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

Bayes Rule

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

for each point j

$P(y=1 | x_j, \mu_1 \ldots \mu_k) = 0.7$

$P(y=2 | x_j, \mu_1 \ldots \mu_k) = 0.2$

$P(y=3 | x_j, \mu_1 \ldots \mu_k) = 0.1$

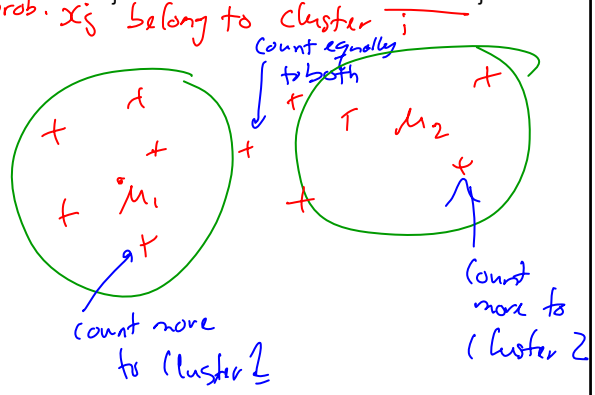
EM for simple case of GMMs: The M-step

- 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 $P(y=i|x_j)$:

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



E.M. for GMMs

E-step

Compute "expected" classes of each datapoint for each class

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

Just evaluate a Gaussian at x_j

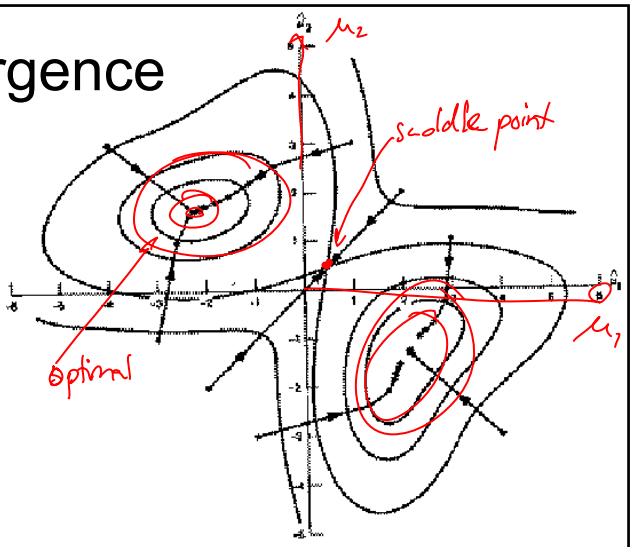
M-step

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

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

E.M. Convergence

- EM is coordinate ascent on an interesting potential function
- Coord. ascent for bounded pot. func. ! convergence to a local optimum guaranteed
- See Neal & Hinton reading on class webpage



- This algorithm is REALLY USED. And in high dimensional state spaces, too. E.G. Vector Quantization for Speech Data

E.M. for axis-aligned GMM

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

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

$\Sigma_i = \begin{pmatrix} \sigma_{i1}^2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_{i2}^2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \sigma_{i3}^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{im-1}^2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \sigma_{im}^2 \end{pmatrix}$

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

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

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

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

m = #records

E.M. for General GMMs

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

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

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

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

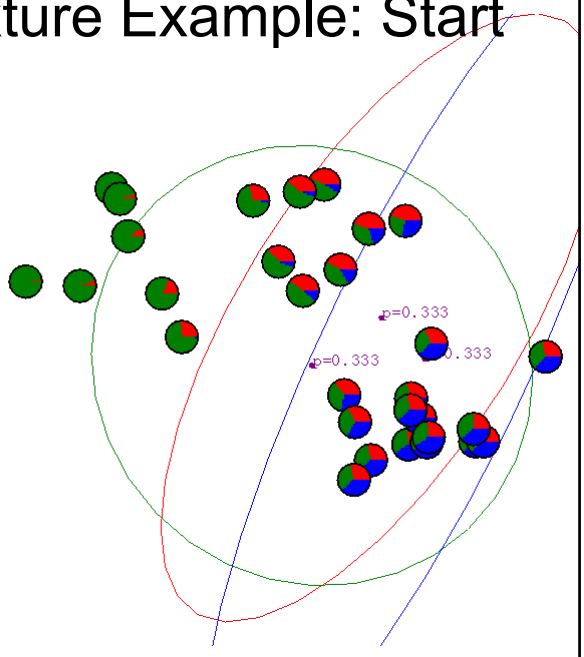
$$\mu_i^{(t+1)} = \frac{\sum_j P(y=i|x_j,\lambda_t) x_j}{\sum_j P(y=i|x_j,\lambda_t)}$$

