COSC160: Detection and Classification

Jeremy Bolton, PhD

Assistant Teaching Professor

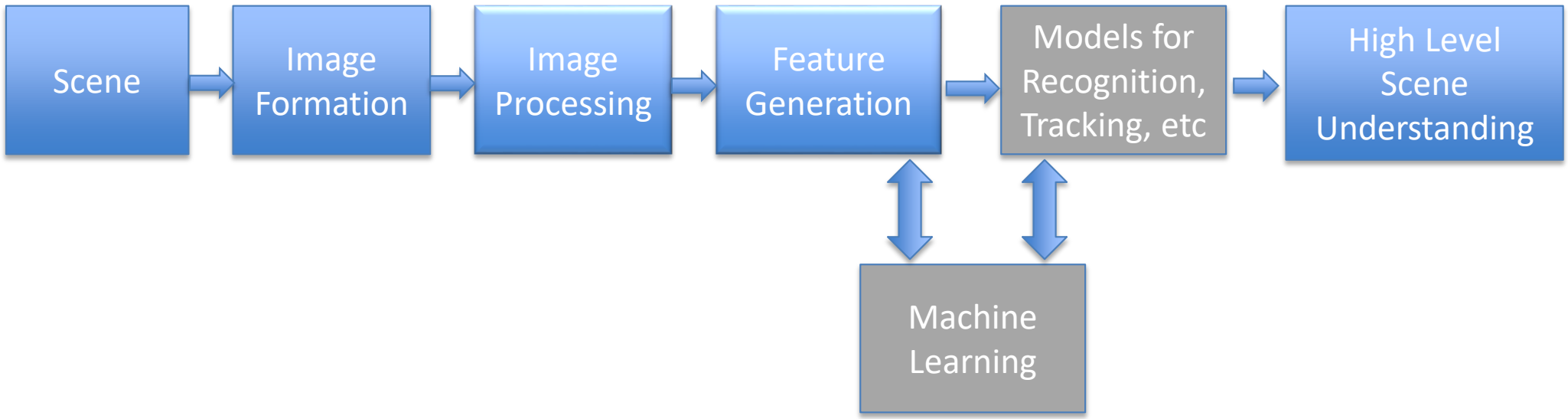
GEORGETOWN UNIVERSITY

Outline

I. Problem
   I. Strategies
   II. Features for training
   III. Using spatial information?
   IV. Reducing dimensionality / feature selection
   V. Machine Learning and Recognition
II. Resulting Maps
III. Post Processing
IV. Quantitative and Qualitative Results
   I. Experimental Design
   II. Repeatability

GEORGETOWN UNIVERSITY

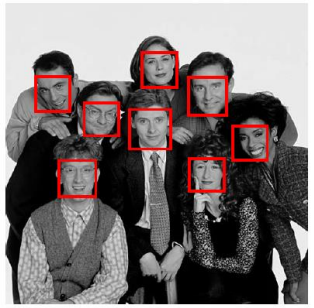
Big Picture

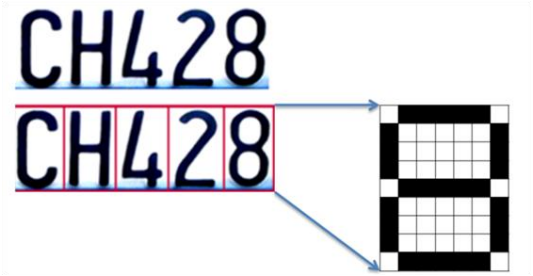

Scene
Image Formation
Image Processing
Feature Generation
Models for Recognition, Tracking, etc
High Level Scene Understanding
Machine Learning

Motivation

- Detection and Classification are common goals in CV applications

- Examples
  - Face Detection (eg in cameras)
  - Face Recognition
  - Object Classification
  - Optical Character Recognition
  - Material Classification





GEORGETOWN UNIVERSITY

Detection vs. Classification

- Detection simply a 2-class classification problem
  - EG: Target vs Background.
  - General discrimination between classes is not necessary
  - Identification of the 1 class is key
- Classification
  - N-class problem
  - Must be able to identify and discriminate between classes.
- Detection vs Recognition
  - EG Face
- The general classification problem is difficult relative to detection.

Problems

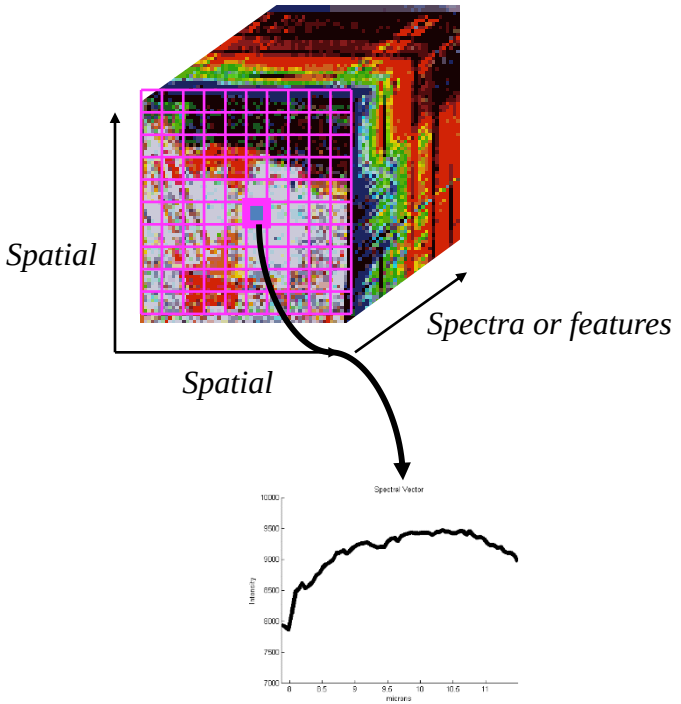
- Image vs Sample.
  - Machine learning and Pattern Recognition Schemes are often used to achieve automation and good results.
    - However, it is not always clear how to optimize models based on an image of observations when most machine learning schemes are based on single observations
  - Scale of classification: pixel – window – image.
- Spatial vs Spectral
- The curse of dimensionality.
- Scale.
  - In image data sets, objects may be observed at different scales.
  - However, standard Pattern Recognition tools will use fixed size-sized windows and/or fixed-sized vectors.

It all starts with the features!

- Garbage in, Garbage out!
- Choose features that effectively distinguish between classes.
- Is classification at the pixel-level or window-level?
  - Is spatial information needed to distinguish between classes
  - Is the instance observation (and features) enough?
- Can the concept be “captured” in one pixel
  - Dependent upon imagery

More observations per pixel should improve results

- Multi-spectral and multi-feature observations
  - If there is multiple spectral or feature observations per pixel, then pixel-based classification may be sufficient.
  - EG Identifying Tree canopy in Satellite imagery
    - Pixel represents multiple square meter swath

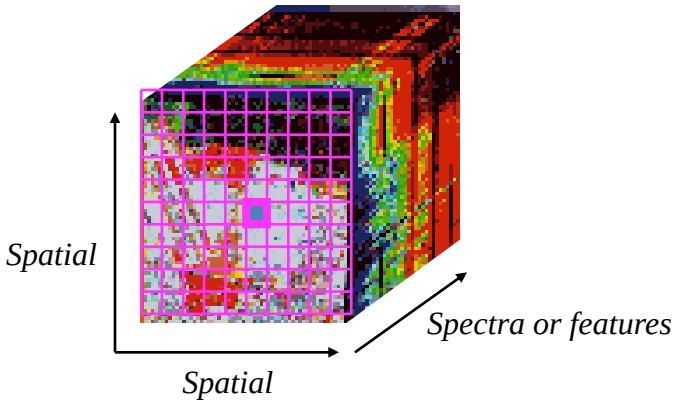


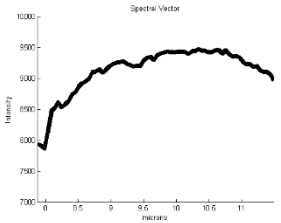
GEORGETOWN UNIVERSITY

Pixel-based Models

- Pros:
  - Able to use standard Machine Learning / Pattern Recognition approaches
    - Based on feature vectors
  - Many approaches well-studied and verified.

- Cons:
  - *May need labels for each pixel when training!*
  - Ignoring spatial information
    - Unless feature computation includes spatial information
  - To have enough discriminative powers, may usually need a large number of observations (high dimensionality)



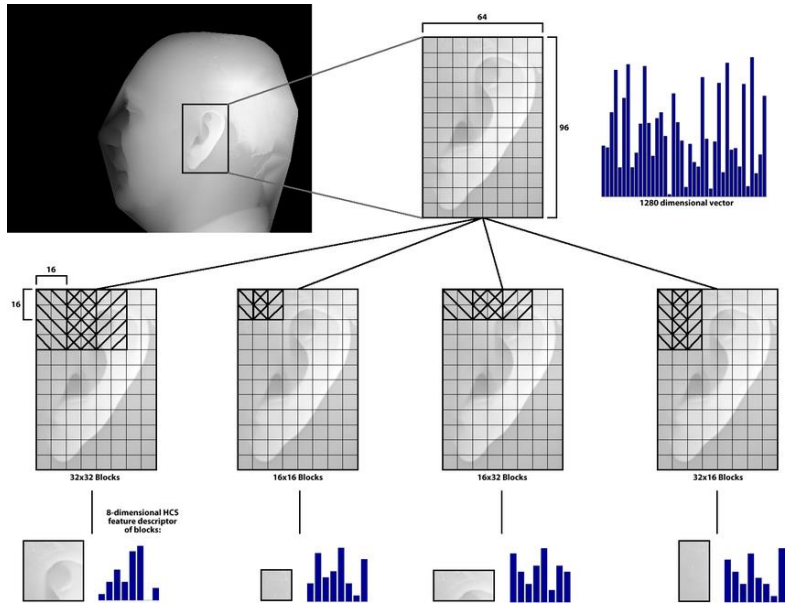


GEORGETOWN UNIVERSITY

"Window"-based Models

- Collections of pixels used for learning and detection / recognition
  - Use spatial information (co-occurrence information)

