Pattern Recognition

Examples
Matlab Decision Trees

- The function calls in the latest version of Matlab are slightly different than what is shown in the lecture.

<table>
<thead>
<tr>
<th>Old</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a tree</td>
<td><code>tree = treefit(X, y)</code></td>
</tr>
<tr>
<td>Display tree</td>
<td><code>treedisp(tree)</code></td>
</tr>
<tr>
<td>Classify a point</td>
<td><code>treeeval(tree, x)</code></td>
</tr>
<tr>
<td>Minimum # points to split a node</td>
<td>‘splitmin’</td>
</tr>
</tbody>
</table>
clear all
close all

% Loads:
% meas(150,4) - each row is a pattern (a 4-dimensional vector)
% species{150} - each element is the name of a flower
load fisheriris

% Create a vector of class numbers. We know that the input data is grouped
% so that 1..50 is the 1st class, 51..100 is the 2nd class, 101..150 is the
% 3rd class.
y(1:50,1) = 1;     % class 'setosa'
y(51:100,1) = 2;    % class 'versico'
y(101:150,1) = 3;    % class 'virginica'

X = meas(:, 1:2);       % just use first 2 features (easier to visualize)

% We will just use the first 2 features, since it is easier to visualize.
% However, when we do that there is a chance that some points will be
% duplicated (since we are ignoring the other features). If so, just keep
% the first point.
indicesToKeep = true(size(X,1),1);
for i=1:size(X,1)
    % See if we already have the ith point.
    if any((X(i,1)==X(1:i-1,1)) & (X(i,2)==X(1:i-1,2)))
        indicesToKeep(i) = false;   % Skip this point
    end
end
X = X(indicesToKeep, :);
y = y(indicesToKeep);
Iris Example (continued)

% Let's take out a data point to use for testing. The rest we will
% use for training.
 t = randi(size(X,1));       % Pick a point at random
 xTest = X(t,:);
yTest = y(t);

% Remove that point from the training data. Note - if there are
% duplicates, remove those too.
 indicesTraining = (X(:,1)~=xTest(1)) | (X(:,2)~=xTest(2));
 X = X(indicesTraining, :);
y = y(indicesTraining);

% Plot the feature vectors.
 figure
 hold on
 plot(X(y==1,1), X(y==1,2), '*r');
 plot(X(y==2,1), X(y==2,2), '*g');
 plot(X(y==3,1), X(y==3,2), '*b');
xlabel('Sepal length'), ylabel('Sepal width');

plot(xTest(1), xTest(2), 'ok');   % black is the test point
hold off
% A "classification" tree produces classification decisions that are 
% "nominal" (ie, names). A "regression" tree produces classification 
% decisions that are numeric.

ctree = ClassificationTree.fit(X, y, ... 
    'MinParent', 10); % default is 10 
view(ctree); % Prints a text description 
view(ctree, 'mode', 'graph'); % Draws a graphic description of the tree
% Visualize the space of input data, and what class each belongs to.
xmin = min(X(:,1));
xmax = max(X(:,1));
ymin = min(X(:,2));
ymax = max(X(:,2));

hold on;
dx = (xmax-xmin)/40;
dy = (ymax-ymin)/40;
for x=xmin:dx:xmax
    for y=ymin:dy:ymax
        class = predict(ctree, ...
            [x y]);
        if class==1
            plot(x,y,'.r');
        elseif class==2
            plot(x,y,'.g');
        else
            plot(x,y,'.b');
        end
    end
end
hold off;

% Now classify the test point, using the decision tree.
class = predict(ctree, xTest);
fprintf('Test point is classified as %d, true value is %d\n', class, yTest);
Example 2 - Nearest Neighbor Classifier

• Find the $k$ nearest neighbors to the query point, in the training data.
• Take the majority vote as the classification decision.

Example: using closest $k=5$ neighbors, majority vote is blue
% Find distances from the test point to all other points.
distances = ((xTest(1)-X(:,1)).^2 + (xTest(2)-X(:,2)).^2).^0.5;

% Sort in ascending order, and save indices of sorted points.
[distSorted, indices] = sort(distances, 'ascend');

k = 5;
fprintf('The %d closest neighbors are classes:
', k);
hold on
for i=1:k
    fprintf(' class %d
', y(indices(i)));
    % Show point on the plot.
    xn = X(indices(i),:);
    plot(xn(1), xn(2), 'dk');
end
% Get majority vote (ie, the mode) among neighbors.
class = mode(y(indices(1:k)));
fprintf('Test point is classified as %d, true value is %d
', class, yTest);