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

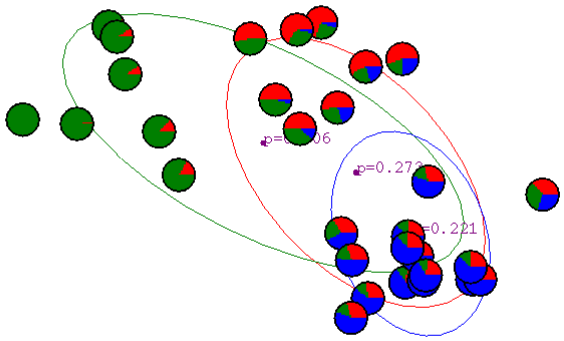
$$p_i^{(t+1)} = \frac{\sum_j P(y=i|x_j,\lambda_t)}{m}$$

m = #records

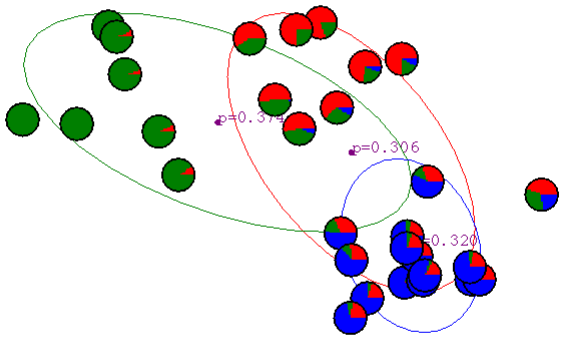
Gaussian Mixture Example: Start



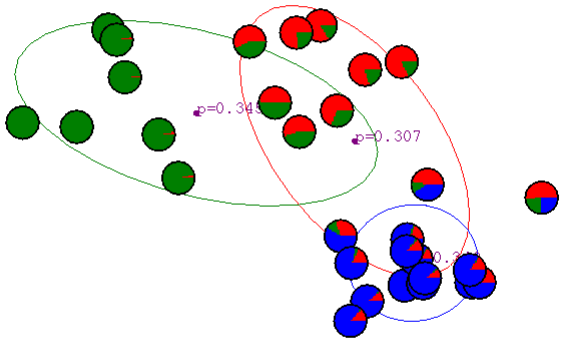
After first iteration



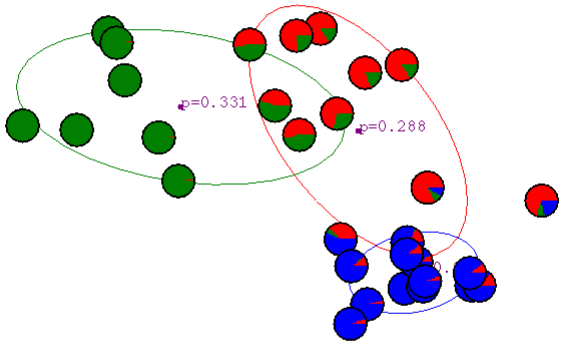
After 2nd iteration



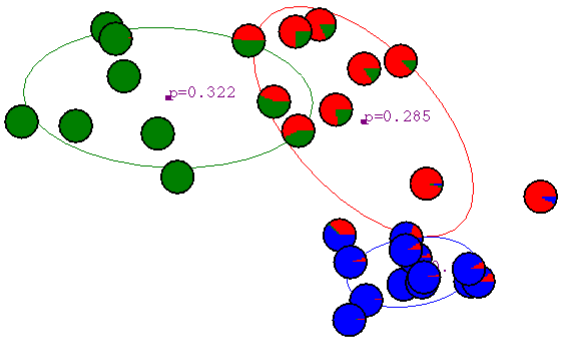
After 3rd iteration



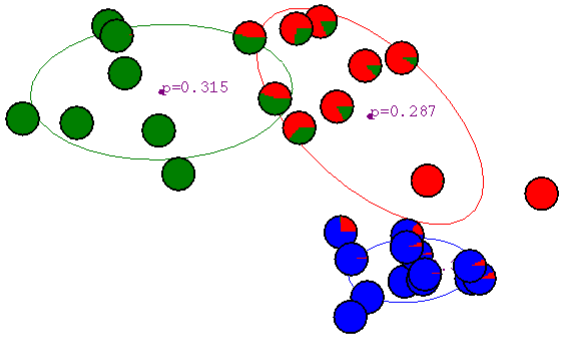
After 4th iteration



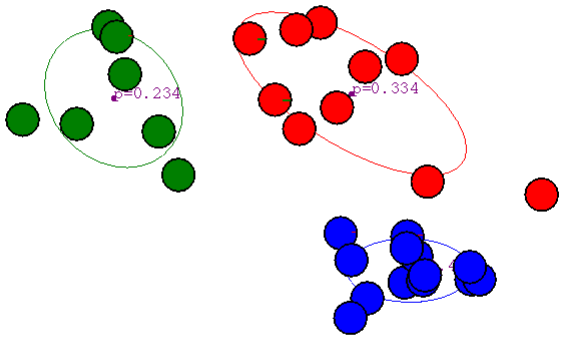
After 5th iteration



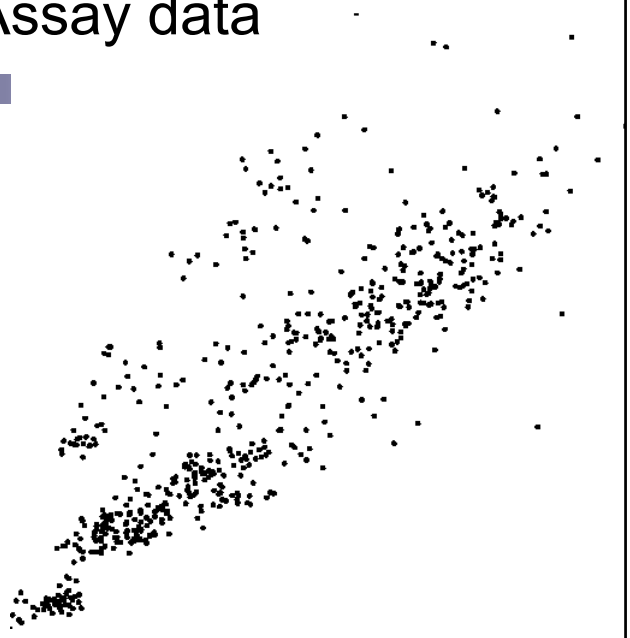