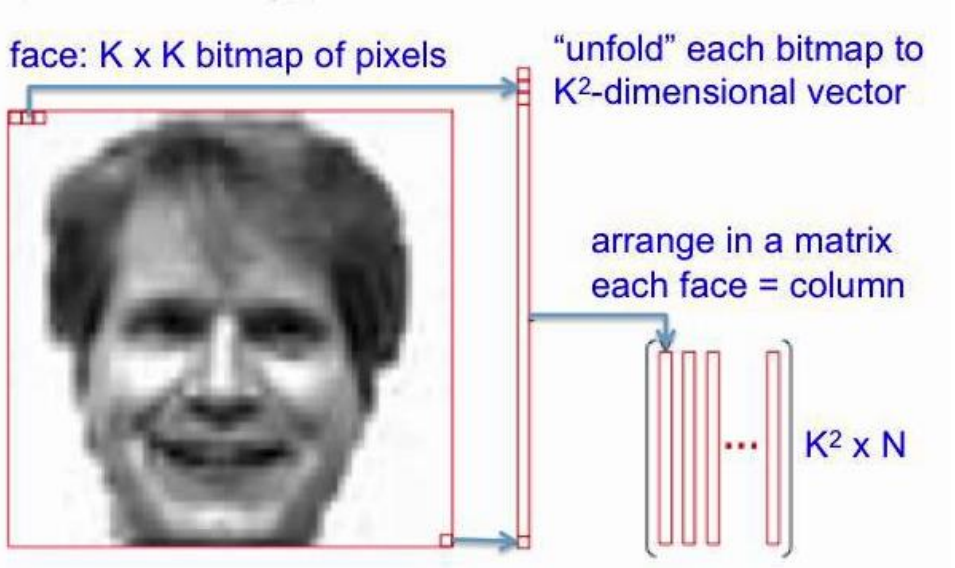
- Window or sub-image subimage is a collection of observation vectors
  - Maintain order
    - Vectorize windows
    - Kernel (matched-filter)
    - Markov models / graph models
  - Does not maintain order
    - Set-based models
    - Histogram based models



GEORGETOWN UNIVERSITY

Classifying entire images

- Classification done at image-level
  - Vectorize entire image: often not tractable
  - (Global) features computed across entire image
    - Image characterized by resulting feature vector.
- Examples
  - Labels are categories of imagery
    - Urban image
    - Rural image
    - Agricultural
    - …
  - Content-based image retrieval



GEORGETOWN UNIVERSITY

The Curse of Dimensionality

- To increase discrimination abilities
  - Many features are computed for each pixel
  - Many pixels are used to characterize some object
- Resulting vector may be huge!
- If the number of dimensions is high relative to the number of samples, models learned will be “erratic”.
  - Moreover: the data itself may only occupy a small subspace (lower dimensional subspace). Using a higher dimensional space may simply lead to over training or learning noise.
  - Hughes phenomenon: the predictive power of a classifier decreases as the number of dimensions increases.

Reducing dimension

- High dimensional features may present problems
  - hard to summarize
  - estimates of classifier coefficients may be noisy

- Solution: Reduce dimensionality
- Principal components analysis
  - project features onto dimensions that preserve variance
- Linear discriminant analysis
  - project features onto dimensions that are discriminative

Assume we have a set of n feature vectors $\boldsymbol{x}_i$ $(i = 1, \ldots, n)$ in $\mathbb{R}^d$. Write

$$\boldsymbol{\mu} = \frac{1}{n} \sum_i \boldsymbol{x}_i$$

$$\Sigma = \frac{1}{n-1} \sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$$

The unit eigenvectors of Σ—which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue—give a set of features with the following properties:

- They are independent.
- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the k-dimensional set of linear features that preserves the most variance.

Algorithm 16.1: Principal Components Analysis

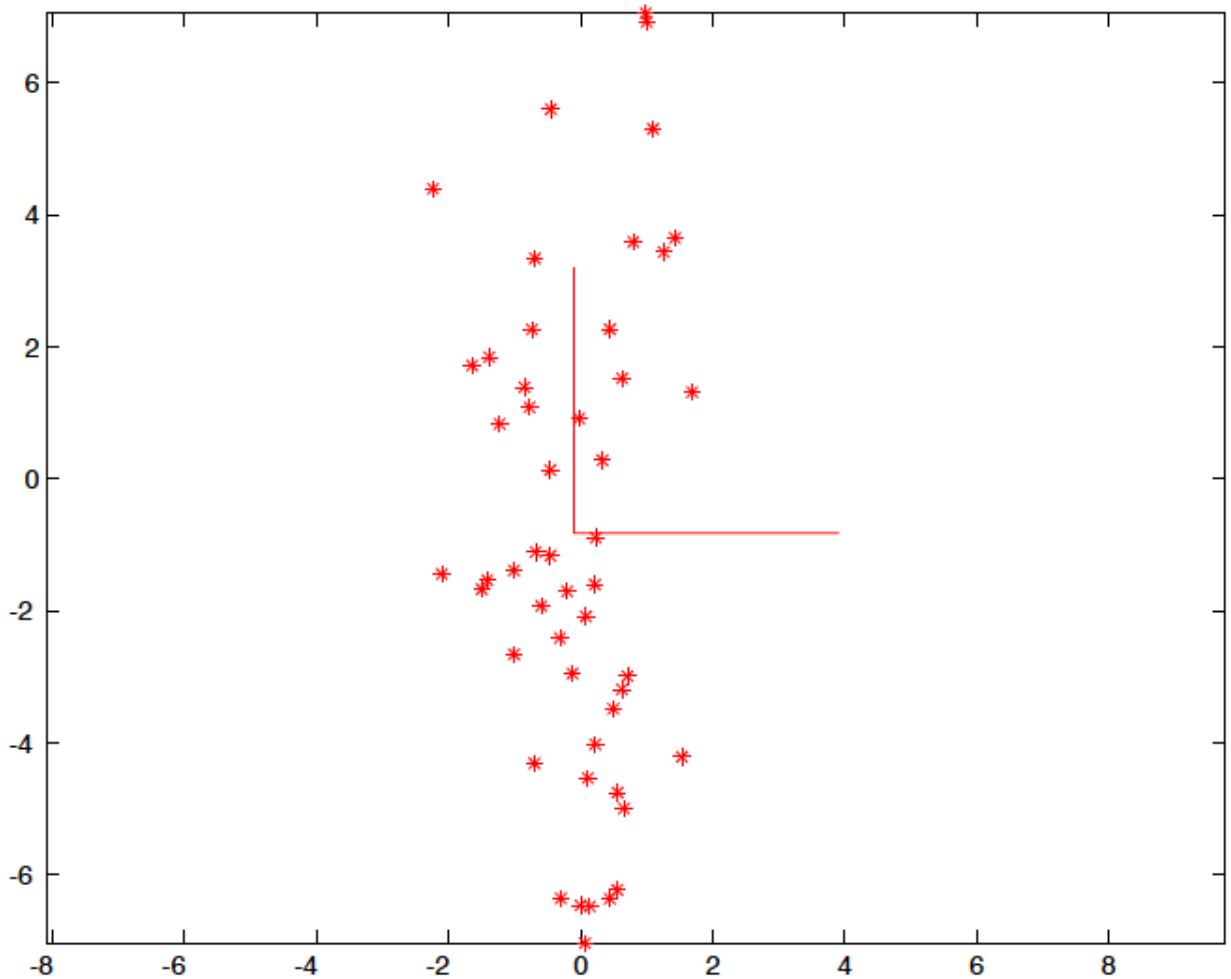


FIGURE 16.11: A dataset that is well represented by a principal component analysis. The axes represent the directions obtained using PCA; the vertical axis is the first principal component, and is the direction in which the variance is highest.

Difficulties with PCA

- Projection may suppress important detail
  - ***smallest variance directions may not be unimportant***

- Method does not take discriminative task into account
  - typically, we wish to compute features that allow good discrimination
  - not the same as largest variance

PCA Example

- Will projecting this 2-d data set onto the first eigenvector help?

