HIERARCHICAL OBJECT CATEGORIZATION

Gregory Griffin and Pietro Perona. *Learning and Using Taxonomies For Fast Visual Categorization*. CVPR 2008

Marcin Marszalek and Cordelia Schmid. *Constructing Category Hierarchies for Visual Recognition*. ECCV 2008

Chloé Kiddon

Object Classification Problem

- Humans can visually recognize 10^4 – 10^5 different object categories
- **How can we get a machine to be able to do the same thing?**

Object Classification Problem

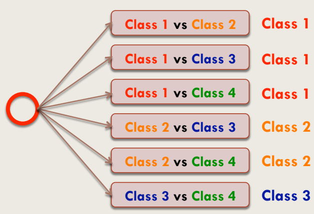
- Better image representations
  - Global visual histograms, Bag of features, Spatial Pyramid Matching, GIST
- Better classification methods
  - Maximum likelihood, k-Nearest Neighbor, linear models, SVMs, trees
- Scalable Techniques
  - **Hierarchical models**

Multi-class classification problem

- For each data instance, must choose between a large group of class labels
- Usually no single mathematical function exists to correctly separate data into multiple categories at once
  - We do have binary classifiers that can make decisions between two classes
- Use a set of binary classifiers!

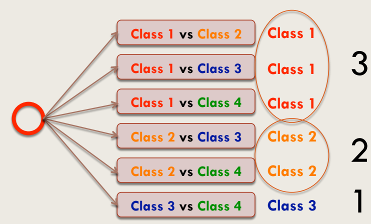
Binary Classifiers to the Rescue!

- Voting – One Vs. One



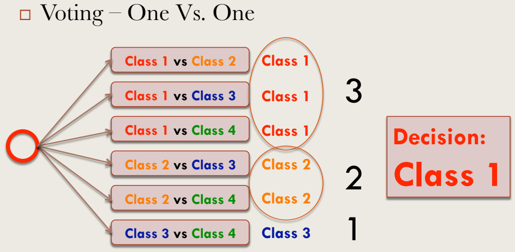
Binary Classifiers to the Rescue!

- Voting – One Vs. One



Binary Classifiers to the Rescue!

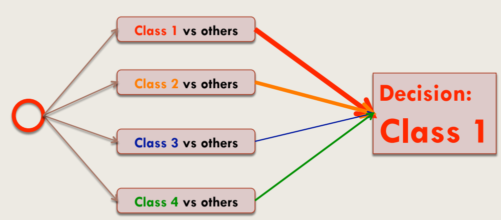
- Voting – One Vs. One



- k(k-1)/2 classifiers: O(k^2) complexity ☹☹

Binary Classifiers to the Rescue!

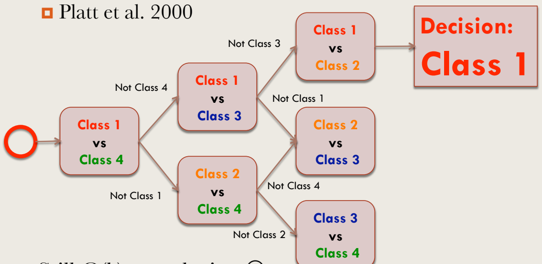
- Competition – One Vs. The Rest



- k classifiers: O(k) complexity ☹

Binary Classifiers to the Rescue!

- Discarding subsequent hypotheses – DAG-SVM
  - Platt et al. 2000



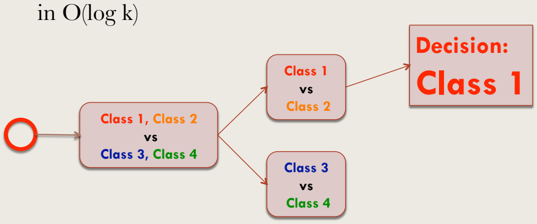
- Still O(k) complexity ☹

Past Approaches

- Classification time scales at ***best*** linearly with # of categories
- Need to do better if we have hundreds of thousands of categories!