After 6th iteration



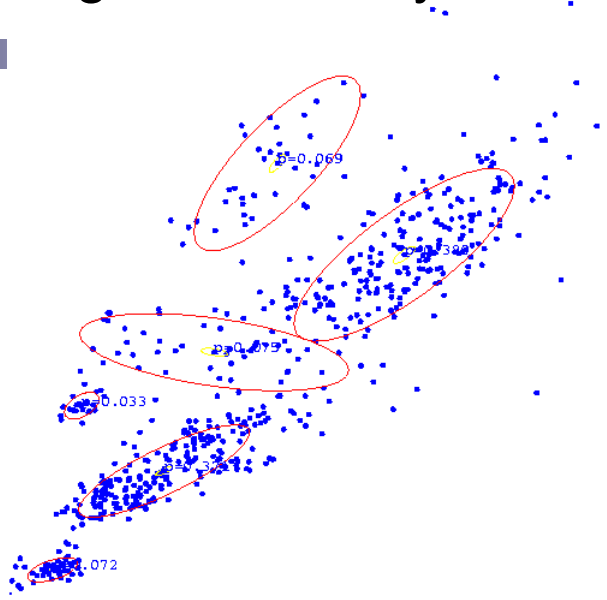
After 20th iteration



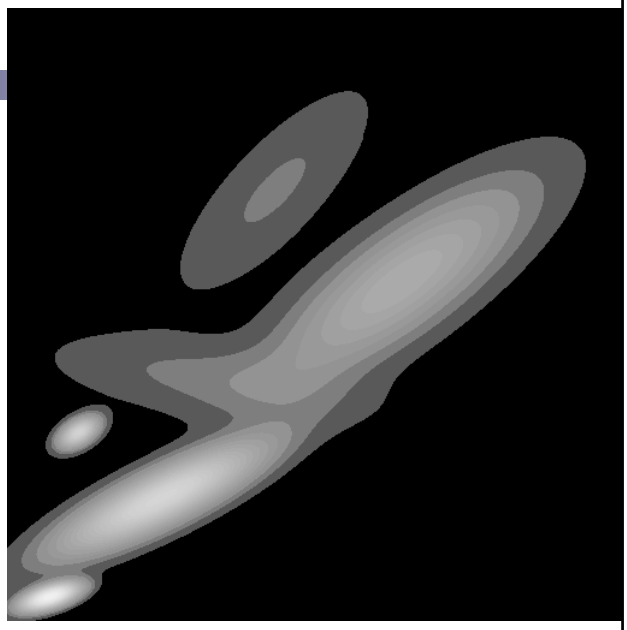
Some Bio Assay data



GMM clustering of the assay data

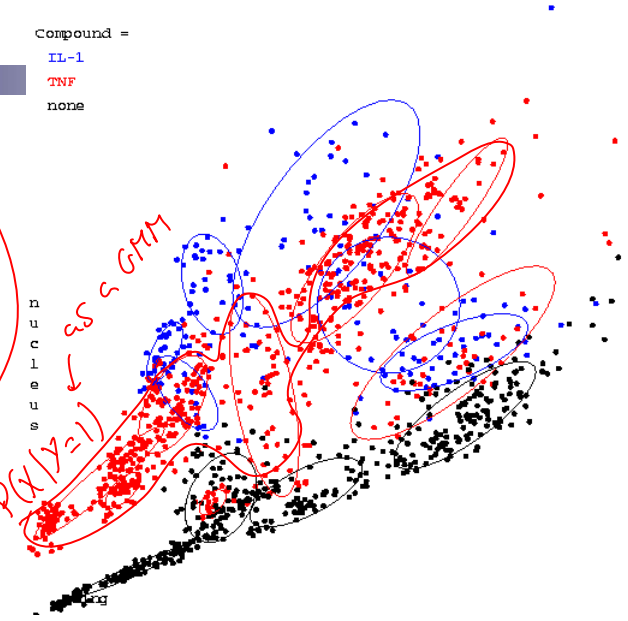


Resulting Density Estimator

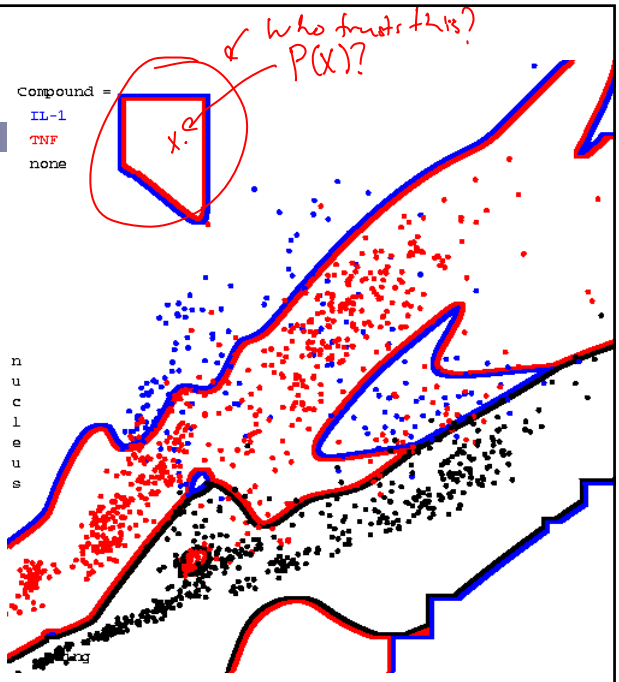


Three classes of assay

(each learned with it's own mixture model)



Resulting Bayes Classifier

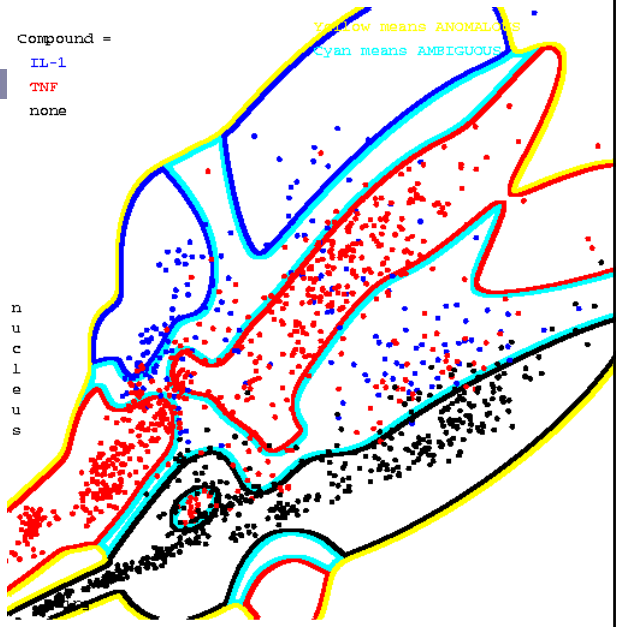


in yellow: regions where p(X) is low.

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

Yellow means anomalous

Cyan means ambiguous