The 5 closest neighbors are classes:
class 3
class 3
class 2
class 3
class 3
Test point is classified as 3, true value is 2
clear all
close all

% Loads:
% meas(150,4) - each row is a pattern (a 4-dimensional vector)
% species{150} - each element is the name of a flower
load fisheriris

% Create a vector of class numbers. We know that the input data is grouped
% so that 1..50 is the 1st class, 51..100 is the 2nd class, 101..150 is the
% 3rd class.
y(1:50,1) = 1; % class 'setosa'
y(51:100,1) = 2; % class 'versico'
y(101:150,1) = 3; % class 'virginica'

X = meas(:, 2:3); % just use 2 of the features (easier to visualize)

% We will just use 2 of the features, since it is easier to visualize.
% However, when we do that there is a chance that some points will be
% duplicated (since we are ignoring the other features). If so, just keep
% the first point.
indicesToKeep = true(size(X,1),1);
for i=1:size(X,1)
    % See if we already have the ith point.
    if any((X(i,1)==X(1:i-1,1)) & (X(i,2)==X(1:i-1,2)))
        indicesToKeep(i) = false; % Skip this point
    end
end
X = X(indicesToKeep, :);
y = y(indicesToKeep);

Use iris data again, but this time use features 2 & 3
Classification using nearest class mean

% Let's take out a data point to use for testing. The rest we will
% use for training.
% t = randi(size(X,1));       % Pick a point at random
xTest = X(t,:);
yTest = y(t);

% Remove that point from the training data. Note - if there are
% duplicates, remove those too.
indicesTraining = (X(:,1)~=xTest(1)) | (X(:,2)~=xTest(2));
X = X(indicesTraining, :);
y = y(indicesTraining);

% Separate out the training points into the 3 classes.
X1 = X(y==1,:);
X2 = X(y==2,:);
X3 = X(y==3,:);

% Get mean of each class.
m1 = sum(X1)/length(X1);
m2 = sum(X2)/length(X2);
m3 = sum(X3)/length(X3);

% Plot the feature vectors.
figure
hold on
plot(X1(:,1), X1(:,2), '.r');   % Data points
plot(X2(:,1), X2(:,2), '.g');
plot(X3(:,1), X3(:,2), '.b');
plot( m1(1), m1(2), 'sr' );     % Class means
plot( m2(1), m2(2), 'sg' );
plot( m3(1), m3(2), 'sb' );
plot(xTest(1), xTest(2), 'ok');  % black is the test point
hold off
Classification using nearest class mean

- The Mahalanobis distance of feature vector \( \mathbf{x} \) to the \( i \)th class is

\[
d_i = \sqrt{(\mathbf{x} - \mathbf{z}_i)^T \mathbf{C}_i^{-1} (\mathbf{x} - \mathbf{z}_i)}
\]

- where \( \mathbf{C}_i \) is the covariance matrix of the feature vectors in the \( i \)th class

% Get Mahalanobis distance to each class mean.
d(1) = mahal(xTest, X1);
d(2) = mahal(xTest, X2);
d(3) = mahal(xTest, X3);
[dm, c] = min(d);

% Classify test point to the nearest class mean.
fprintf('Test point classified as class %d.
', c);
fprintf('True class is class %d.
', yTest);
fprintf('Mahalanobis distance = %f.
', dm);
Example 4 – Shape classification

• On the course web page are two binary images, “hearts.bmp” and “plus.bmp”. These are images of hand-drawn shapes of hearts and pluses, respectively.

• Extract connected components from these images, and compute feature properties for each image, using the Matlab function regionprops.
Example 4 (continued)

• Plot the feature “Area” vs “Solidity” for each of the two classes (using Matlab’s plot function). You will need to put those features into a 2xN array:

```matlab
for i=1:n1
    X(i,1) = props1(i).Area;
    X(i,2) = props1(i).Solidity;
    y(i,1) = 1;   % class label
end
for i=1:n2
    X(i+n1,1) = props2(i).Area;
    X(i+n1,2) = props2(i).Solidity;
    y(i+n1,1) = 2; % class label
end
```
Example 4 (continued)

• Create a decision tree for these two classes, using the Matlab function `ClassificationTree.fit`.
  – Make sure that you set the input parameter “MinParent” to 1 (the default is 10).
  – Setting MinParent =1 will cause the decision tree to split (make a new node) if there are any instances that are still not correctly labeled.

• Draw the resulting decision tree (using the function `view`):
  ```matlab
  view(ctree, 'mode', 'graph');
  ```

• Apply the tree to classify the new shape in the image “test.bmp”. Which class do you assign this shape to?