Motivation for Hierarchical Structures

- Need classification costs that are sub-linear to the number of categories
- Searching a balanced hierarchy/tree structure runs in O(log k)



Open Research Question

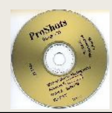
- What is the best way to build hierarchies for object category classification?
  - Top-down vs. bottom-up
  - How to choose splits at each node
- Case studies: two recent approaches
  - Griffin and Perona 2008 – tree hierarchy
  - Marszalek and Schmid 2008 – relaxed hierarchy

Griffin and Perona - 2008

Learning and Using Taxonomies For Fast Visual Categorization

Motivation

- Hierarchies are useful for object classification
- Manually-created hierarchies will not scale well
  - **Need to be able to automatically generate useful hierarchies**
- Hierarchies built on existing lexical hierarchies (such as WordNet) may not be optimal for visual classification



Motivation

- Hierarchies are useful for object classification
- Manually-created hierarchies will not scale well
  - **Need to be able to automatically generate useful hierarchies**
- Hierarchies built on existing lexical hierarchies (such as WordNet) may not be optimal for visual classification
  - **Need to find a more appropriate way to build hierarchies for this task**

Building Taxonomies

- Construct confusion matrix from training data
  - Train multi-class SVM with Spatial Pyramid Matching kernel
  - One vs. all classification scheme
- Tried two ways of building a taxonomy from confusion matrix
  - Top-down
  - Bottom-up

Building Taxonomies

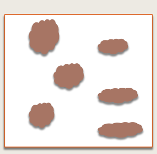
- Construct confusion matrix from training data
  - Train multi-class SVM with Spatial Pyramid Matching kernel
  - One vs. all classification scheme
- Tried two ways of building a taxonomy from confusion matrix
  - Top-down
  - Bottom-up

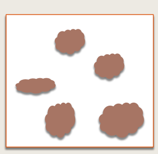
Earth Mover's Distance

- Measure of distance between two distributions over a region
- Minimal cost to convert one distribution into the other

Earth Mover's Distance

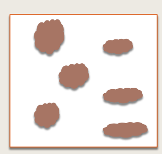
- Measure of distance between two distributions over a region
- Minimal cost to convert one distribution into the other

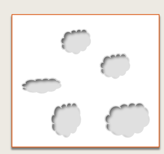




Earth Mover's Distance

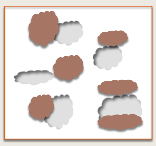
- Measure of distance between two distributions over a region
- Minimal cost to convert one distribution into the other





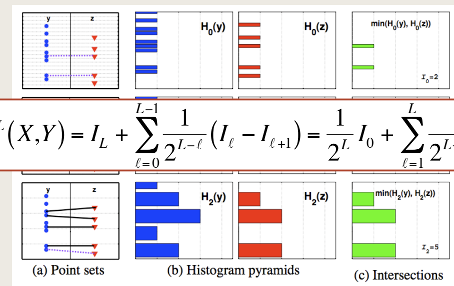
Earth Mover's Distance

- Measure of distance between two distributions over a region
- Minimal cost to convert one distribution into the other



Spatial Pyramid Matching – Grauman and Darrell 2006

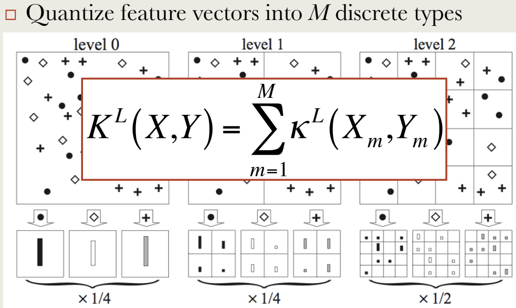
- Similarity measured by feature histogram intersections



$$\kappa^L(X,Y) = I_L + \sum_{\ell=0}^{L-1} \frac{1}{2^{L-\ell}}\left(I_\ell - I_{\ell+1}\right) = \frac{1}{2^L} I_0 + \sum_{\ell=1}^{L} \frac{1}{2^{L-\ell+1}} I_\ell$$

Spatial Pyramid Matching for Images – Lazebnik et al. 2006

- Quantize feature vectors into *M* discrete types



$$K^L(X,Y) = \sum_{m=1}^{M} \kappa^L(X_m, Y_m)$$

Spatial Pyramid Matching - Image Features

- SIFT features extracted from de-saturated image
  - Over 72x72 uniform grid
- M-word vocabulary chosen (M=200)
  - Fit random features to a Gaussian mixture model
- Map features to vocabulary words
- Reduce spatial grid to 4x4 for histogramming
- Train SVM based on spatial pyramid matching kernel

Spatial Pyramid Matching - Image Features

- SIFT features extracted from de-saturated image
  - Over 72x72 uniform grid
- M-word vocabulary chosen (M=200)
  - Fit random features to a Gaussian mixture model
- Map features to vocabulary words
- Reduce spatial grid to 4x4 for histogramming
- Train SVM based on spatial pyramid matching kernel

- Any image representation could have been used.

