Supervised Segmentation: Pixel Classification
Example: A Classification Problem

- Categorize images of fish—say, “Atlantic salmon” vs. “Pacific salmon”
- Use features such as length, width, lightness, fin shape & number, mouth position, etc.

Steps
1. Preprocessing (e.g., background subtraction)
2. Feature extraction
3. Classification
Bayes Risk

Some errors may be inevitable: the minimum risk (shaded area) is called the Bayes risk.

Probability density functions (area under each curve sums to 1)
Discriminative vs Generative Models

Finding a decision boundary is not the same as modeling a conditional density.
Bayes theorem

\[ P(x, y) = P(x|y) \ P(y) \]
so
\[ P(x|y) \ P(y) = P(y|x) \ P(x) \]
and
\[ P(x|y) = P(y|x) \ P(x) / P(y) \]

The parameters you want to estimate
What you observe
Likelihood function
Prior probability
Constant w.r.t. parameters x.
Loss functions in classifiers

• Loss
  – some errors may be more expensive than others
    • e.g. a fatal disease that is easily cured by a cheap medicine with no side-effects -> false positives in diagnosis are better than false negatives
  – We discuss two class classification: $L(1 \rightarrow 2)$ is the loss caused by calling 1 a 2

• Total risk of using classifier $s$

  \[ R(s) = Pr \{1 \rightarrow 2|\text{using } s\} L(1 \rightarrow 2) + Pr \{2 \rightarrow 1|\text{using } s\} L(2 \rightarrow 1) \]
Histogram based classifiers

• Use a histogram to represent the class-conditional densities
  – (i.e. p(x|1), p(x|2), etc)

• Advantage: Estimates converge towards correct values with enough data

• Disadvantage: Histogram becomes big with high dimension so requires too much data
  – but maybe we can assume feature independence?
Example Histograms
Kernel Density Estimation

- **Parzen windows**: Approximate probability density by estimating local density of points (same idea as a histogram)
  - Convolve points with window/kernel function (e.g., Gaussian) using scale parameter (e.g., sigma)
Density Estimation at Different Scales

- Example: Density estimates for 5 data points with differently-scaled kernels
- Scale influences accuracy vs. generality (overfitting)

from Duda et al.
Example: Kernel Density Estimation Decision Boundaries

Smaller

Larger

from Duda et al.
Application: Skin Colour Histograms

• Skin has a very small range of (intensity independent) colours, and little texture
  – Compute colour measure, check if colour is in this range, check if there is little texture (median filter)
  – Get class conditional densities (histograms), priors from data (counting)

• if \( p(\text{skin}|x) > \theta \), classify as skin
• if \( p(\text{skin}|x) < \theta \), classify as not skin
• if \( p(\text{skin}|x) = \theta \), choose classes uniformly and at random
Skin Colour Models

Skin chrominance points

Smoothed, [0,1]-normalized
courtesy of G. Loy

PSY 5018H: Math Models Hum Behavior, Prof. Paul Schrater, Spring 2005
Skin Colour Classification

For every pixel $p_i$ in $I_{\text{test}}$

- Determine the chrominance values $(a_i, b_i)$ of $I_{\text{test}}(p_i)$
- Lookup the skin likelihood for $(a_i, b_i)$ using the skin chrominance model.
- Assign this likelihood to $I_{\text{skin}}(p_i)$
Statistical Color Models with Application to Skin Detection

Michael J. Jones and James M. Rehg

(a) 2-D rendering of 3-D histogram model viewed along the green-magenta axis.

(b) Surface plot of the marginal density formed by integrating along the viewing direction in (a).
$\frac{P(\text{rgb|skin})}{P(\text{rgb|¬skin})} \geq \Theta,$

(a) Contour plot for skin model, marginalized along the green-magenta axis.

(b) Contour plot for skin model, marginalized along the gray axis.
Results

Figure from “Statistical color models with application to skin detection,” M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE

PSY 5018H: Math Models Hum Behavior, Prof. Paul Schrater, Spring 2005
ROC Curves (Receiver operating characteristics)

Plots trade-off between false positives and false negatives

Figure from “Statistical color models with application to skin detection,” M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 copyright 1999, IEEE

PSY 5018H: Math Models Hum Behavior, Prof. Paul Schrater, Spring 2005
Nearest Neighbor Classifier
• Assign label of nearest training data point to each test data point

Voronoi partitioning of feature space for 2-category 2-D and 3-D data

from Duda et al.
K-Nearest Neighbors

• For a new point, find the k closest points from training data
• Labels of the k points “vote” to classify
• Avoids fixed scale choice—uses data itself (can be very important in practice)
• Simple method that works well if the distance measure correctly weights the various dimensions

\( k = 5 \)

Example density estimate

from Duda et al.
Finding faces

• Faces “look like” templates (at least when they’re frontal).

