Pattern Recognition and Machine Learning

James L. Crowley

ENSIMAG 3 - MMIS
Lesson 2

Fall Semester 2017
11 October 2017

Face Detection using Color Histograms

Outline

Notation 2

1. Lab 1: Face Detection using Color 3
   1.1 Algorithm Overview 3
   1.2 Test Data 5
   1.3 Evaluation of Performance 7
   1.4 Grading for Lab project 1. 7

2. Performance Metrics for 2 Class Detectors............. 8
   2.1 ROC Curves 8
   2.2 True Positives and False Positives 8
   2.3. Comparing ROC curves 10

Notation

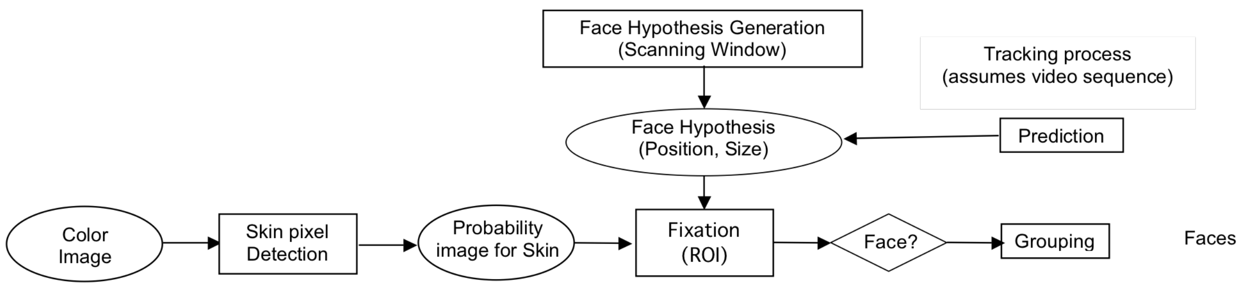
$\vec{y}$	The true class for an observation $\vec{X}$
$\{\vec{X}_m\}$ $\{y_m\}$	Training samples for learning.
$y(\vec{X}_m)$	An annotation (or ground truth) function $y(\vec{X}_m) \in \{P, N\}$
$g(\vec{X}_m)$	Discriminant function. $0 \le g(\vec{X}_m) \le 1$
M	The number of training samples.
M_T	The number of training samples in the target class
$h(\vec{X})$	A multidimensional histogram of for $\vec{X}$
Q	The number of cells in a histogram

1. Lab 1: Face Detection using Color

Skin color can be used to construct a simple detector for skin pixels in images. Color skin pixels can then be used to detect and track "blobs" that represent faces, hands and other skin colored regions in images.

1.1 Algorithm Overview

The detector works by first computing the probability that each pixel contains skin. A sliding window (Region of Interest or ROI) is then scanned over the image. At each position, a weighted sum of probabilities is determined. Regions for which the weighted sum is above threshold are detected as faces. Adjacent face detections are grouped to form a single face detection.



Algorithm:

1) Compute probability of skin at each pixel.
2) Test for faces at possible positions and sizes (scanning Window).
3) Cluster adjacent detections.
4) Estimate final face parameters from clusters of detections.

Summary of the algorithm:

1. Color Skin detection

A color (RGB) image, *C(i,j)* is transformed into an image where each pixel provides an estimate of the probability of skin, *P(i,j)*.

Assume a color image : $C(i,j) = \begin{pmatrix} R \\ G \\ B \end{pmatrix}(i,j)$

The algorithm will use a lookup table to convert color to probability.

$$P(i,j) \leftarrow L(R,G,B)$$

The lookup table is constructed as a ratio of histograms, as explained below.
We can improve the results can be obtained by using a normalized color space as well as by tuning the quantization of the histogram to the data.

2) Face Detection

We will assume faces are vertical. Hypotheses for the presence of a face at a particular position and size can tested as the sum of skin probabilities with a region (Region of Interest or ROI) at a position (c_i, c_j) and size (width, height). A simple ROI can be defined as a vector of four corners: (top, left, bottom, right) or *(t,b,l,r)*.

Let us define a face hypothesis as a ROI: $\vec{X}_n = \begin{pmatrix} t \\ l \\ b \\ r \end{pmatrix}$

The likelihood of a face at the ROI is the average probability:

$$F(\vec{X}_n) = \frac{1}{(t-b)(l-r)} \sum_{i=l}^{r} \sum_{j=t}^{b} P(i,j)$$

If $F(\vec{X}_n) \geq$ Threshold then Face else NOT Face.