Building the Taxonomy – Top Down

- Recursively split the confusion matrix into two parts based on the Self-Tuning Spectral Clustering algorithm (Zelnik-Manor and Perona 2004)

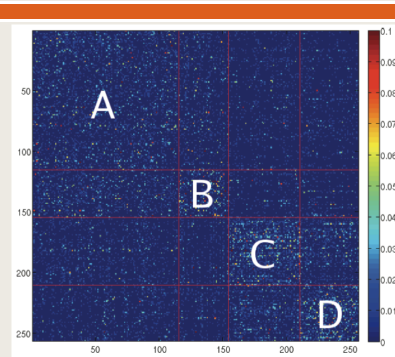
- Repeat process until leaves each have one category

Splitting the Confusion Matrix

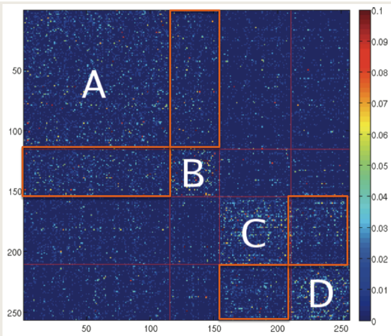
- Spectral Clustering uses an affinity matrix to cluster points
  - Affinity matrix $A \in R^{n \times n}$ defined by $A_{ij} = \exp\left(\frac{-d^2(s_i, s_j)}{\sigma^2}\right)$ where $i \neq j$, zeros along diagonal
  - σ is a static scaling parameter

- Self-tuning Spectral Clustering uses local scaling parameters
  - $\sigma_i = d(s_i, s_K)$

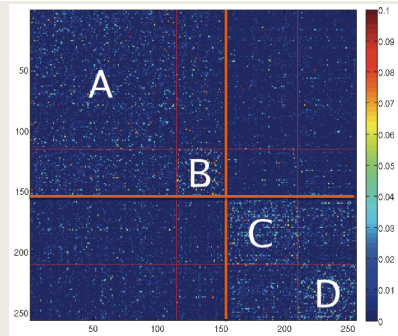
Splitting the Confusion Matrix



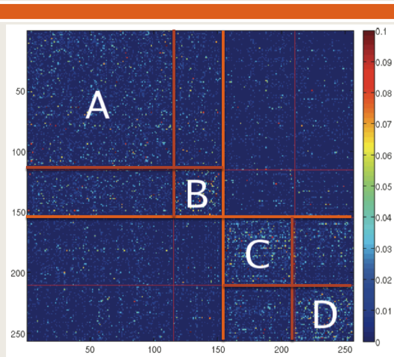
Splitting the Confusion Matrix



Splitting the Confusion Matrix



Splitting the Confusion Matrix



Building the Taxonomy – Bottom Up

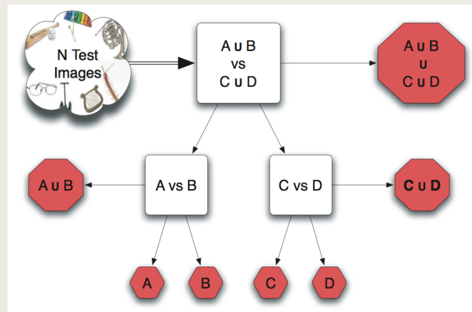
1. Start with each category in its own group
2. Pick two groups with the most mutual confusion and combine into a larger group
3. Continue until there is one group that contains all the categories

(Agglomerate clustering)