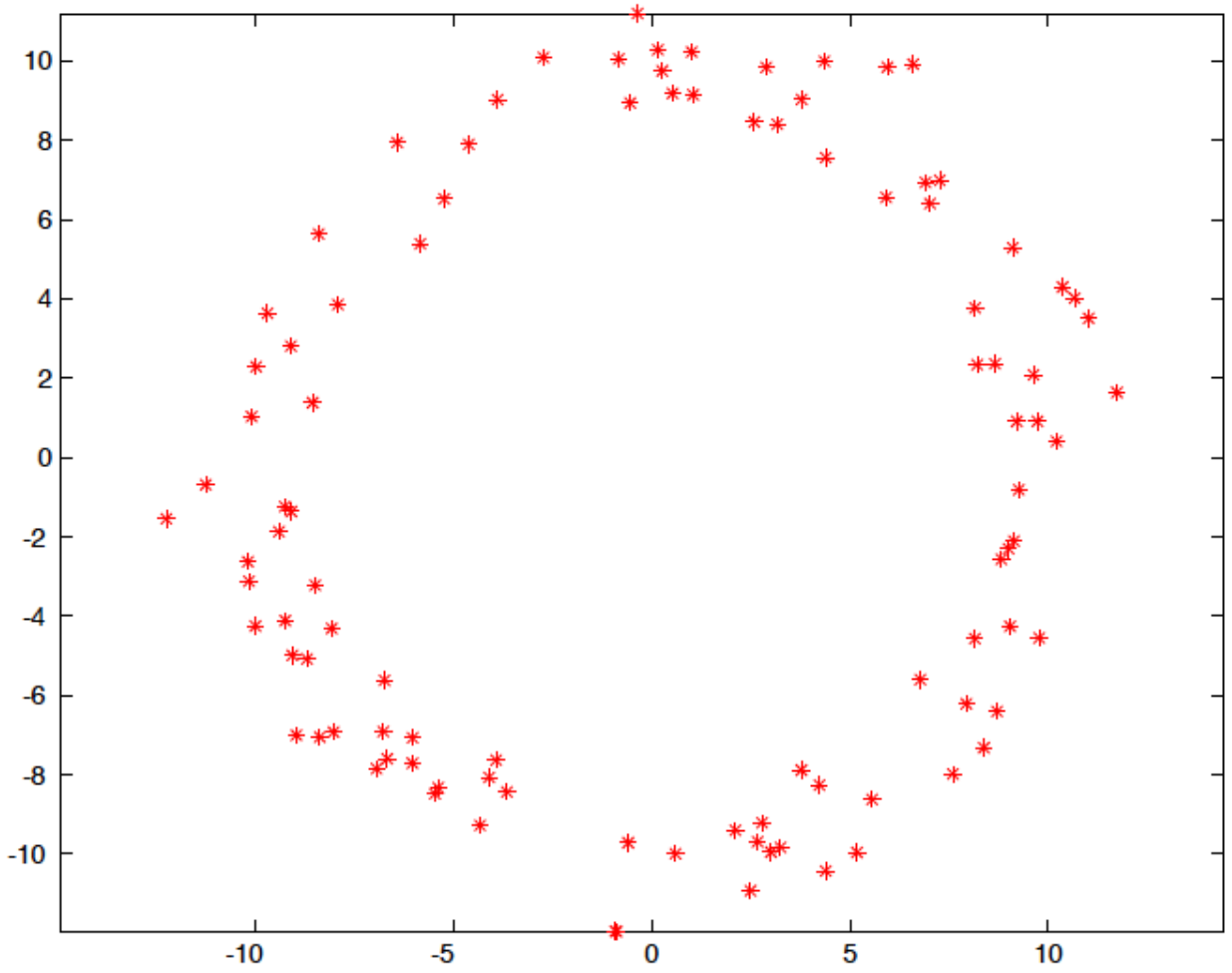
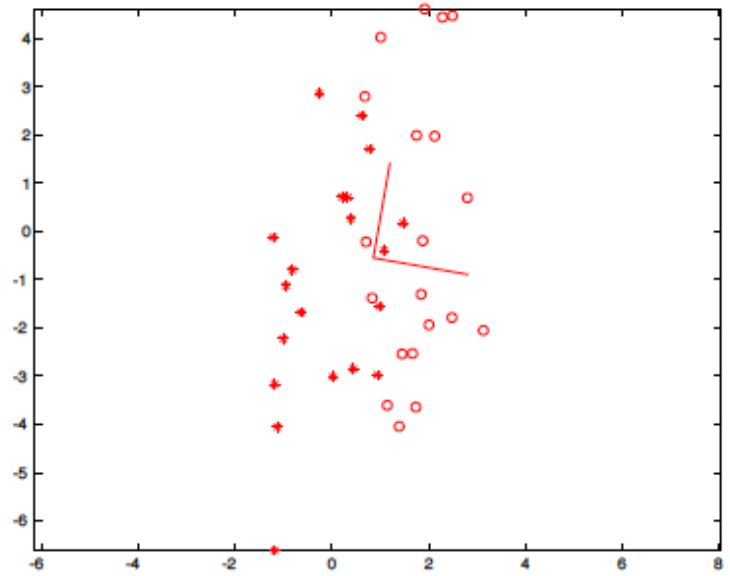


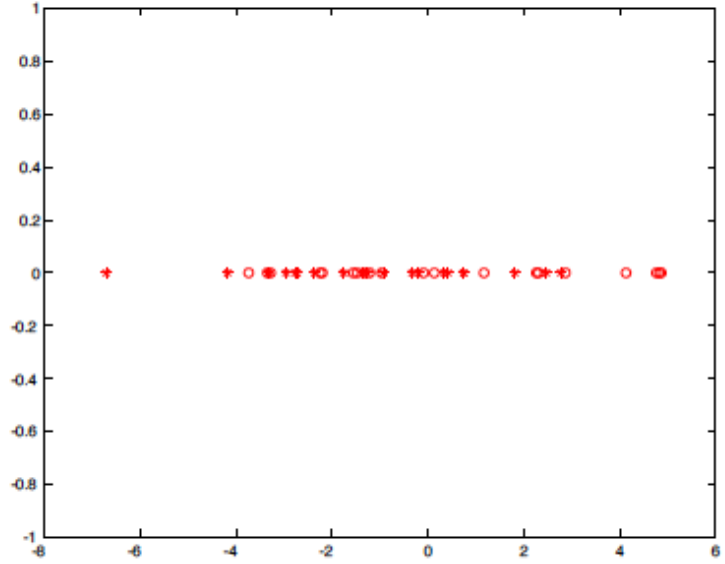
FIGURE 16.12: Not every dataset is well represented by PCA. The principal components of this dataset are relatively unstable, because the variance in each direction is the same for the source. This means that we may well report significantly different principal components for different datasets from this source. This is a secondary issue; the main difficulty is that projecting the dataset onto some axis suppresses the main feature, its circular structure.

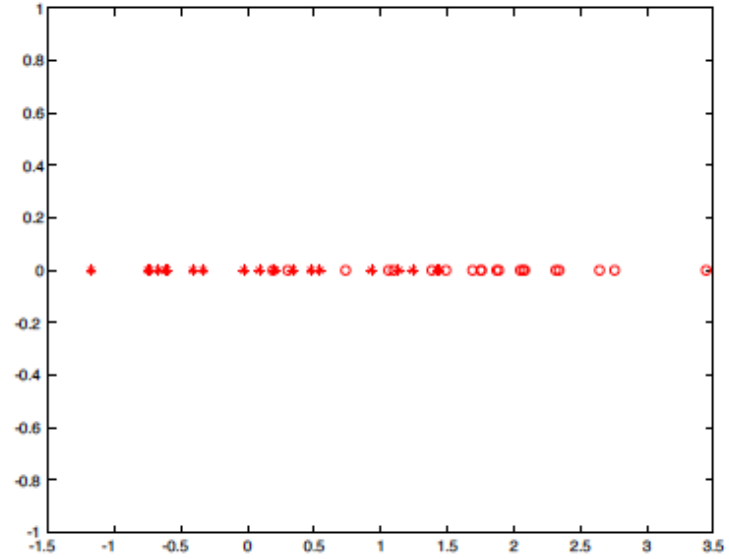
PCA Example

- Classes
  - Star
  - Circle

- The principal component analysis does not take into account the discriminative capabilities after the dimensionality reduction.







Linear Discriminant Analysis

- Otherwise known as canonical variates
- Project onto dimensions that preserve discrimination
  - by comparing between class variance to within class variance

Assume that we have a set of data items of g different classes. There are n_k items in each class, and a data item from the kth class is $\boldsymbol{x}_{k,i}$, for $i \in \{1, \ldots, n_k\}$. The jth class has mean $\boldsymbol{\mu}_j$. We assume that there are p features (i.e., that the $\boldsymbol{x}_i$ are p-dimensional vectors).
Write $\overline{\boldsymbol{\mu}}$ for the mean of the class means, that is,

$$\overline{\boldsymbol{\mu}} = \frac{1}{g} \sum_{j=1}^{g} \boldsymbol{\mu}_j,$$

Write

$$\mathcal{B} = \frac{1}{g-1} \sum_{j=1}^{g} (\boldsymbol{\mu}_j - \overline{\boldsymbol{\mu}})(\boldsymbol{\mu}_j - \overline{\boldsymbol{\mu}})^T.$$

Assume that each class has the same covariance Σ, which is either known or estimated as

$$\Sigma = \frac{1}{N-1} \sum_{c=1}^{g} \left\{ \sum_{i=1}^{n_c} (\boldsymbol{x}_{c,i} - \boldsymbol{\mu}_c)(\boldsymbol{x}_{c,i} - \boldsymbol{\mu}_c)^T \right\}.$$

The unit eigenvectors of $\Sigma^{-1}\mathcal{B}$, which we write as $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_d$, where the order is given by the size of the eigenvalue and $\boldsymbol{v}_1$ has the largest eigenvalue, give a set of features with the following property:

- Projection onto the basis $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ gives the k-dimensional set of linear features that best separates the class means.

Algorithm 16.2: Canonical Variates

LDA (Linear Discriminant Analysis)

- LDA will perform a projection that scales each dimension based on within and between class variance.

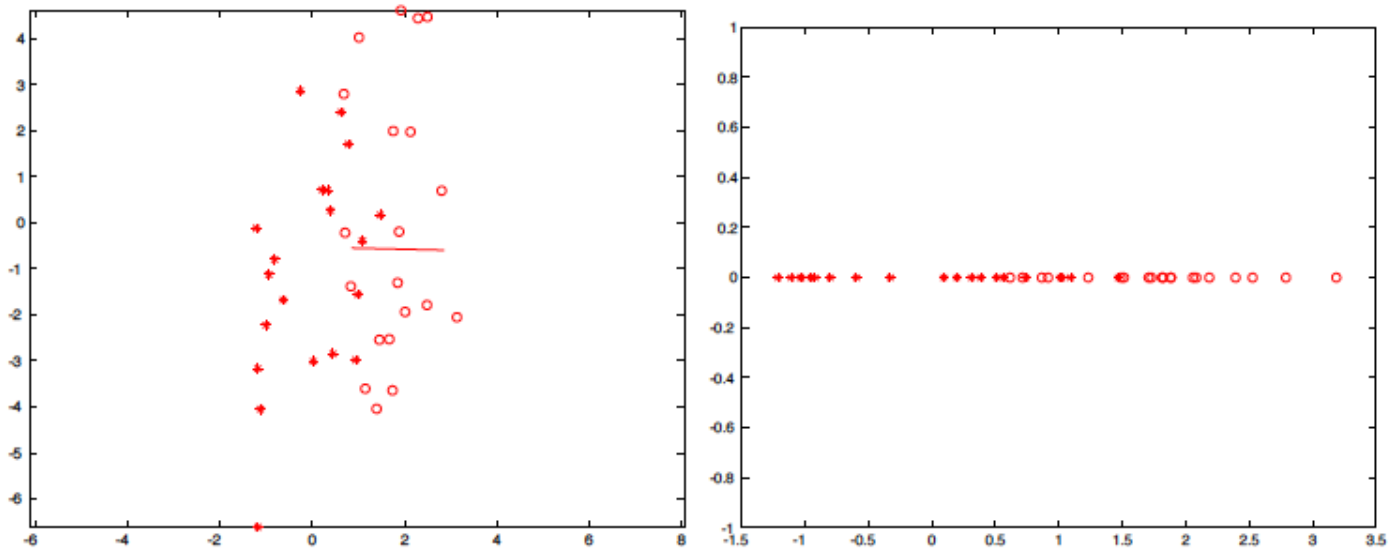


FIGURE 16.14: Canonical variates use the class of each data item as well as the features in estimating a good set of linear features. In particular, the approach constructs axes that separate different classes as well as possible. The dataset used in Figure 16.13 is shown on the **left**, with the axis given by the first canonical variate overlaid. On the **bottom right**, we show the projection onto that axis, where the classes are rather well separated.

