

# HW4: Skin Finding

Assigned: Thursday, October 23

Due: Thursday, October 30



face image



segmented by color



skin pixels highlighted

# Details: Skin Classification

- $r = R/(R+G+B)$
- $g = G/(R+G+B)$
- Start with the face training image set. Run your K-means algorithm on the face training set to get K clusters with small K, ie  $K < 9$ , represented by average  $(r,g)$ .
- Represent each cluster by its mean in color space, ie  $(r_{\text{mean}}, g_{\text{mean}})$ .
- Use the groundtruth images to assign the true label (skin or not) of each cluster. The pixels of the cluster can be used to vote for the final label of the whole cluster.

# Sample ARFF

## Attribute Relation File Format

```
@relation skin  
@attribute 0 numeric  
@attribute 1 numeric  
@attribute class {0,1}  
@data  
129.046013 128.952284 0  
151.097297 146.511336 0  
148.497444 162.167033 0  
144.097158 144.516707 1  
128.101831 128.544512 0  
121.993783 91.654521 0  
145.180214 143.650830 1  
133.587150 133.247295 0  
132.382377 117.473634 0  
153.213978 143.872653 1  
128.760509 128.093794 0  
143.193117 136.694335 1  
125.730460 126.652810 0  
131.812171 142.671438 0
```

relation name

feature names and types

class name and type

your clusters with the means  
(from k-means) and skin class  
assignments (from ground truth)

Note: append all clusters from  
training images into one big training  
ARFF

# Continued

- Train a WEKA classifier to learn skin vs. nonskin color in (r,g) space. Your training vectors will have (r,g) plus class for each of the clusters in the training set. Start with the Naïve Bayes classifier. You may try others, such as SVM.
- Run your skin finder on images from both the training and testing set, feeding it cluster (r,g) vectors to be classified.
- Report on its performance: pixel classification accuracy plus images.
- In the report, make a table with the following columns (see the report template)