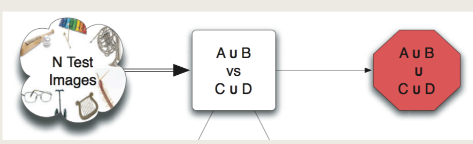
Training the Branch Classifiers

- Each branch node is trained on the training images for the classes used in the decision
- Picked a random subset for SVM training
  - F_{train} = 10%
- Split sampled training examples into two "classes"
- Trained an SVM for those two "classes"
  - Spatial Pyramid Matching kernel again

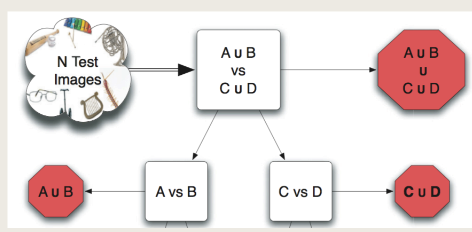
Classification



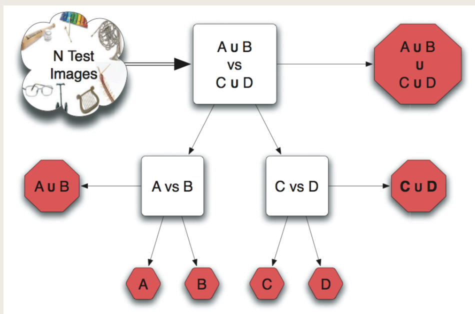
Classification



Classification

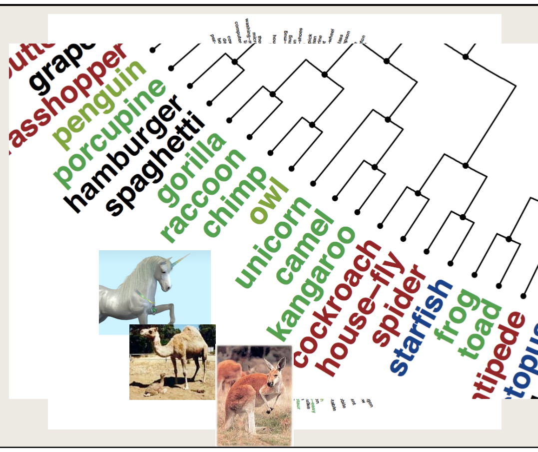


Classification

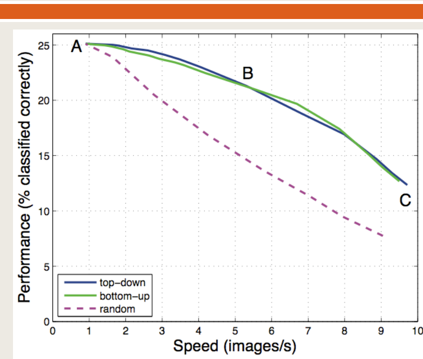


Experiments

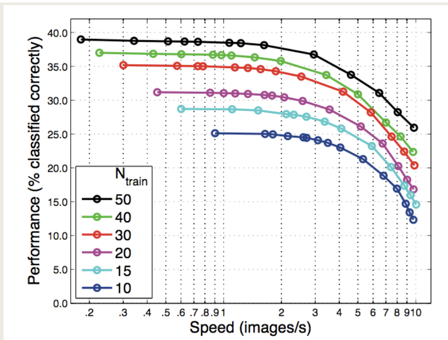
- Caltech-256 dataset – training and testing
  - Removed clutter category (background images)
  - 256 object categories
  - Tested performance for N_{train} = 10 to 50
- Tested different hierarchy approaches
  - Top down vs. bottom up vs. random control
- Tested performance vs. speed for classification
  - For different values of N_{train}



Experiments



Experiments

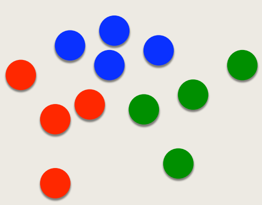


Marszalek and Schmid 2008

Constructing Category Hierarchies for Visual Recognition

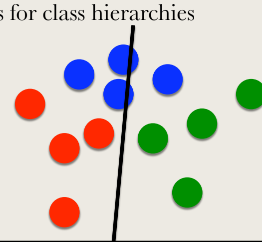
Problem with Previous Approaches

- As # of classes grows, finding partitions in the feature space becomes more difficult
- Separation problems within the hard constraint of tree models for class hierarchies



