Today: - Spectral clustering
- Bayes classifiers

Next time: - Decision trees
Spectral clustering

• We talked about the following three steps:
  1. construct a neighborhood graph
  2. assign weights to the edges in the graph
  3. define a transition probability matrix based on the weights
Properties of the random walk

- The distributions of points we end up in after $t$ steps converge as $t$ increases. If the graph is connected, the resulting distribution is independent of the starting point.

- However, even for large $t$, the transition probabilities $[P^t]_{ij}$ have a slightly higher probability of transitioning within "clusters" than across; we want to recover this effect from eigenvalues/vectors.
Eigenvalue decomposition

• W is the distance matrix
• D is a diagonal matrix such that $D_{ii} = \Sigma_j W_{ij}$.

$$P = D^{-1}W$$

• We have also defined a symmetric matrix and discussed its eigen decomposition

$$D^{-1/2}W D^{-1/2} = \lambda_1 z_1 z_1^T + \lambda_2 z_2 z_2^T \cdots + \lambda_n z_n z_n^T$$

where $z_i$ is the $i$th eigenvector and the ordering is such that $1=|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$.
Eigen decomposition (cont.)

- The symmetric matrix is related to $P^t$ since

$$
(D \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}) \cdots (D \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}) = D^2 (P \cdots P) D^{-2}
$$
Eigen decomposition (cont.)

• The symmetric matrix is related to $P^t$ since

\[
(D^{-1}WD^{-1}) \cdots (D^{-1}WD^{-1}) = D^2 (P \cdots P) D^{-2}
\]

• This allows us to write the $t$ step transition probability matrix in terms of the eigenvalues/vectors of the symmetric matrix

\[
P^t = D^{-2} D^2 (P \cdots P) D^{-2} D^2
\]

\[
= D^{-2} (D^{-2} WD^{-2})^t D^2
\]
Eigen decomposition (cont.)

• The symmetric matrix is related to $P^t$ since

\[
(D^{-\frac{1}{2}} WD^{-\frac{1}{2}}) \cdots (D^{-\frac{1}{2}} WD^{-\frac{1}{2}}) = D^2 (P \cdots P) D^{-\frac{1}{2}}
\]

• This allows us to write the $t$ step transition probability matrix in terms of the eigenvalues/vectors of the symmetric matrix

\[
P^t = D^{-\frac{1}{2}} D^2 (P \cdots P) D^{-\frac{1}{2}} D^2
= D^{-\frac{1}{2}} (D^{-\frac{1}{2}} WD^{-\frac{1}{2}})^t D^2
= D^{-\frac{1}{2}} (\lambda_1^t z_1 z_1^T + \lambda_2^t z_2 z_2^T \cdots + \lambda_n^t z_n z_n^T) D^2
\]
Expressing $P^t$

\[
P^t = D^{\frac{-1}{2}} \left( \lambda_1 z_1 z_1^T + \lambda_2 z_2 z_2^T + \cdots + \lambda_n z_n z_n^T \right) D^{\frac{1}{2}}
\]

Where $\lambda_1 = 1$ and for all other $i$, $\lambda_i < 1$

• Thus:

\[
P^\infty =
\]
Expressing $P^t$

$$P^t = D^{-\frac{1}{2}} \left( \lambda_1^t z_1 z_1^T + \lambda_2^t z_2 z_2^T + \cdots + \lambda_n^t z_n z_n^T \right) D^{\frac{1}{2}}$$

Where $\lambda_1 = 1$ and for all other $i$, $\lambda_i < 1$

• Thus:

$$P^\infty = D^{-\frac{1}{2}} (z_1 z_1^T) D^{\frac{1}{2}}$$
Clustering

• We are interested in the largest correction to the asymptotic limit

\[ P^t = P^\infty + D \frac{1}{2} (\lambda_2 z_2 z_2^T) \frac{1}{D^2} \]

• The largest correction term should *increase* the probability of transitions between points in the same cluster and *decrease* the transition probability between points in different clusters.