Feature bagging

- Random Subspace Method
  - Randomly select a subset of features
  - Train multiple classifiers on different on random feature subsets

- Strategies
  - Ensemble Classification: Use all resulting classifiers as ensemble of weak learners.
  - Feature Selection: Identify which features yield good classifiers.

Generalizing for multiple scales

- During feature generation
  - Scale invariant features, eg, SIFT features
- Before classification
  - Create features using Scale Space Analysis
    - Gaussian / Laplacian Pyramids
- During classification
  - Have multiple classifiers trained to recognize pattern at different scales
    - Fuse results across multiple classifiers.

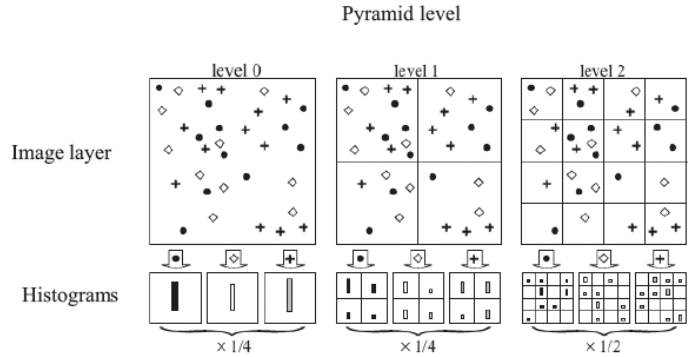


FIGURE 16.8: A simplified example of constructing a spatial pyramid kernel, with three levels. There are three feature types, too (circles, diamonds, and crosses). The image is subdivided into one-, four-, and sixteen-grid boxes. For each level, we compute a histogram of how many features occur in each box for each feature type. We then compare two images by constructing an approximate score of the matches from these histograms. *This figure was originally published as Figure 1 of "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," by S. Lazebnik, C. Schmid, and J. Ponce, Proc. IEEE CVPR 2006, © IEEE 2006.*

GEORGETOWN UNIVERSITY

Recognition and Classification problems

- What is it?
  - Object and scene recognition
- Who is it?
  - Identity recognition
- Where is it?
  - Object detection
- What is it doing?
  - Activities
- All of these are **classification** problems
  - Choose one class from a list of possible candidates

What is recognition?

- A different taxonomy from [Csurka *et al.* 2006]:
- Recognition
  - Where is *this* particular object?
- Categorization
  - What *kind* of object(s) is(are) present?
- Content-based image retrieval
  - Find me something that looks similar
- Detection
  - Locate *all* instances of a given class

Detection with a classifier

- Searching the image
  - all windows
  - at relevant scales
- Prepare Features
- Classify
- Issues
  - How to get only one response
    - Fusion
    - Confidence accumulation or aggregation
  - Speed
    - Images can be large; all classification schemes (and feature generation schemes) do not scale well.
  - Accuracy
    - Are results good, repeatable, generalizable?

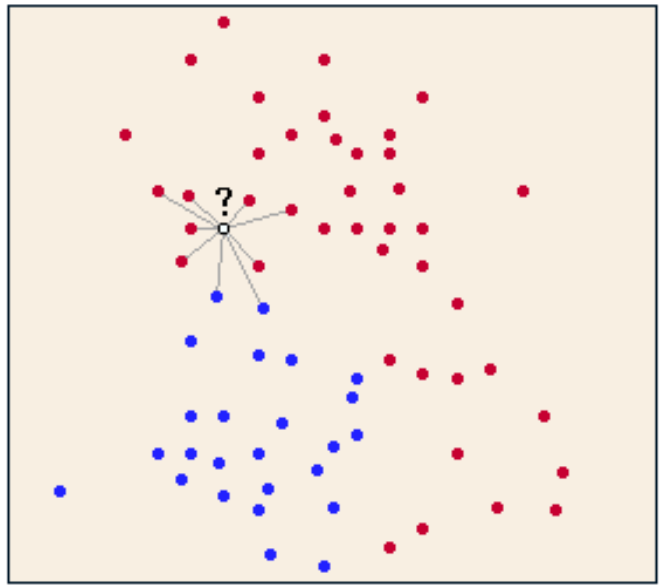
Classifiers

- Take a measurement x, predict a bit (yes/no; 1/-1; 1/0; etc)
  - Note: spatial information may be included with feature calculation

- Common Strategies:
  - non-parametric
    - nearest neighbor
  - probabilistic
    - histogram
    - logistic regression
  - decision boundary
    - SVM

Nearest neighbor classification

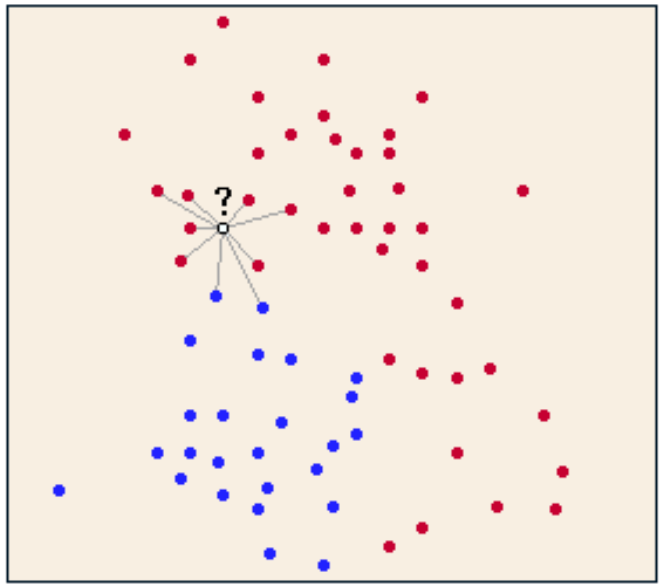
- Training: Construct a Set of Examples
  - (x_i, y_i)
    - here y is yes/no or -1/1 or 1/0 or....
  - training set
- Strategy
  - to label new example (test example)
    - find closest training example
    - report its label
- Advantage
  - Maintains essentially training information; no summarization
- Issue
  - how do we find closest example?
  - what distance should we use?



k-nearest neighbors

- Strategy
  - to classify test example
    - find k-nearest neighbors of test point
    - vote (it's a good idea to have k odd)

- Issues (again)
  - how do we find nearest neighbors?
  - what distance should we use?



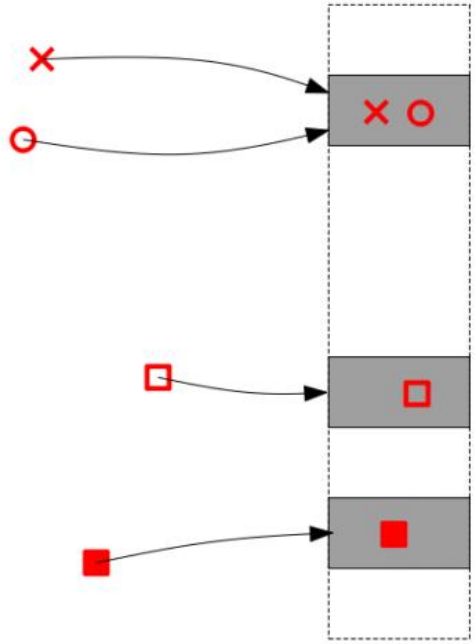
Nearest neighbors

- Exact nearest neighbor in large dataset
  - linear search is very good
  - very hard to do better (surprising fact)

- Approximate nearest neighbor is easier
  - methods typically give probabilistic guarantees
  - Time tradeoff: may have good “enough” accuracy
  - methods
    - locality sensitive hashing
    - k-d tree

Locality Sensitive Hashing

- Locality Hashing
  - Vectors “near” to each other are mapped to the same hash bucket (collide).
  - Thus finding a nearest neighbor, or k nearest neighbors
  - Issues:
    - Finding LSH is hard
    - Various Bucket Size issues:
      - EG Bucket(s) may be empty