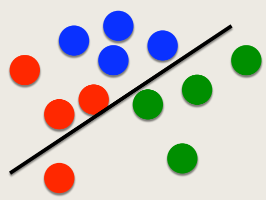
Problem with Previous Approaches

- As # of classes grows, finding partitions in the feature space becomes more difficult
- Separation problems within the hard constraint of tree models for class hierarchies



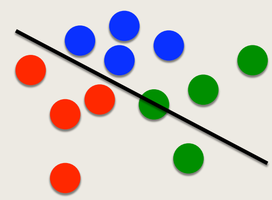
Problem with Previous Approaches

- As # of classes grows, finding partitions in the feature space becomes more difficult
- Separation problems within the hard constraint of tree models for class hierarchies



Problem with Previous Approaches

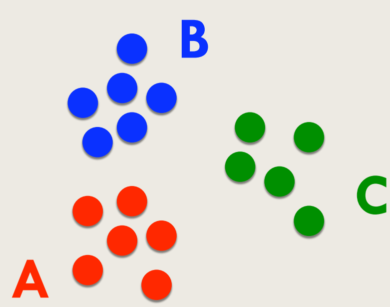
- As # of classes grows, finding partitions in the feature space becomes more difficult
- Separation problems within the hard constraint of tree models for class hierarchies



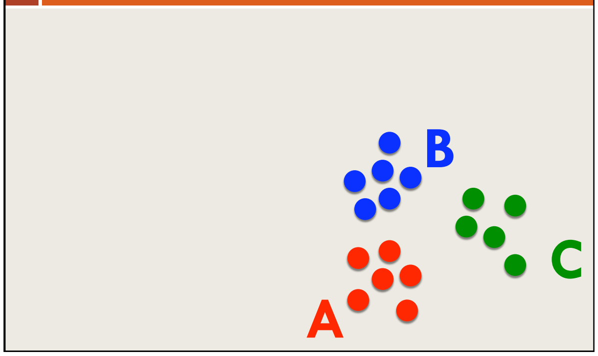
The Relaxed Hierarchy (RH)

- **Solution:** Avoid disjoint partitioning by postponing difficult classification decisions
- Do not force classes that lie on a partition boundary into either partition
  - Include in both
- Some slow down, still better than O(k) models