• Thus, points in the same cluster will share the same sign of $z_2$ and points in different clusters will differ in their sign.
Binary clustering

- We divide the points into two clusters based on the sign of the elements of $z_2$:
  
  $z_{2j} > 0 \implies$ cluster 1, otherwise cluster 0

![Diagram showing binary clustering with points and clusters highlighted]
The sign of the second eigenvector

The entries in the eigenvector corresponding to the second largest eigenvalue
Bayesian classifiers
Where we are

- **Classifier** predicts category
- **Density Estimator** predicts probability
- **Regressor** predicts real number
Text classification

• Text classification (information retrieval):
  - A few labeled documents $D_l = \{(x_1,y_1),\ldots,(x_n,y_n)\}$
  - Many unlabeled data points: $D_t = \{(x_{n+1}),\ldots,(x_{n+m})\}$

• Problem formulations:
  – train with $D_l$
  – classify all the unlabeled examples in $D_t$
  (there are also other ways to formulate this problem)

• Several steps:
  1. feature transformation
  2. model/classifier specification
  3. model/classifier estimation (with regularization)
  4. feature selection
Feature transformation

• We can construct $m$ (about 10,000) indicator features (basis functions) $\{\phi_i(x)\}$ for whether a word appears in the document
  - $\phi_i(x) = 1$, if word $i$ appears in document $x$; zero otherwise
• $\Phi(x) = [\phi_1(x), \ldots, \phi_m(x)]^T$ is the resulting feature vector
• For notational simplicity we will replace each document $x$ with a fixed length vector $\Phi = [\phi_1, \ldots, \phi_m]^T$, where $\phi_i = \phi_i(x)$. 
Example

Dictionary

- Washington
- Congress

... 

54. Bush
55. Kerry
56. Nader

\[ \phi_{54} = \phi_{54}(x) = 1 \]
\[ \phi_{55} = \phi_{55}(x) = 1 \]
\[ \phi_{56} = \phi_{56}(x) = 0 \]
Classifiers

• Two main types:
  1. Discriminative (e.g. support vector machines)
     - need to chose the kernel function
  2. Generative
     - need to define class conditional probabilities
Naïve Bayes Classifier

• In order to find the class of the document we would like to find $y$ such that:

$$
\hat{y} = \arg \max_y p(y = v \mid \Phi)
$$

$$
= \arg \max_y \frac{p(\Phi \mid y = v) p(y = v)}{p(\Phi)}
$$

$$
= \arg \max_y p(\Phi \mid y = v) p(y = v)
$$

• We may assume, for example, that within each class of documents, the presence/absence of each word is independent of other words.

• This gives rise to a “Naïve Bayes” model over documents and labels:

$$
p(\Phi \mid y) p(y) = \left[ \prod_i p(\phi_i \mid y, i) \right] p(y)
$$
Determining the parameters

- If we have a two class problem then \( y \in \{0, 1\} \).
- Thus, we can write the individual conditional probabilities compactly as
  \[
p(\phi_i \mid y, i) = (1 - \theta_{i|y})^{1-\phi_i} \theta_{i|y}^{\phi_i}
\]
  where \( \theta_{i|y} \) is the conditional probability that \( \phi_i = 1 \) given the class \( y \).
- We can find the parameters that maximize the likelihood of our data in a similar way to what we did in the logistic regression case.
Parameter estimation

- The log likelihood can be written as:

\[
l(\Phi, y) = \sum_i N_i(1, y) \log(\theta_{i|y}) + N_i(0, y) \log(1 - \theta_{i|y})
\]

\(N_i(1,y) = \# \text{ of documents containing word } i \text{ and labeled } y\)
\(N_i(0,y) = \# \text{ of documents without word } i \text{ and labeled } y\)

- Taking the derivative w.r.t. \(\theta_{i|y}\) gives:

\[
\frac{\partial l(\Phi, y)}{\partial \theta_{i|y}} = \frac{N_i(1, y)}{\theta_{i|y}} - \frac{N_i(0, y)}{1 - \theta_{i|y}}
\]

\(\Rightarrow \hat{\theta}_{i|y} = \frac{N_i(1, y)}{N_i(1, y) + N_i(0, y)}\)
Parameter estimation

- The log likelihood can be written as:

\[ l(\Phi, y) = \sum_{i} N_{i}(1, y) \log(\theta_{i|y}) + N_{i}(0, y) \log(1 - \theta_{i|y}) \]

\[ N_{i}(1, y) = \text{# of documents containing word } i \text{ and labeled } y \]

\[ N_{i}(0, y) = \text{# of documents without word } i \text{ and labeled } y \]