Homeworks to Monica Please

Announcements

Please fill out your University Course Assessment

Postbox away next week

Monday/Tuesday 12/10-11 many extra office hours, arrive 15 mins early

- Project:
  - Poster session: NSH Atrium, Friday 11/30, 2-5pm 4:45pm
    - Print your poster early!!!
      - SCS facilities has a poster printer, ask helpdesk
      - Students from outside SCS should check with their departments
      - It's OK to print separate pages
    - We'll provide pins, posterboard and an easel
      - Poster size: 32x40 inches
    - Invite your friends, there will be a prize for best poster, by popular vote
- Last lecture:
  - Thursday, 11/29, 5-6:20pm, Wean 7500

The general learning problem with missing data

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

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

want to max

summing out hidden vars z

E-step

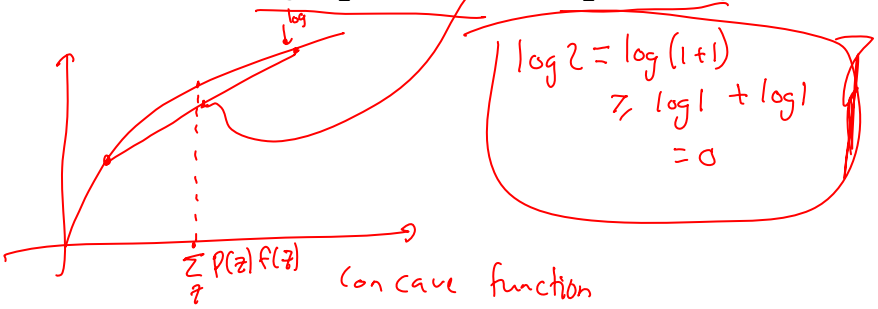
- **x** is observed, **z** is missing
- Compute probability of missing data given current choice of θ
  - $Q(\mathbf{z}|\mathbf{x}_j)$ for each $\mathbf{x}_j$
    - e.g., probability computed during classification step
    - corresponds to "classification step" in K-means