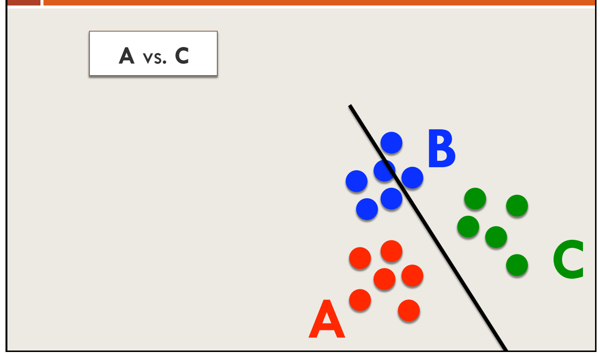
Example



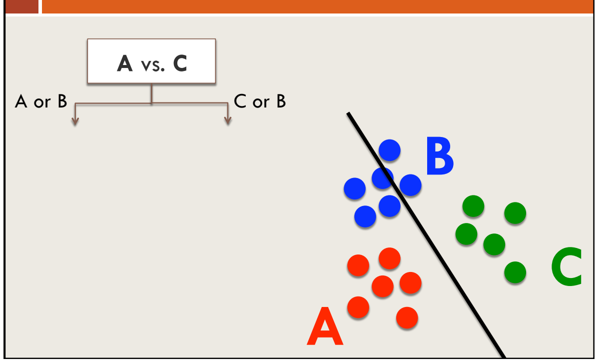
Example



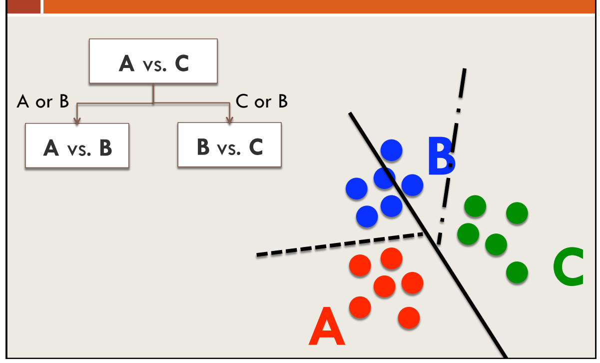
Example



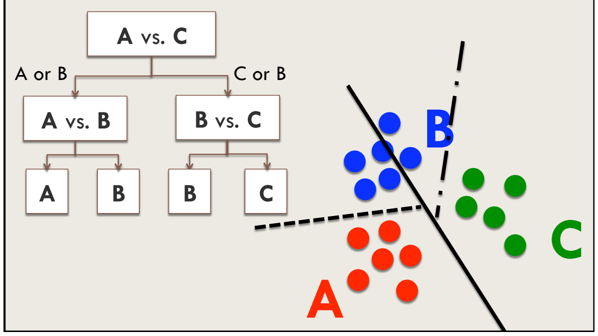
Example



Example

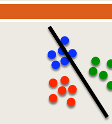


Example

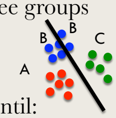


Building the Hierarchy

- Split the training examples S = A ∪ B
  - Normalized Cuts



- Use that split to split the classes into three groups
  - A-side classes, B-side classes, split classes



- Continue to split for each branch step until:
  - Only one class remains in the group
  - Classes impossible to split
    - Use One vs. The Rest

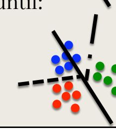


Image Representation

- Bag of features representation over SIFT features
- Regions for SIFT features chosen using interest point detectors
  - Harris-Laplace and Laplacian
  - Invariant to scale transformations
- Use k-means to cluster features and construct visual vocabulary (V = 8000)
- Again, methods can be used with any image representation

Training branch classifiers

- Train an SVM for each node using the training examples from the classes in the node decision
- Instances from classes split at the decision boundary are ignored

Experiments - Data

- Caltech-256 dataset – training and testing
- Use the first 250 categories
- $N_{train} = 15$
- Rest of images in each category in study are used for testing

Hierarchy Results

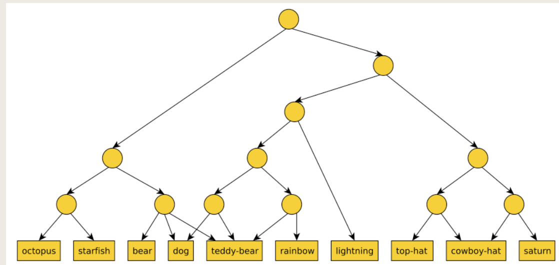


Fig. 5: Class hierarchy constructed by our method for the Caltech-256 dataset, displayed for a subset of 10 categories.

Results – Classification Accuracy

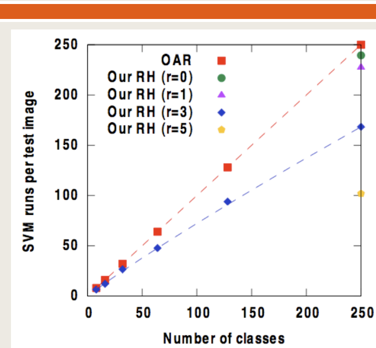
Using image representation just described:

OAR (reimpl. of Zhang et al.)	23.6%
Relaxed Hierarchy (r = 0, sparse IPs)	23.4%

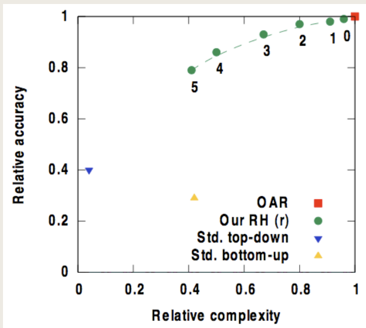
Using Spatial Pyramid Matching:

Griffin and Perona 2008 (Ntrain = 15)	Roughly 14.5% - 28%
Relaxed Hierarchy (r = 0, dense/grid))	27.9%

Results - Complexity



Results – Speed vs. Accuracy



Summary

- Class hierarchies can be used to perform classification in sub-linear time
- Hard class splits in branch classifiers can decrease accuracy
- Splits can be relaxed at a computational cost
- Neither paper shows future work, so apparently the problem is solved! ☺

The End!