Note that 1 and 0 in \( N_{i} \) refer to the presence or absence of word \( i \) in the set of documents (and not to the class label).
Gaussian Bayes Classifier Assumption

- The i’th record in the database is created using the following algorithm
  1. Generate the output (the “class”) by drawing $y_i \sim \text{Multinomial}(p_1, p_2, \ldots p_{N_y})$
  2. Generate the inputs from a Gaussian PDF that depends on the value of $y_i$:
     \[ x_i \sim N(\mu_i, \Sigma_i). \]
MLE Gaussian Bayes Classifier

Let $DB_i = \text{Subset of database } DB \text{ in which the output class is } y = i$.

The $i^{th}$ record in the database is created using

1. Generate the output (the "class") by drawing $y_i \sim \text{Multinomial}(p_1, p_2, \ldots, p_{Ny})$

2. Generate the inputs from a Gaussian PDF that depends on the value of $y_i$:

$$x_i \sim N(\mu_i, \Sigma_i).$$
MLE Gaussian Bayes Classifier

Let \( \text{DB}_i \) = Subset of database DB in which the output class is \( y = i \)

1. Generate the output (the "class") by drawing \( y_i \sim \text{Multinomial}(p_1, p_2, \ldots, p_N) \)
2. Generate the inputs from a Gaussian PDF that depends on the value of \( y_i \):
   \[
   x_i \sim N(\mu_i, \Sigma_i) .
   \]

(\( \mu_i^{\text{mle}}, \Sigma_i^{\text{mle}} \) = MLE Gaussian for \( \text{DB}_i \))
MLE Gaussian Bayes Classifier

The database is created using the following algorithm:

1. Generate the output (the “class”) by drawing $y_i \sim \text{Multinomial}(p_1, p_2, \ldots, p_{N_y})$.

2. Generate the inputs from a Gaussian PDF that depends on the value of $y_i$:
   
   $$ x_i \sim N(\mu_{i,mle}, \Sigma_{i,mle}) $$

Let $DB_i = \text{Subset of database DB in which the output class is } y = i$.

Let $(\mu_{i,mle}, \Sigma_{i,mle}) = \text{MLE Gaussian for } DB_i$.

Finally, let:

$$ \mu_{i,mle} = \frac{1}{|DB_i|} \sum_{x_k \in DB_i} x_k $$

$$ \Sigma_{i,mle} = \frac{1}{|DB_i|} \sum_{x_k \in DB_i} (x_k - \mu_{i,mle})(x_k - \mu_{i,mle})^T $$
Gaussian Bayes Classification

\[ P(y = i \mid x) = \frac{p(x \mid y = i)P(y = i)}{p(x)} \]
Gaussian Bayes Classification

\[ P(y = i \mid x) = \frac{p(x \mid y = i)P(y = i)}{p(x)} \]

\[ P(y = i \mid x) = \frac{1}{(2\pi)^{m/2} \| \Sigma_i \|^{1/2}} \exp \left[ -\frac{1}{2} (x_k - \mu_i)^T \Sigma_i^{-1} (x_k - \mu_i) \right] p_i \]