We can bias the detection by adding a bias term B.

We can improve detection by weighting the probabilities with a Gaussian or face shaped mask. This is described below.

3) Scanning window detector.

A scanning window detector systematically tests hypotheses over a range of positions and sizes. This can be made more efficient by a form of hierarchical search.

4) Grouping adjacent detections:

When a face is present, it will be detected at multiple adjacent positions and sizes. The best face position can be obtained by "grouping" adjacent detections. This is discussed below.

Demonstration Video

1.2 Test Data

For this exercise we will use the FDDB (Face Detection Data Set and Benchmark) data base maintained at UMASS: http://vis-www.cs.umass.edu/fddb/

This data-base was constructed for face detection and not for skin detection.
Face regions have been hand-labeled as both boxes and ellipses.

All images are RGB with each pixel containing 3 colors: Red, Green and Blue.

We will use the ellipses as ground truth for skin regions. A typical image with and annotated face regions as an ellipse looks like.

Any pixel inside the ellipse can be considered skin and included in the skin color histogram.

Note that there are skin pixels that are NOT in the ellipse (hand, ears, neck etc), and there are non-skin pixels that ARE in the face (hair, teeth, etc). This will lead to minor errors in the detection. Your job will be to measure the impact of these errors in building a face detector using detection of skin pixels.

Note that faces appear with an elliptical form, with major axis in the vertical direction. Annotations of face regions in FDDB are represented as an elliptical region, denoted by a 6-tuple (r_a, r_b, θ, c_x, c_y, 1) where r_a and r_b refer to the half-length of the major and minor axes, θ is the angle of the major axis with the horizontal axis, and c_x and c_y are the column and row image coordinates of the center of this ellipse.

Ellipse Data:
2002/07/24/big/img_82
1
59.268600 35.142400 1.502079 149.366900 59.365500 1

the standard form of an ellipse with a major axis along the horizontal (x) axis is:

$$\frac{(x-c_x)^2}{r_a^2}+\frac{(y-c_y)^2}{r_b^2}=1$$

for any pixel x,y inside the ellipse, $\frac{(x-c_x)^2}{r_a^2}+\frac{(y-c_y)^2}{r_b^2}<1$

For a hypothesis of a face $\vec{X}=\begin{pmatrix} c_x \\ c_y \\ r_a \\ r_b \\ \theta \end{pmatrix}$ can define a ground truth function as $y(\vec{X}_m)$

$$y(\vec{X}_m)=if\left(\frac{(x-c_x)^2}{r_a^2}+\frac{(y-c_y)^2}{r_b^2}\leq 1\right) \text{ then P else N.}$$

If it is necessary to rotate the face to an angle θ we can use:

$$\frac{\left((x-c_x)\cos(\theta)+(y-c_y)\sin(\theta)\right)^2}{r_a^2}+\frac{\left((x-c_x)\sin(\theta)+(y-c_y)\cos(\theta)\right)^2}{r_b^2}=1$$

Face hypotheses can be limited to a single size or tested over a range of sizes. We can use the major and minor ellipses to define the range of height and width over which we need to find faces.

1.3 Evaluation of Performance

As we have seen, there are many possible variations for the algorithm. These include:
1) Different color codings (RGB, normalized chrominance, etc).
2) Different detection algorithms (raw sum of probabilities, wighted sum, face mask, etc).
3) Different algorithms for clustering adjacent detections.
4) Variations in selection of training and test data (N-Fold, leave one out, etc).

Your job is to measure the variations in performance of different detectors and to explain the differences.

What constitutes as a TRUE face detection? For each image you will need a function that tells if a face hypothesis is in any of the faces.

1.4 Grading for Lab project 1.

The objective of this project is to evaluate the effectiveness of face detection using color. Evaluations of variations in the algorithm will be performed using ROC curves that plot True Positive Rate vs False Positive Rate.

Each programming team should
1) Train a set of skin pixel from sets of folds from the test data.
2) Construct a sliding window face detector that sum probabilities in a ROI and decides Face/No Face for each position and size.
3) Plot ROC curves for the detectors using folds that were not used in training
4) Interpret the results, describing the effectiveness of the detectors and explaining the sources of errors.

A grading scale that is published with the exercise. Minimal effort gets a minimal grade. Gain additional points by trying different variations.

Lab work will be reported with a written report in either French or English. Work will be evaluated based on the effectiveness of the experimental evaluations, and the clarity and depth of the explanation of experimental results. Written reports are dues on Thursday 15 November.