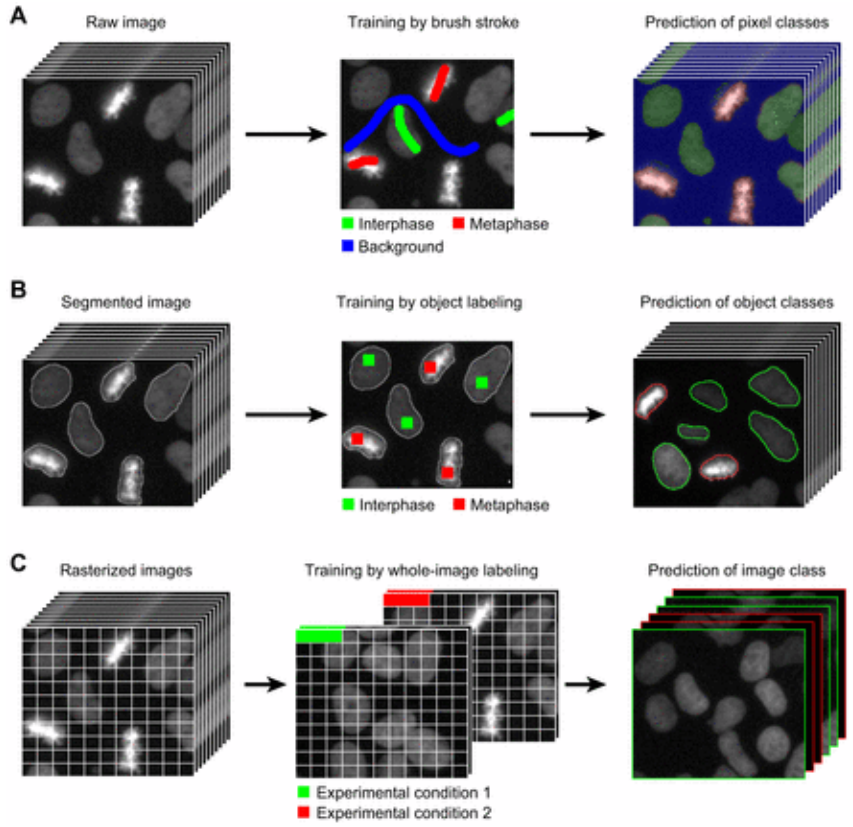
Common Classifiers

- Probabilistic Models
  - Naïve Bayes Classifier
  - Relevance Vector Machine (RVM)
  - Gaussian Process
  - Random Forest
  - Markov Models

- Other common models
  - SVM
  - Neural Network
  - Graph-Based Models

Machine Learning in Vision

- Learn model parameters given labeled examples
  - But HOW do we get the labels?
  - EG:
    - LabelMe app
    - Annotation Tools
- Ground Truth
  - Pixelwise
  - Pixel-set
  - Image-level

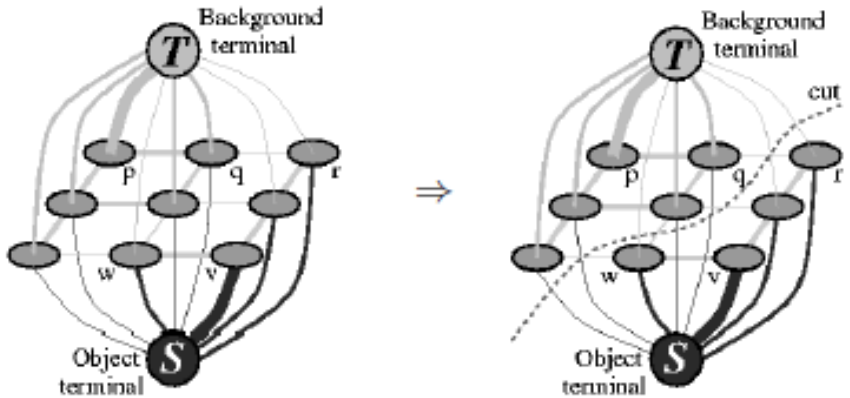


GEORGETOWN UNIVERSITY

Classification schemes with spatial information

- Features may include spatial information, but do classifiers?

- Markov Models / Graph-based models
  - Both observation and location of samples are considered
  - Often posed as energy minimization of multiple terms



(b) Graph. (c) Cut.

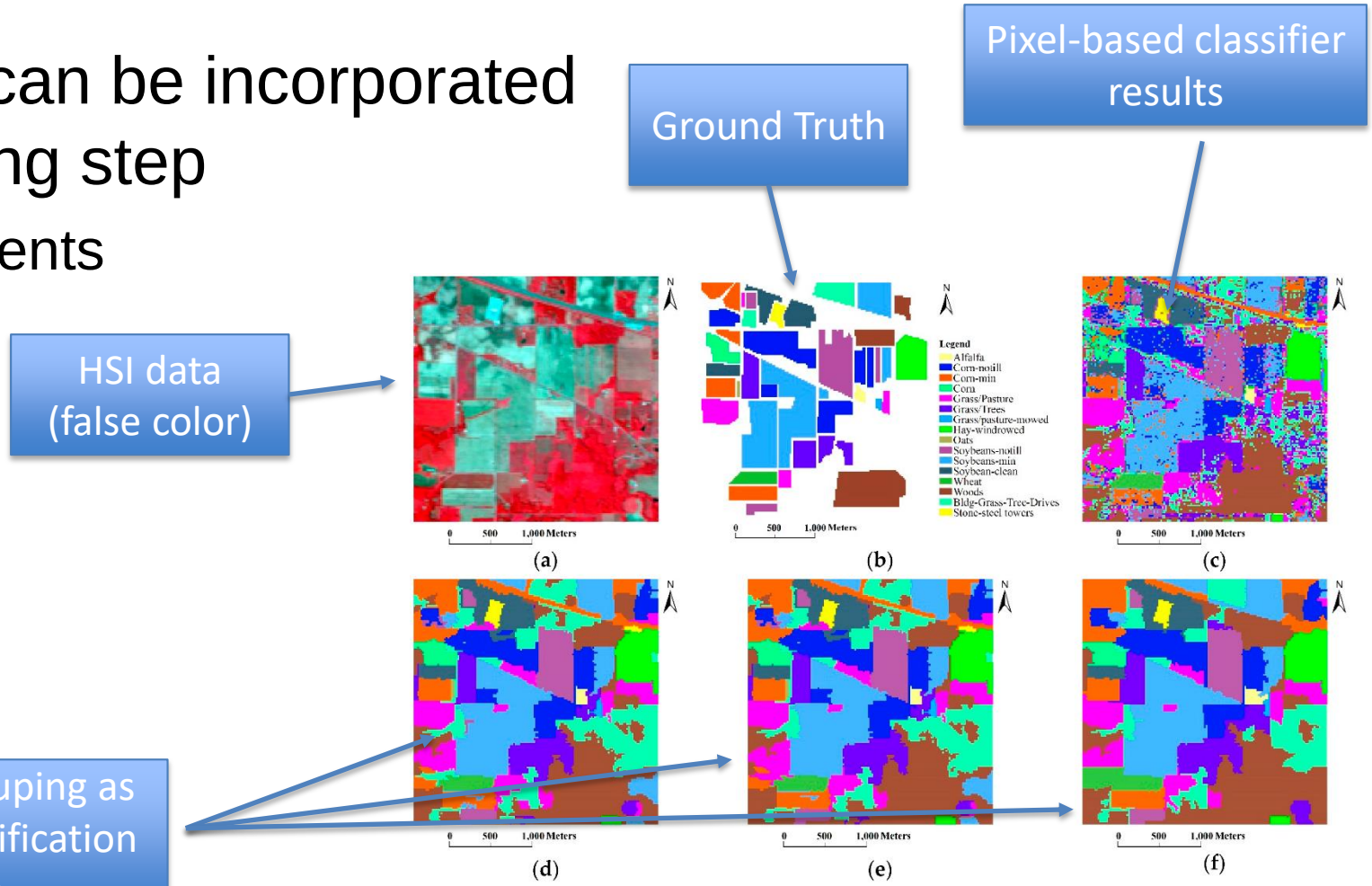
GEORGETOWN UNIVERSITY

Detection Maps

- Once detection or classification is completed for a pixel-based classifier, a label (with confidence) is associated with each pixel
- In detection, resulting map can be thresholded to make decision
  - Often called decision map.
  - Confidence is displayed for each pixel (as an image), thus a detection map
- If no spatial information was included in classification, this can be incorporated into a post processing step …

Post processing of Detection and Classification Maps: Decision Maps

- Spatial information can be incorporated into a post processing step
  - Connected Components
  - Morphology
- Example
  - Indian Pines



GEORGETOWN UNIVERSITY

Evaluating Results

- Classification Accuracy
  - Confusion Matrix
  - Receiver Operating Characteristic Curve
    - AUC
  - Precision vs recall

- Accuracy as a function of …
  - Number of training samples
  - Training time

Confusion Matrix

- Is the classifier confused?
  - Which classes does classifier have difficulty distinguishing between?

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Predicted Class

True Class	1	2	…	m
1	$C_{1,1}$	$C_{1,3}$	…	$C_{1,m}$
2	$C_{2,1}$	$C_{2,2}$	…	$C_{2,m}$
⋮	⋮	⋮	⋱	⋮
m	$C_{m,1}$	$C_{m,2}$	…	$C_{m,3}$

Sensitivity for class k
(total correct / row total)

$$S_k = \frac{C_{k,k}}{\sum_{j=1}^{m} C_{k,j}}$$

Predictive value for class k
(total correct / column total)

$$P_k = \frac{C_{k,k}}{\sum_{j=1}^{m} C_{j,k}}$$

Summary measures:

Accuracy:

$$A = \frac{1}{m}\sum_{k=1}^{m} S_k$$

Overall Predictive Value:

$$P = \frac{1}{m}\sum_{j=1}^{m} P_k$$

GEORGETOWN UNIVERSITY

Confusion Matrix

- Example (SVM-KNN) Confusion matrix (illustrated using 2-d color map)
  - Red: high probability
  - Blue: low probability