How do we deal with that?
Here is a dataset

<table>
<thead>
<tr>
<th>age</th>
<th>employment</th>
<th>education</th>
<th>marital</th>
<th>job</th>
<th>relation</th>
<th>race</th>
<th>gender</th>
<th>hours</th>
<th>country</th>
<th>wealth</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>State_gov</td>
<td>Bachelors</td>
<td>Never Married</td>
<td>Adm_clerical</td>
<td>Not_in_family</td>
<td>White</td>
<td>Male</td>
<td>40</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>51</td>
<td>Self_emp</td>
<td>Bachelors</td>
<td>Married</td>
<td>Exec_man_Husband</td>
<td>White</td>
<td>Male</td>
<td>13</td>
<td>United_S States</td>
<td>poor</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Private</td>
<td>HS_grad</td>
<td>Divorced</td>
<td>Handlers_c</td>
<td>Not_in_family</td>
<td>White</td>
<td>Male</td>
<td>40</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>54</td>
<td>Private</td>
<td>11th</td>
<td>Married</td>
<td>Handlers_c</td>
<td>Husband</td>
<td>Black</td>
<td>Male</td>
<td>40</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>28</td>
<td>Private</td>
<td>Bachelors</td>
<td>Married</td>
<td>Prof_speci</td>
<td>Wife</td>
<td>Black</td>
<td>Female</td>
<td>40</td>
<td>Cuba</td>
<td>poor</td>
</tr>
<tr>
<td>38</td>
<td>Private</td>
<td>Masters</td>
<td>Married</td>
<td>Exec_man</td>
<td>Wife</td>
<td>White</td>
<td>Female</td>
<td>40</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>50</td>
<td>Private</td>
<td>9th</td>
<td>Married_s</td>
<td>Other_services</td>
<td>Not_in_family</td>
<td>Black</td>
<td>Female</td>
<td>16</td>
<td>Jamaica</td>
<td>poor</td>
</tr>
<tr>
<td>52</td>
<td>Self_emp</td>
<td>HS_grad</td>
<td>Married</td>
<td>Exec_man</td>
<td>Husband</td>
<td>White</td>
<td>Male</td>
<td>45</td>
<td>United_S States</td>
<td>rich</td>
</tr>
<tr>
<td>51</td>
<td>Private</td>
<td>Masters</td>
<td>Never Married</td>
<td>Prof_speci</td>
<td>Not_in_family</td>
<td>White</td>
<td>Female</td>
<td>50</td>
<td>United_S States</td>
<td>rich</td>
</tr>
<tr>
<td>42</td>
<td>Private</td>
<td>Bachelors</td>
<td>Married</td>
<td>Exec_man</td>
<td>Husband</td>
<td>White</td>
<td>Male</td>
<td>40</td>
<td>United_S States</td>
<td>rich</td>
</tr>
<tr>
<td>37</td>
<td>Private</td>
<td>Some_coll</td>
<td>Married</td>
<td>Exec_man</td>
<td>Husband</td>
<td>Black</td>
<td>Male</td>
<td>80</td>
<td>United_S States</td>
<td>rich</td>
</tr>
<tr>
<td>30</td>
<td>State_gov</td>
<td>Bachelors</td>
<td>Married</td>
<td>Prof_speci</td>
<td>Husband</td>
<td>Asian</td>
<td>Male</td>
<td>40</td>
<td>India</td>
<td>rich</td>
</tr>
<tr>
<td>24</td>
<td>Private</td>
<td>Bachelors</td>
<td>Never Married</td>
<td>Adm_clerical</td>
<td>Own_child</td>
<td>White</td>
<td>Female</td>
<td>30</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>33</td>
<td>Private</td>
<td>Assoc_acc</td>
<td>Never Married</td>
<td>Prof_speci</td>
<td>Not_in_family</td>
<td>Black</td>
<td>Male</td>
<td>50</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>41</td>
<td>Private</td>
<td>Assoc_voc</td>
<td>Married</td>
<td>Prof_speci</td>
<td>Husband</td>
<td>Asian</td>
<td>Male</td>
<td>40</td>
<td><em>MissingValue</em></td>
<td>rich</td>
</tr>
<tr>
<td>34</td>
<td>Private</td>
<td>7th 8th</td>
<td>Married</td>
<td>Transport</td>
<td>Husband</td>
<td>Amer_Indian</td>
<td>Male</td>
<td>45</td>
<td>Mexico</td>
<td>poor</td>
</tr>
<tr>
<td>26</td>
<td>Self_emp</td>
<td>HS_grad</td>
<td>Never Married</td>
<td>Farming</td>
<td>Own_child</td>
<td>White</td>
<td>Male</td>
<td>35</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>33</td>
<td>Private</td>
<td>HS_grad</td>
<td>Never Married</td>
<td>Machine_c</td>
<td>Unmarried</td>
<td>White</td>
<td>Male</td>
<td>40</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>38</td>
<td>Private</td>
<td>11th</td>
<td>Married</td>
<td>Sales</td>
<td>Husband</td>
<td>White</td>
<td>Male</td>
<td>50</td>
<td>United_S States</td>
<td>poor</td>
</tr>
<tr>
<td>44</td>
<td>Self_emp</td>
<td>Masters</td>
<td>Divorced</td>
<td>Exec_man</td>
<td>Unmarried</td>
<td>White</td>
<td>Female</td>
<td>45</td>
<td>United_S States</td>
<td>rich</td>
</tr>
<tr>
<td>41</td>
<td>Private</td>
<td>Doctorate</td>
<td>Married</td>
<td>Prof_speci</td>
<td>Husband</td>
<td>White</td>
<td>Male</td>
<td>60</td>
<td>United_S States</td>
<td>rich</td>
</tr>
</tbody>
</table>