Image Name	Orig. Image	Labeled Image	N.B. accuracy	Other accuracy

# Required Test Images

from  
training  
set



face01.ppm



face04.ppm



face05.ppm



face08.ppm

from test  
set

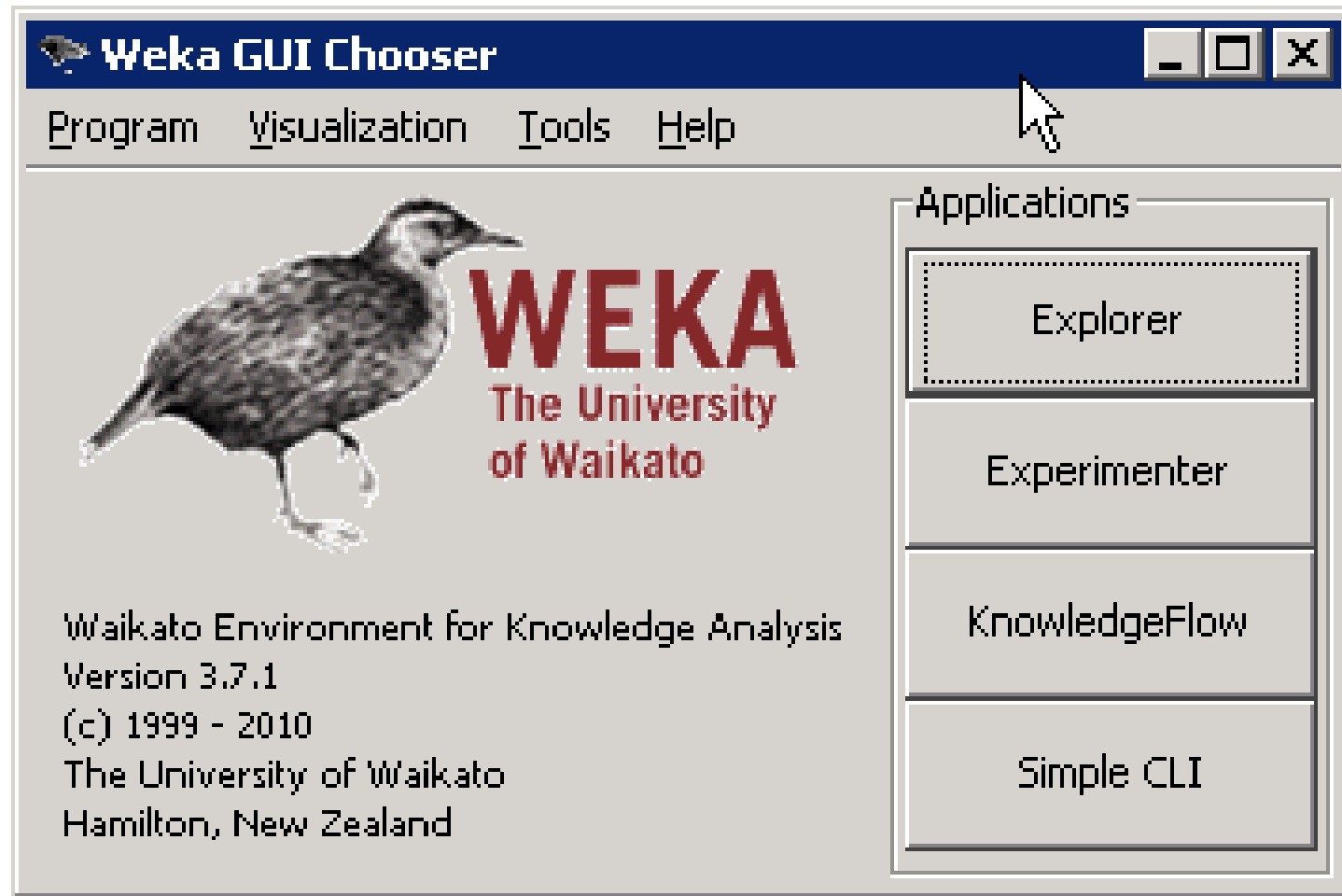


face23.ppm

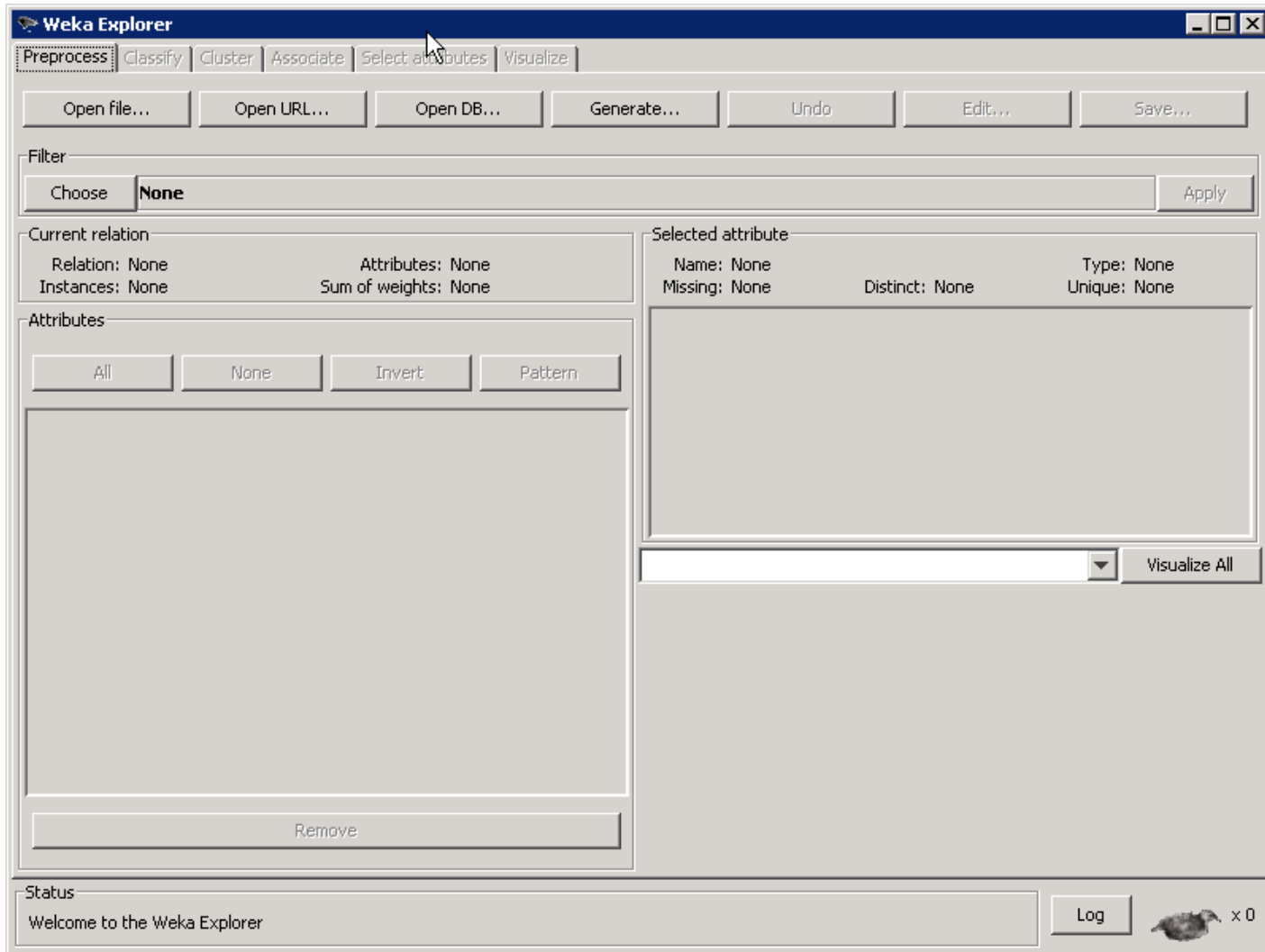


face28.ppm

# Weka Instructions



# Explorer



# Loading Training Data

**Weka Explorer**

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... | Open URL... | Open DB... | Generate... | Undo | Edit... | Save...

Filter: Choose **None** Apply

Current relation: Relation: skin, Instances: 45, Attributes: 3, Sum of weights: 45

Attributes: All | None | Invert | Pattern

No.	Name
1	0
2	1
3	class

Remove

Selected attribute: Name: 0, Missing: 0 (0%), Distinct: 45, Type: Numeric, Unique: 45 (100%)

Statistic	Value
Minimum	113.305
Maximum	176.513
Mean	135.611
StdDev	12.507

Class: class (Nom) Visualize All

113.3 144.91 176.51

Status: OK Log x 0



# Classification

The image shows the Weka Explorer interface with the 'Classifier' tab selected. The 'NaiveBayes' classifier is chosen. The 'Test options' section is visible, showing 'Supplied test set' selected. The 'Classifier output' area is empty. A dialog box titled 'Classifier evaluation options' is open, showing various checkboxes and input fields for configuring the classifier's output and evaluation settings.

**Classifier**  
Choose **NaiveBayes**

**Test options**

- Use training set
- Supplied test set
- Cross-validation Folds
- Percentage split %

(Nom) class

**Classifier output**

Result list (right-click for options)

**Classifier evaluation options**

- Output model
- Output per-class stats
- Output entropy evaluation measures
- Output confusion matrix
- Store predictions for visualization
- Output predictions

Output additional attributes

Cost-sensitive evaluation

Random seed for XVal / % Split