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

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**: $\log \sum_z P(z) f(z) \geq \sum_z P(z) \log f(z)$



Applying Jensen's inequality

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

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

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

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

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

$-H(Q^{(t+1)} \mid x_j)$

The M-step maximizes lower bound on weighted data

- Lower bound from Jensen's:

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

- 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

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

- Maximization step:

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

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

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

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

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

E-step also doesn't decrease potential function 1

- Fixing θ to $\theta^{(t)}$:

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

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

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

$$\sum_{z} Q(z \mid x_j) = 1 \quad \text{(doesn't depend on } z\text{)}$$

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

$-$ KL divergence

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

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

chain rule $P(z, x_j \mid \theta) = P(z \mid x_j, \theta) P(x_j \mid \theta)$

KL-divergence

- Measures distance between distributions

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

- KL=zero if and only if Q=P

E-step also doesn't decrease potential function 2

- Fixing θ to $\theta^{(t)}$:

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

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

~KL

maximize the right side, we know that $KL(Q||P) \geq 0$, $= 0 \Leftrightarrow P = Q$

as small as possible to max. $\ell(\theta : \mathcal{D})$

↳ by setting $Q(z|x_j) = P(z|x_j, \theta^{(t)})$

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 → set Q to posterior probability:

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

- Note that

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

KL = 0 ⇓

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 $\ell(\theta : \mathcal{D})$):

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

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

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

  - "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

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

Dimensionality Reduction (PCA)

Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

November 26th, 2007

Dimensionality reduction

- Input data may have thousands or millions of dimensions!
  - e.g., text data has 40K words
- **Dimensionality reduction**: represent data with fewer dimensions
  - easier learning – fewer parameters
  - visualization – hard to visualize more than 3D or 4D
  - discover "intrinsic dimensionality" of data
    - high dimensional data that is truly lower dimensional

Feature selection

- Want to learn f:**X**→Y
  - **X**=<X_1,…,X_n> = 40K
  - but some features are more important than others
- **Approach**: select subset of features to be used by learning algorithm
  - **Score** each feature (or sets of features)
  - **Select** set of features with best score

Simple greedy **forward** feature selection algorithm

- Pick a dictionary of features
  - e.g., polynomials for linear regression
- Greedy heuristic:
  - Start from empty (or simple) set of features $F_0 = \emptyset$
  - Run learning algorithm for current set of features F_t
    - Obtain h_t
  - Select **next best feature X_i**
    - e.g., X_j that results in lowest cross-validation error learner when learning with $F_t \cup \{X_j\}$
  - $F_{t+1} \leftarrow F_t \cup \{X_i\}$
  - Recurse

Simple greedy **backward** feature selection algorithm

- Pick a dictionary of features
  - e.g., polynomials for linear regression
- Greedy heuristic:
  - Start from all features $F_0 = F$
  - Run learning algorithm for current set of features F_t
    - Obtain h_t
  - Select **next worst feature X_i**
    - e.g., X_j that results in lowest cross-validation error learner when learning with $F_t - \{X_j\}$
  - $F_{t+1} \leftarrow F_t - \{X_i\}$
  - Recurse

Impact of feature selection on classification of fMRI data [Pereira et al. '05]

Accuracy classifying category of word read by subject

#voxels	mean	subjects							
		233B	329B	332B	424B	474B	496B	77B	86B
50	0.735	0.783	0.817	0.55	0.783	0.75	0.8	0.65	0.75
100	0.742	0.767	0.8	0.533	0.817	0.85	0.783	0.6	0.783
200	0.737	0.783	0.783	0.517	0.817	0.883	0.75	0.583	0.783
300	**0.75**	**0.8**	**0.817**	**0.567**	**0.833**	**0.883**	**0.75**	**0.583**	**0.767**
400	0.742	0.8	0.783	0.583	0.85	0.883	0.75	0.583	0.75
800	0.735	0.833	0.817	0.567	0.833	0.883	0.7	0.55	0.75
1600	0.698	0.8	0.817	0.45	0.783	0.883	0.633	0.5	0.75
all (~2500)	0.638	0.767	0.767	0.25	0.75	0.883	0.567	0.433	0.733