48,000 records, 16 attributes [Kohavi 1995]
Predicting wealth from age

wealth = poor
(prior = 0.760718)

1  mean  cov
age  37.374  198.935

wealth = rich
(prior = 0.239282)

1  mean  cov
age  44.7727  111.618
Predicting wealth from age
Wealth from hours worked

wealth = poor
(prior = 0.760718)

1

hours_worked 38.84 152.692

wealth = rich
(prior = 0.239262)

1

hours_worked 45.4529 123.014

wealth values: poor rich

prob

1

0.6

0.2

hours_worked

10 25 40 55 70 85
Wealth from years of education

Wealth = poor
(prior = 0.760718)

1   mean  cov
edunum  9.59849  5.94225

density

Wealth = rich
(prior = 0.239282)

1   mean  cov
edunum  11.6028  5.6769

density

Wealth values: poor rich

prob

edunum
age, hours $\rightarrow$ wealth

wealth = poor
(prior = 0.760718)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>37.374</td>
<td>198.935</td>
</tr>
<tr>
<td>hours_worked</td>
<td>38.84</td>
<td>8.70283</td>
</tr>
</tbody>
</table>

wealth = rich
(prior = 0.239282)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>44.7727</td>
<td>111.618</td>
</tr>
<tr>
<td>hours_worked</td>
<td>45.4529</td>
<td>-14.1565</td>
</tr>
</tbody>
</table>
age, hours → wealth

wealth = poor
(prior = 0.760718)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>37.374</td>
<td>198.935</td>
</tr>
<tr>
<td>hours_worked</td>
<td>38.842</td>
<td>452.698</td>
</tr>
</tbody>
</table>

wealth = rich
(prior = 0.239282)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>44.7727</td>
<td>111.618</td>
</tr>
<tr>
<td>hours_worked</td>
<td>45.4528</td>
<td>176.665</td>
</tr>
</tbody>
</table>

wealth values: poor rich

hours_worked

age
Having 2 inputs instead of one helps in two ways:
Having 2 inputs instead of one helps in two ways:
1. Combining evidence from two 1d Gaussians
2. Off-diagonal covariance distinguishes class “shape”

<table>
<thead>
<tr>
<th>age</th>
<th>hours_worked</th>
<th>wealth (prior)</th>
</tr>
</thead>
<tbody>
<tr>
<td>37.374</td>
<td>38.84</td>
<td>poor (0.760718)</td>
</tr>
<tr>
<td>198.935</td>
<td>8.70283</td>
<td></td>
</tr>
<tr>
<td>8.70283</td>
<td>152.692</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>age</th>
<th>hours_worked</th>
<th>wealth (prior)</th>
</tr>
</thead>
<tbody>
<tr>
<td>44.7727</td>
<td>45.4529</td>
<td>rich (0.239282)</td>
</tr>
<tr>
<td>111.618</td>
<td>-14.1565</td>
<td></td>
</tr>
<tr>
<td>-14.1565</td>
<td>123.014</td>
<td></td>
</tr>
</tbody>
</table>
age, edunum $\rightarrow$ wealth

wealth = poor

$\text{(prior = 0.760718)}$

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>37.374</td>
<td>198.935</td>
</tr>
<tr>
<td>edunum</td>
<td>9.59849</td>
<td>-1.94765</td>
</tr>
<tr>
<td>edunum</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

wealth = rich

$\text{(prior = 0.239282)}$

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>44.7727</td>
<td>111.618</td>
</tr>
<tr>
<td>edunum</td>
<td>11.6028</td>
<td>-0.557852</td>
</tr>
<tr>
<td>edunum</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>edunum</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
age, edunum → wealth

wealth = poor
(prior = 0.760718)