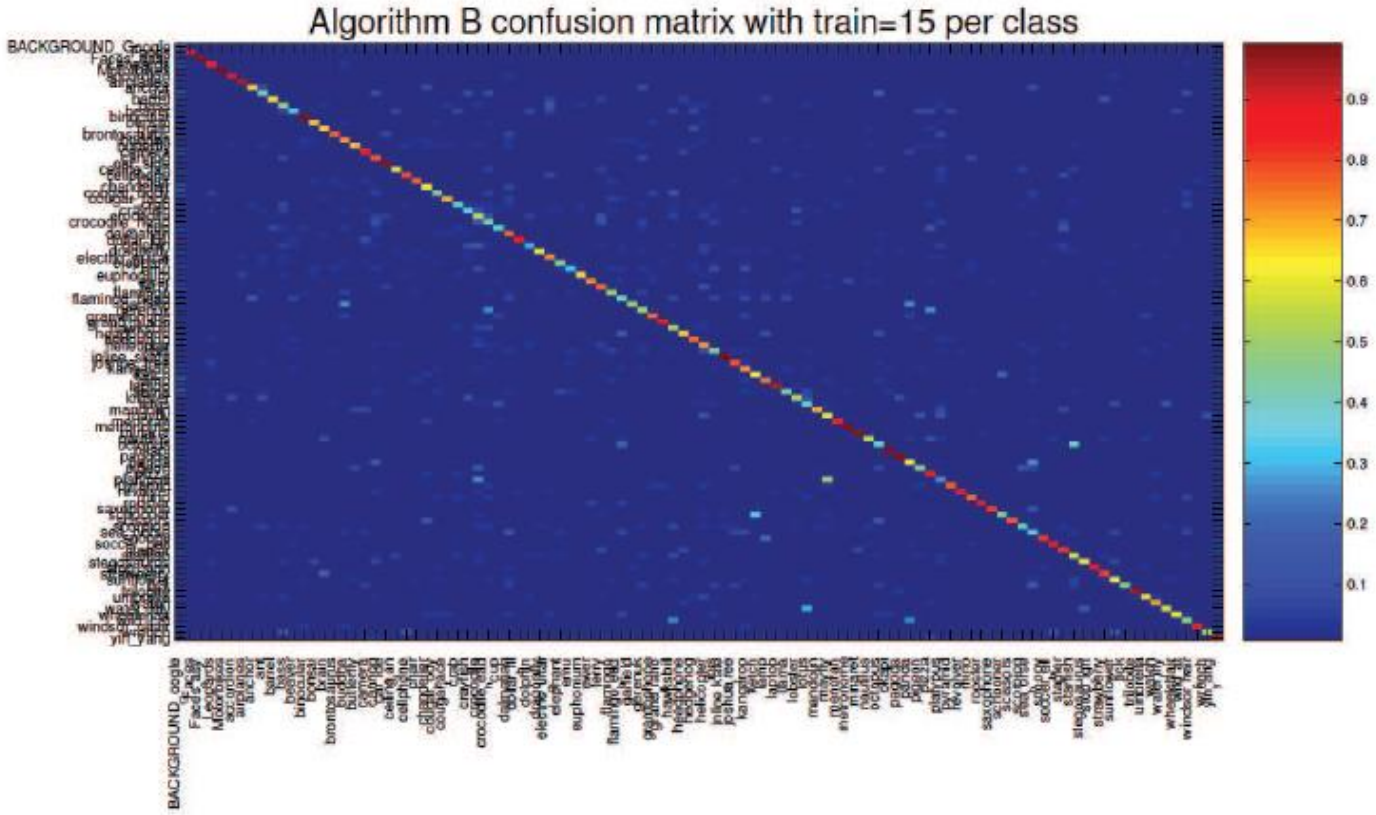


FIGURE 15.3: An example of a class confusion matrix from a recent image classification system, due to Zhang *et al.* (2006*a*). The vertical bar shows the mapping of color to number (warmer colors are larger numbers). Note the redness of the diagonal; this is good, because it means the diagonal values are large. There are spots of large off-diagonal values, and these are informative, too. For example, this system confuses: schooners and ketches (understandable); waterlily and lotus (again, understandable); and platypus and mayfly (which might suggest some feature engineering would be a good idea). *This figure was originally published as Figure 5 of "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," by H. Zhang, A. Berg, M. Maire, and J. Malik, Proc. IEEE CVPR, 2006,* © *IEEE, 2006.*

Material classification

- Example (using SIFT features) Confusion matrix (illustrated using 2-d color map)
  - Purple: high probability
  - Blue: low probability

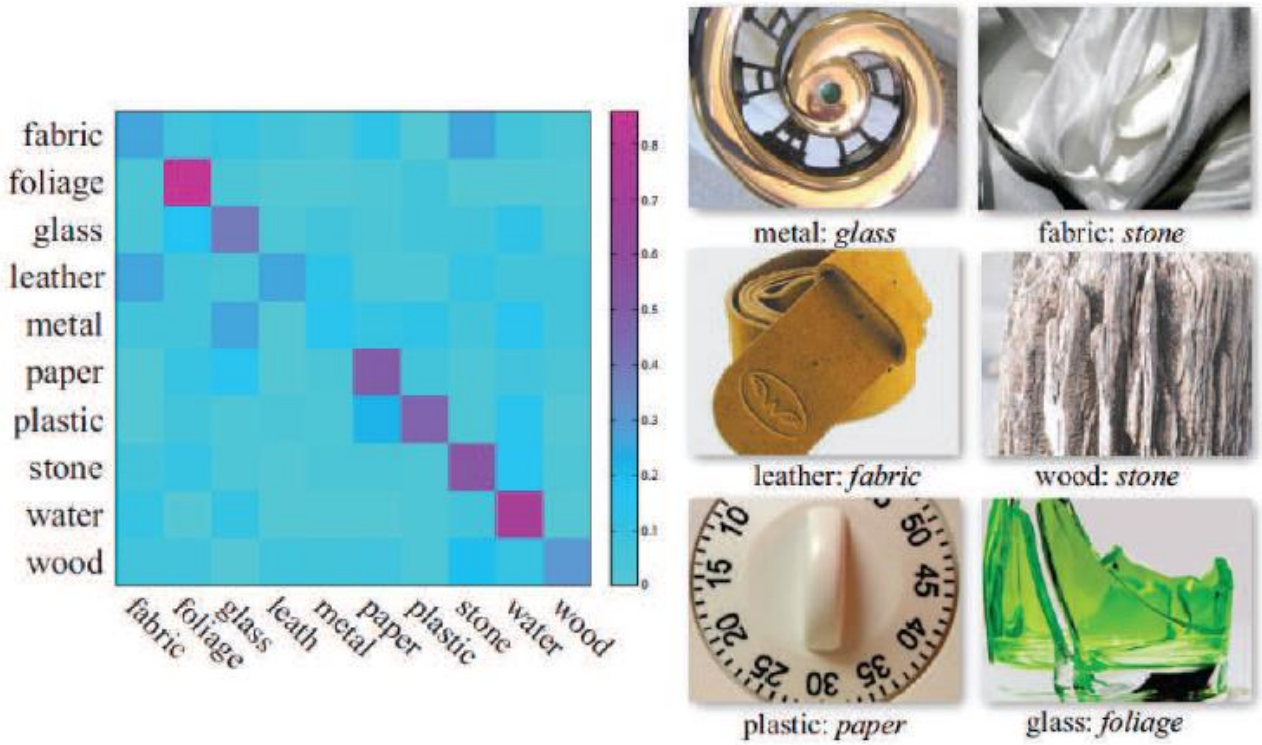
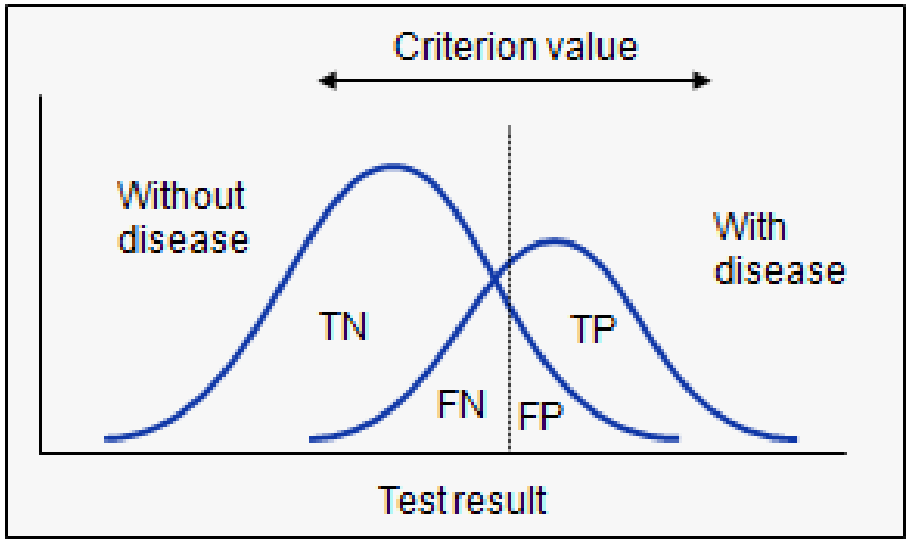
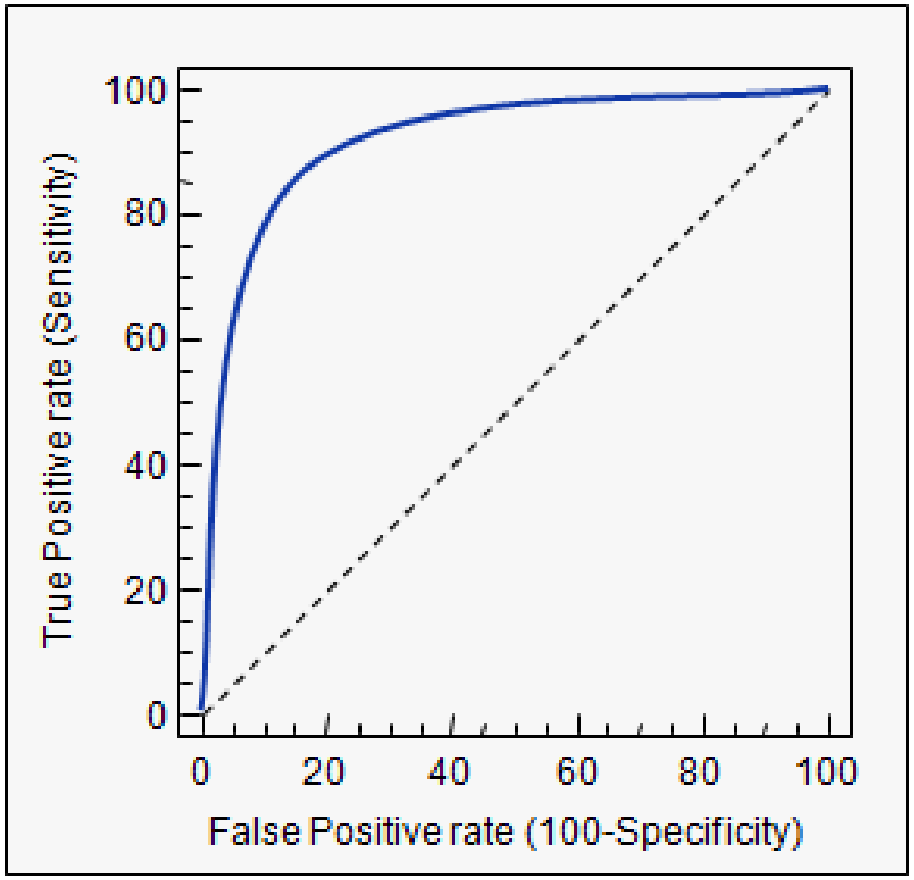