Preserve order for % Split

Output source code

Status  
OK  x 0

# Prediction

The screenshot shows the Weka Explorer interface with the NaiveBayes classifier selected. The 'Classifier output' window displays the following table:

inst#	actual	predicted	error	probability distribution
1	2:1	2:1		0.069 *0.931
2	1:0	1:0		*1 0
3	1:0	1:0		*1 0
4	2:1	2:1		0.084 *0.916
5	1:0	1:0		*0.972 0.028
6	1:0	1:0		*0.981 0.019
7	1:0	2:1	+	0.084 *0.916
8	1:0	1:0		*1 0
9	1:0	1:0		*0.999 0.001
10	2:1	2:1		0.289 *0.711
11	1:0	1:0		*1 0
12	1:0	2:1	+	0.258 *0.742
13	1:0	1:0		*0.88 0.12
14	1:0	1:0		*1 0
15	2:1	2:1		0.21 *0.79

The interface also shows test options (Use training set, Supplied test set, Cross-validation, Percentage split) and a result list containing '16:43:17 - bayes.NaiveBayes'. The status bar at the bottom indicates 'OK' and has a 'Log' button.

# Confusion Matrix

The screenshot shows the Weka Explorer interface with the NaiveBayes classifier selected. The 'Classifier output' pane displays the following performance metrics:

- Root mean squared error: 0.3023
- Relative absolute error: 39.9303 %
- Root relative squared error: 64.8327 %
- Total Number of Instances: 45

The 'Detailed Accuracy By Class' section shows the following table:

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.903	0.143	0.933	0.903	0.918	0.929	0
	0.857	0.097	0.8	0.857	0.828	0.929	1
Weighted Avg.	0.889	0.129	0.892	0.889	0.89	0.929	

The 'Confusion Matrix' section shows the following results:

```
=== Confusion Matrix ===  
  
 a b  <-- classified as  
28  3 | a = 0  
 2 12 | b = 1
```

The 'Test options' section shows the following settings:

- Use training set:
- Supplied test set:  (Set...)
- Cross-validation:  (Folds: 10)
- Percentage split:  (%: 66)

The 'Result list' shows a single entry: 16:43:17 - bayes.NaiveBayes.

The 'Status' bar at the bottom shows 'OK' and a 'Log' button.