2. Performance Metrics for 2 Class Detectors

2.1 ROC Curves

Two-class classifiers have long been used for signal detection problems in communications and have been used to demonstrate optimality for signal detection methods. The quality metric that is used is the Receiver Operating Characteristic (ROC) curve. This curve can be used to describe or compare any method for signal or pattern detection.

The ROC curve is generated by adding a variable Bias term to a discriminant function.

$$R(\vec{X}) = d(g(\vec{X}) + B)$$

and plotting the rate of true positive detection vs false positive detection where $R(\vec{X}_m)$ is the classifier as in lesson 1. As the bias term, B, is swept through a range of values, it changes the ratio of true positive detection to false positives.

For a ratio of histograms, $g(\vec{X}_m)$ is a probability ranging from 0 to 1.
B can range from less than *–0.5* to more than *+0.5*.
When $B < -0.5$ all detections will be Negative.
When $B > +0.5$ all detections will be Positive.
Between *–0.5* and *+0.5* $R(\vec{X})$ will give a mix of Positive and Negative results.

The bias term, B, can act as an adjustable gain that sets the sensitivity of the detector. The bias term allows us to trade False Positives for False Negatives.

The resulting curve is called a Receiver Operating Characteristics (ROC) curve.
The ROC plots True Positive Rate (TPR) against False Positive Rate (FPR) as a function of B for the test data $\{\vec{X}_m\}$ with ground truth $\{y_m\}$.

2.2 True Positives and False Positives

For each training sample, the detection as either Positive (P) or Negative (N)

IF $g(\vec{X}_m)+B > 0.5$ THEN $R(\vec{X}_m) = P$ else $R(\vec{X}_m) = N$

The detection can be TRUE (T) or FALSE (F) depending on the indicator variable y_m

IF $R(\vec{X}_m) = y_m$ THEN T else F

Combining these two values, any detection can be a True Positive (TP), False Positive (FP), True Negative (TN) or False Negative (FN).

For the M samples of the test data $\{\vec{X}_m\}$, $\{y_m\}$ we can define:

#P as the number of Positives,
#N as the number of Negatives,
#T as the number of True and
#F as the number of False,

From this we can define:

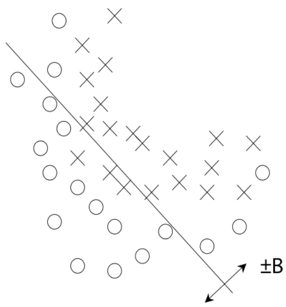
#TP as the number of True Positives,
#FP as the number of False Positives,
#TN as the number of True Negative,
#FN as the number of False Negatives.

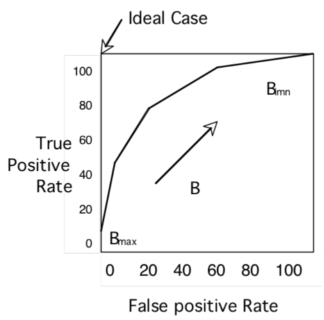
Note that #P = #TP + #FN and #N = #FP+ #TN

The True Positive Rate (TPR) is $TPR = \frac{\#TP}{\#P} = \frac{\#TP}{\#TP + \#FN}$

The False Positive Rate (FPR) is $FPR = \frac{\#FP}{\#N} = \frac{\#FP}{\#FP + \#TN}$

The ROC plots the TPR against the FPR as a bias B is swept through a range of values.





When B is less than –0.5, all the samples are detected as N, and both the TPR and FPR are 0. As B increases both the TPR and FPR increase. Normally TPR should rise monotonically with FPR. If TPR and FPR are equal, then the detector is no better than chance. If TPR ≤ FPR then the detector is worse than chance.

The closer the curve approaches the upper left corner, the better the detector.

		$y_m = R(\vec{X}_m)$	
		T	F
$d(g(\vec{X}_m)+B > 0.5)$	P	True Positive (TP)	False Positive (FP)
	N	True Negative (TN)	False Negative (FN)

2.3. Comparing ROC curves

Area Under the Curve.

AUC is an abbreviation for area under the curve. AUC is used to determine which of the used models predicts the classes best. AUC=0.5 is a chance detector. AUC=1.0 is a perfect detector.

Product of FPR and FNR.

The area of the smallest rectangle between the ROC curve and the upper left corner is another measure to compare detectors. This rectangle is

$$A = FNR \cdot FPR = (1 - TPR) \cdot FPR$$

Sum of FPR and FNR.

Some authors have also used the sum of FPR + FNR as an approximation to the overall error rate.