1  mean  cov
age   37.374  198.935 -1.94765
edunum  9.59849 -1.94765  5.94225

wealth = rich
(prior = 0.239282)

1  mean  cov
age   44.7727  111.618 -0.557852
edunum  11.6028 -0.557852  5.6769

wealth values: poor rich

---

age
50 60 70 80 90

edunum
## Accuracy

<table>
<thead>
<tr>
<th>Name</th>
<th>Model</th>
<th>Parameters</th>
<th>FracRight</th>
</tr>
</thead>
</table>
| age+hours      | bayesclass | density=joint submodel=gauss
              |           | gausstype=general
|                |            | 0.760452 +/- 0.00319521                |           |
| age+hours+edunum | bayesclass | density=joint submodel=gauss
              |           | gausstype=general
|                |            | 0.796513 +/- 0.00542432                |           |
| a+h+e+capgain  | bayesclass | density=joint submodel=gauss
              |           | gausstype=general
|                |            | 0.793518 +/- 0.00319241                |           |
| a+h+e+c+laxweight | bayesclass | density=joint submodel=gauss
              |           | gausstype=general
|                |            | 0.793477 +/- 0.00321524                |           |
Overfitting dangers

• Problemette with Gaussian Bayes classifier: 
  \[\text{\#parameters quadratic with \#dimensions.}\]
  With 10,000 dimensions and only 1,000 datapoints we could overfit.

Question: Any suggested solutions?
General: $O(m^2)$ parameters

$$\Sigma = \begin{pmatrix}
\sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1m} \\
\sigma_{12} & \sigma_2^2 & & \sigma_{2m} \\
& \vdots & \ddots & \vdots \\
\sigma_{1m} & \sigma_{2m} & & \sigma_m^2
\end{pmatrix}$$
General: $O(m^2)$ parameters

$$\Sigma = \begin{pmatrix}
\sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1m} \\
\sigma_{12} & \sigma_2^2 & \cdots & \sigma_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{1m} & \sigma_{2m} & \cdots & \sigma_m^2
\end{pmatrix}$$
Alined: $O(m)$ parameters

$$
\Sigma = \begin{pmatrix}
\sigma_1^2 & 0 & 0 & \cdots & 0 & 0 \\
0 & \sigma_2^2 & 0 & \cdots & 0 & 0 \\
0 & 0 & \sigma_3^2 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \sigma_{m-1}^2 & 0 \\
0 & 0 & 0 & \cdots & 0 & \sigma_m^2 \\
\end{pmatrix}
$$
Aligned: $O(m)$ parameters

$$
\Sigma = \begin{pmatrix}
\sigma_1^2 & 0 & 0 & \ldots & 0 & 0 \\
0 & \sigma_2^2 & 0 & \ldots & 0 & 0 \\
0 & 0 & \sigma_3^2 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \sigma_{m-1}^2 & 0 \\
0 & 0 & 0 & \ldots & 0 & \sigma_m^2 \\
\end{pmatrix}
$$
Spherical: $O(1)$

cov parameters

\[
\Sigma = \begin{pmatrix}
\sigma^2 & 0 & 0 & \cdots & 0 & 0 \\
0 & \sigma^2 & 0 & \cdots & 0 & 0 \\
0 & 0 & \sigma^2 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & \sigma^2 & 0 \\
0 & 0 & 0 & \cdots & 0 & \sigma^2 \\
\end{pmatrix}
\]
Spherical: $O(1)$

cov parameters

\[
\Sigma = \begin{pmatrix}
\sigma^2 & 0 & 0 & \ldots & 0 & 0 \\
0 & \sigma^2 & 0 & \ldots & 0 & 0 \\
0 & 0 & \sigma^2 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \ldots & \sigma^2 & 0 \\
0 & 0 & 0 & \ldots & 0 & \sigma^2
\end{pmatrix}
\]
Acknowledgment

These slides are based in part on slides from previous machine learning classes taught by Andrew Moore at CMU and Tommi Jaakkola at MIT. I thank Andrew and Tommi for letting me use their slides.