FIGURE 16.17: Liu *et al.* (2010) prepared a material classification dataset from flickr images, and used a combination of SIFT features and novel features to classify the materials. This is a difficult task, as the class confusion matrix on the **left** shows; for example, it is quite easy to mix up metal with most other materials, particularly glass. On the **right**, examples of misclassified images (the italic label is the incorrect prediction). *This figure was originally published as Figure 12 of "Exploring Features in a Bayesian Framework for Material Recognition," by C. Liu, L. Sharan, E. Adelson, and R. Rosenholtz Proc. CVPR 2010, 2010 © IEEE, 2010.*

Receiver Operating Characteristic Curve

- ROC
  - For each threshold, plot true positive vs false positive rates





Distributions represent truth; vertical line represents threshold

GEORGETOWN UNIVERSITY

Receiver operating curve

- Example ROC for skin detector

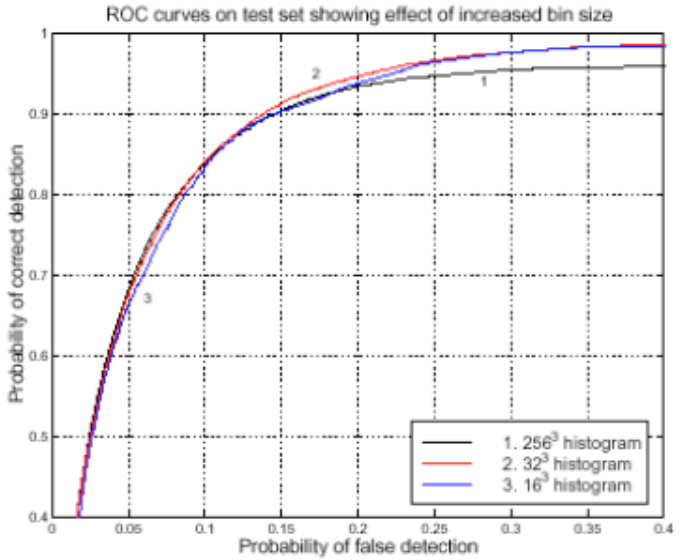
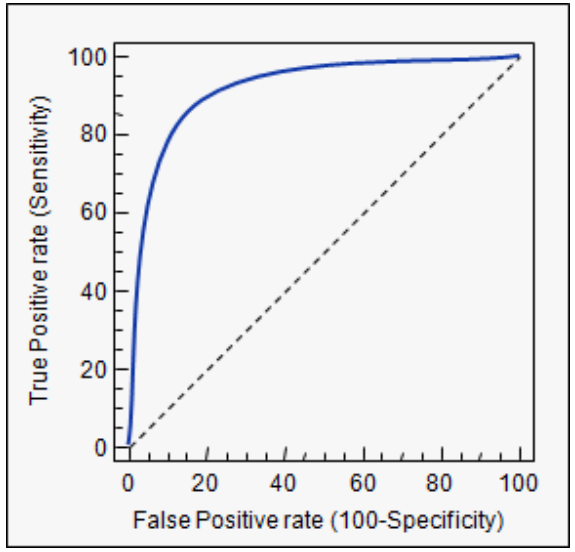


FIGURE 15.4: The receiver operating curve for a classifier, used to build a skin detector by Jones and Rehg. This curve plots the detection rate against the false-negative rate for a variety of values of the parameter θ. A perfect classifier has an ROC that, on these axes, is a horizontal line at 100% detection. There are three different versions of this classifier, depending on the detailed feature construction; each has a slightly different ROC. *This figure was originally published as Figure 7 of "Statistical color models with application to skin detection," by M.J. Jones and J. Rehg, Proc. IEEE CVPR, 1999* © *IEEE, 1999.*

GEORGETOWN UNIVERSITY

Summarizing a ROC curve

- AUC (Area under the curve)
  - Can be numerically computed similar to Riemann integral



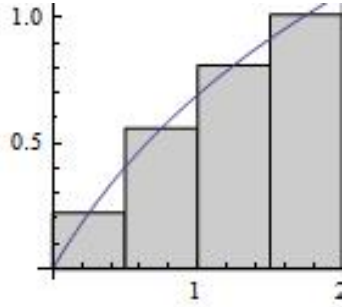


TABLE I
AUC FOR MIL ALGORITHMS USING MUSK 1

Algorithm	Musk 1	Algorithm	Musk 1
R-RSF-MIL-1	0.902	C-RSF-MIL-1	0.916
R-RSF-MIL-4	0.912	C-RSF-MIL-2	0.942
R-RSF-MIL-8	0.948	**C-RSF-MIL-3**	**0.949**
MIRVM	0.942	MIBoost	0.899
MILR	0.846	MISVM	0.899

GEORGETOWN UNIVERSITY

Accuracy on Caltech

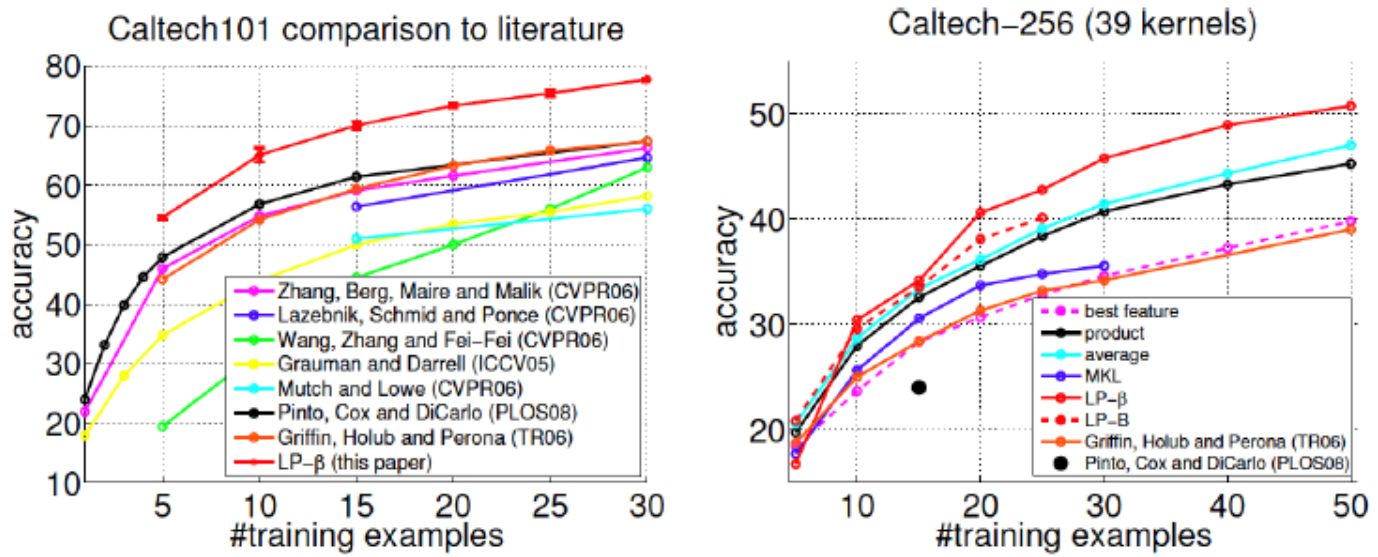


FIGURE 16.20: Graphs illustrating typical performance on Caltech 101 for single descriptor types (**left**) and on Caltech 256 for various types of descriptor (**right**; notice the vertical scale is different), plotted against the number of training examples. Although these figures are taken from a paper advocating nearest neighbor methods, they illustrate performance for a variety of methods. Notice that Caltech 101 results, while not perfect, are now quite strong; the cost of going to 256 categories is quite high. Methods compared are due to: Zhang *et al.* (2006*b*), Lazebnik *et al.* (2006), Wang *et al.* (2006), Grauman and Darrell (2005), Mutch and Lowe (2006), Griffin *et al.* (2007), and Pinto *et al.* (2008); the graph is from Gehler and Nowozin (2009), which describes multiple methods (anything without a named citation on the graph). *This figure was originally published as Figure 2 of "On Feature Combination for Multiclass Object Classification," by P. Gehler and S. Nowozin Proc. ICCV 2009, 2009* © *IEEE 2009.*

GEORGETOWN UNIVERSITY

(More) Evaluation

- Some problems require multiple decisions per input
  - eg Image Retrieval
  - Given an image input, find all images “similar” to it

- Precision
  - percentage of items in retrieved set that are relevant
- Recall
  - percentage of relevant items that are retrieved
- Precision vs recall
  - use classifier to label a collection of images
  - now plot precision against recall for different classifier thresholds