Table 1: **Average accuracy across all pairs of categories, restricting the procedure to use a certain number of voxels for each subject.** The highlighted line corresponds to the best mean accuracy, obtained using 300 voxels.

Voxels scored by p-value of regression to predict voxel value from the task

Lower dimensional projections

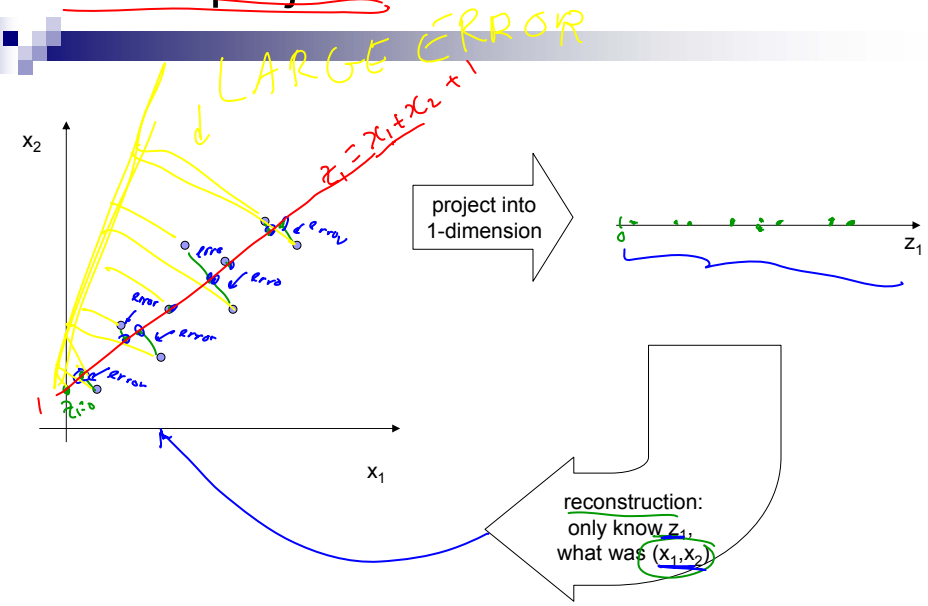
- Rather than picking a subset of the features, we can new features that are combinations of existing features

e.g., feature selection: use X_1, X_7, X_{11}

low. dim. proj. $\tilde{X} = 0.1 X_1 + 0.7 X_2 - 0.35 X_3 \ldots$

- Let's see this in the unsupervised setting
  - just **X**, but no Y

Linear projection and reconstruction



Principal component analysis – basic idea

- Project n-dimensional data into k-dimensional space while preserving information:
  - e.g., project space of 10000 words into 3-dimensions
  - e.g., project 3-d into 2-d

- Choose projection with minimum reconstruction error

Linear projections, a review

- Project a point into a (lower dimensional) space:
  - **point**: $\mathbf{x} = (x_1,\ldots,x_n)$
  - **select a basis** – set of basis vectors – $(\mathbf{u}_1,\ldots,\mathbf{u}_k)$
    - we consider orthonormal basis:
      - $\mathbf{u}_i \bullet \mathbf{u}_i = 1$ and $\mathbf{u}_i \bullet \mathbf{u}_j = 0$ for $i \neq j$
  - **select a center** – $\bar{\mathbf{x}}$, defines offset of space
  - **best coordinates** in lower dimensional space defined by dot-products: $(z_1,\ldots,z_k)$, $z_i = (\mathbf{x}-\bar{\mathbf{x}}) \bullet \mathbf{u}_i$
    - minimum squared error