• General strategy:
  – search image windows at a range of scales
  – Correct for illumination
  – Present corrected window to classifier

• Issues
  – How corrected?
  – What features?
  – What classifier?
Naive Bayes (for faces)

• Quantize image patches, then compute a histogram of patch features within a face
• Histogram doesn’t work when there are too many features
  – So, assume they’re independent
    • This can work well if features are chosen to be quite independent
  – Was found to be very effective for face finders

You can try face finders on the face detection home page
http://home.t-online.de/home/Robert.Frischholz face.htm
Results


PSY 5018H: Math Models Hum Behavior, Prof. Paul Schrater, Spring 2005
Pixel-wise classification is not enough
\[ P(\text{horse}) = 0.04 \]
\[ P(\text{background}) = 0.96 \]
Now evaluate

$$\prod_{j=1}^{N_{\text{measurements}}} \frac{p(r_j \mid \text{horse})}{p(r_j \mid \text{background})}$$
Foreground
background
Pixels in color space

background

horse
Histograms

Entire Image
Log(p(r,g|horse)/p(r,g|backgrnd))
Color variations between images reduced via Histogram Matching

Find current histogram and cumulative

Use current cumulative as an inverse transform,
Use desired cumulative as forward transform.
function matchedvalues = histogrammatch(oldvalues, desiredCount, desiredbinvals)
% matchedvalues = histogrammatch(oldvalues, desiredCount, desiredbinvals)
% nonlinearly transform your numbers to enforce a desired histogram (a table of values and counts)
% written by: P. Schrater 2003

[oldcount,oldbinvals]=hist(oldvalues(:),sqrt(length(oldvalues(:))));
% eliminate zero counts and find cumulative table
zind = find(oldcount==0);
oldcount(zind)=[]; oldbinvals(zind)=[];
cumprobold = cumsum(oldcount)/sum(oldcount);

% assign each oldvalue its cumulative prob
pvaluesold = interp1(oldbinvals,cumprobold,oldvalues(:));

% now do same for desired:
% eliminate zero counts and find cumulative table
zind = find(desiredCount==0);
desiredCount(zind)=[]; desiredbinvals(zind)=[];
cumprobnew = cumsum(desiredCount)/sum(desiredCount);

% translate our pvalues back to values by running through the new probability table
matchedvalues = interp1(cumprobnew,desiredbinvals,pvaluesold);

matchedvalues = reshape(matchedvalues,size(oldvalues));
After histogram Equalization
Moral of the story

• You can’t learn much from one picture:
  – One image does not capture variation due to:
    • camera-based color correction
    • Changes in lighting between images
    • Changes in viewpoint and distance between images
  – These sources are extremely important to model
    • Preprocess your images
    • Use large training set.
Intrinsic difficulty segmentation Problem

- In many scenes, correct segmentation based on image cues is not possible - the information is not there
- People make sophisticated use of object and scene information to guide segmentation in difficult problems

![Figure 1: Left to right, three detection tasks of increasing degrees of difficulty. The stop sign (left) is easy to find. The gila monster (centre) is harder. The dalmation dog (right) is almost impossible.](image-url)
Neural networks

• Compose layered classifiers
  – Use a smooth non-linear weighted sum of elements at the previous layer to compute results at next layer
  – Learn all the weights by performing gradient descent (i.e., perform small adjustments to improve results)
Output units

Hidden units

Input units

PSY 5018H: Math Models Hum Behavior, Prof. Paul Schrater, Spring 2005
Training

- Adjust parameters to minimize error on training set
- Perform gradient descent, making small changes in the direction of the derivative of error with respect to each parameter
- Stop when error is low, and hasn’t changed much
- Network itself is designed by hand to suit the problem, so only the weights are learned
The vertical face-finding part of Rowley, Baluja and Kanade’s system

Architecture of the complete system: they use another neural net to estimate orientation of the face, then rectify it. They search over scales to find bigger/smaller faces.

Face Finder: Training

• Positive examples:
  – Preprocess ~1,000 example face images into 20 x 20 inputs
  – Generate 15 “clones” of each with small random rotations, scalings, translations, reflections

• Negative examples
  – Test net on 120 known “no-face” images
Face Finder: Results

• 79.6% of true faces detected with few false positives over complex test set

135 true faces
125 detected
12 false positives
Face Finder Results:
Examples of Misses

from Rowley et al.
Convolutional neural networks

• Template matching using NN classifiers seems to work
• Natural features are filter outputs
  – probably, spots and bars, as in texture
  – but why not learn the filter kernels, too?
A convolutional neural network, LeNet; the layers filter, subsample, filter, subsample, and finally classify based on outputs of this process.

Figure from “Gradient-Based Learning Applied to Document Recognition”, Y. Lecun et al Proc. IEEE, 1998 copyright 1998, IEEE
LeNet is used to classify handwritten digits. Notice that the test error rate is not the same as the training error rate, because the learning “overfits” to the training data.

Figure from “Gradient-Based Learning Applied to Document Recognition”, Y. Lecun et al Proc. IEEE, 1998 copyright 1998, IEEE
Support Vector Machines

• Try to obtain the decision boundary directly
  – potentially easier, because we need to encode only the geometry of the boundary, not any irrelevant wiggles in the posterior.
  – Not all points affect the decision boundary
Support Vectors
Pedestrian Detection with SVMs
Review: Colour

- Spectrum of illuminant and surface
- Human colour perception (trichromacy)
- Metamerism, Grassman’s laws
- RGB and CIE colour spaces
- Uniform colour spaces
- Detection of specularities
- Colour constancy
Review: Invariant features

- Scale invariance, using image pyramid
- Orientation selection
- Local region descriptor (vector formation)
- Matching with nearest and 2\textsuperscript{nd} nearest neighbours
- Object recognition
- Understand homework 3
Review: Classifiers

• Bayes risk, loss functions
• Histogram-based classifiers
• Kernel density estimation
• Nearest-neighbor classifiers
• Neural networks