Precision vs recall

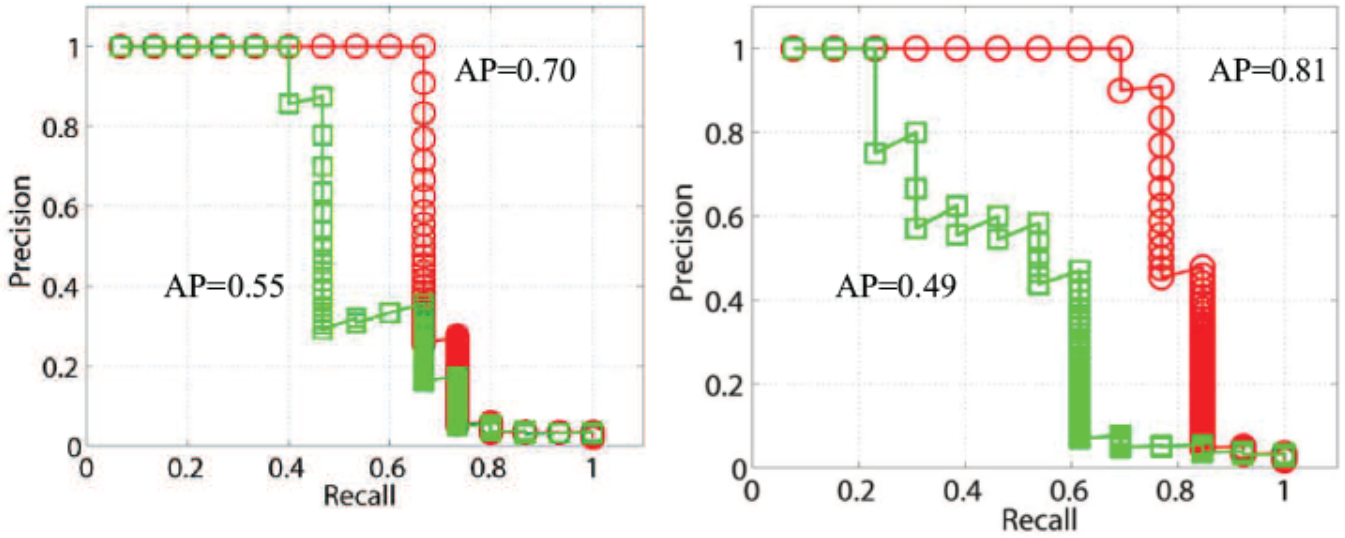


FIGURE 16.19: Plots of precision as a function of recall for six object queries. Notice how precision generally declines as recall goes up (the occasional jumps have to do with finding a small group of relevant images; such jumps would become arbitrarily narrow and disappear in the limit of an arbitrarily large dataset). Each query is made using the system sketched in Figure 16.5. Each graph shows a different query, for two different configurations of that system. On top of each graph, we have indicated the average precision for each of the configurations. Notice how the average precision is larger for systems where the precision is higher for each recall value. *This figure was originally published as Figure 9 of J. Sivic and A. Zisserman "Efficient Visual Search for Objects in Videos," Proc. IEEE, Vol. 96, No. 4, April 2008*

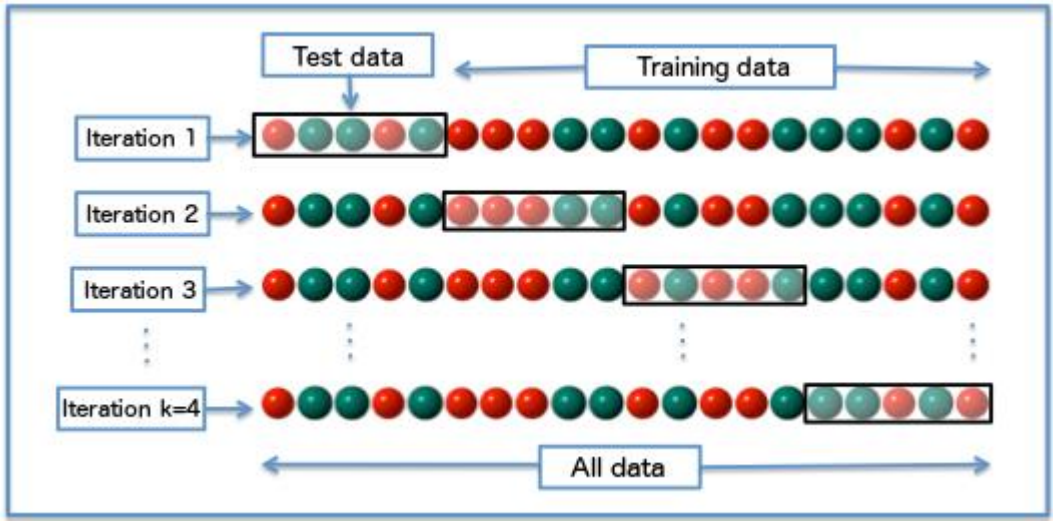
GEORGETOWN UNIVERSITY

Experimental Design

- How well does the classification scheme work?
  - We wary of “overtraining”
  - Do not assess a scheme based on results gather from “test on train”
  - Does it generalize?

- Do not Test on Train: Divide your set into training and testing sets (iteratively)

- Crossvalidation
  - k-fold
  - Leave one out



GEORGETOWN UNIVERSITY

Repeatability of Experimental Results

- Thresholded classification of data sets
  ~ binomial random variable

- Assume *each point* on the ROC curve is the result of *n* Bernoulli trials (binomial random variable)
  - The number of successful target identifications is given for each threshold
  - The MLE of the probability of success, *p*, of the binomial is then precisely the PD as shown on the ROC

Reliability of PD estimates:

Upper and Lower bound estimates

- Using the CDF of the binomial we can find upper and lower bound estimates of the parameter, *p*, given our observed data with x successes from n trials with a confidence interval 1 - α
  - That is, we perform two (distinct), one-tailed tests (Clopper–Pearson method)
  - Given the two equations below, we solve for p_{lower} and p_{upper} using the F distribution approximation (in Matlab)

$$P_{p_{lower}}(X \leq x) = \frac{\alpha}{2} = \sum_{i=0}^{x} \binom{n}{i} p_{lower}^{i} (1 - p_{lower})^{n-i}$$

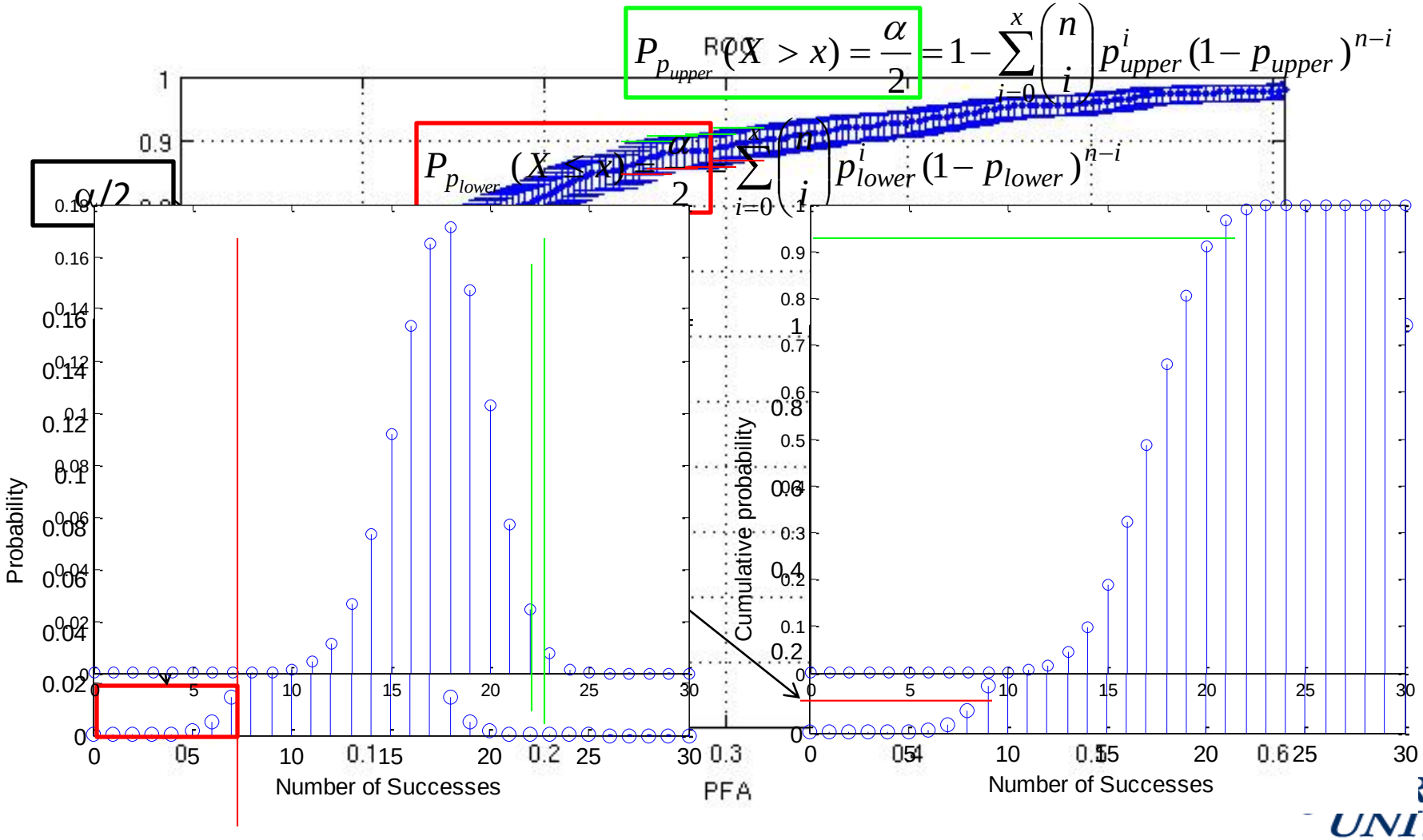
$$P_{p_{upper}}(X > x) = \frac{\alpha}{2} = 1 - \sum_{i=0}^{x} \binom{n}{i} p_{upper}^{i} (1 - p_{upper})^{n-i}$$

GEORGETOWN UNIVERSITY

Confidence Interval Visual

$$P_{p_{upper}}(X > x) = \frac{\alpha}{2} = 1 - \sum_{i=0}^{x} \binom{n}{i} p_{upper}^{i} (1 - p_{upper})^{n-i}$$

$$P_{p_{lower}}(X \le x) = \frac{\alpha}{2} = \sum_{i=0}^{x} \binom{n}{i} p_{lower}^{i} (1 - p_{lower})^{n-i}$$



GEORGETOWN UNIVERSITY

Appendix

Chapter 9

Classification Models

GEORGETOWN UNIVERSITY