

Machine Learning Crash Course



Computer Vision
James Hays

Slides: Isabelle Guyon,
Erik Sudderth,
Mark Johnson,
Derek Hoiem

Machine Learning Problems

Supervised Learning

Unsupervised Learning

Discrete
Continuous

classification or
categorization

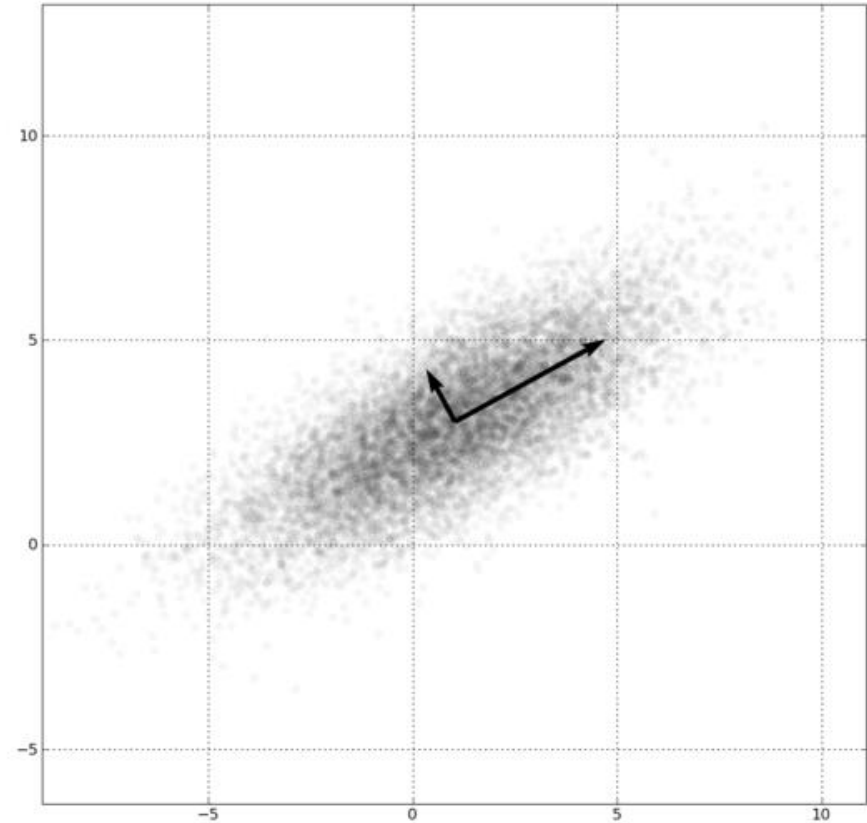
clustering

regression

dimensionality
reduction

Dimensionality Reduction

- **PCA**, ICA, LLE, Isomap, *Autoencoder*
- PCA is the most important technique to know. It takes advantage of correlations in data dimensions to produce the best possible lower dimensional representation based on linear projections (minimizes reconstruction error).
- PCA should be used for dimensionality reduction, not for discovering patterns or making predictions. Don't try to assign semantic meaning to the bases.



Machine Learning Problems

Supervised Learning

Unsupervised Learning

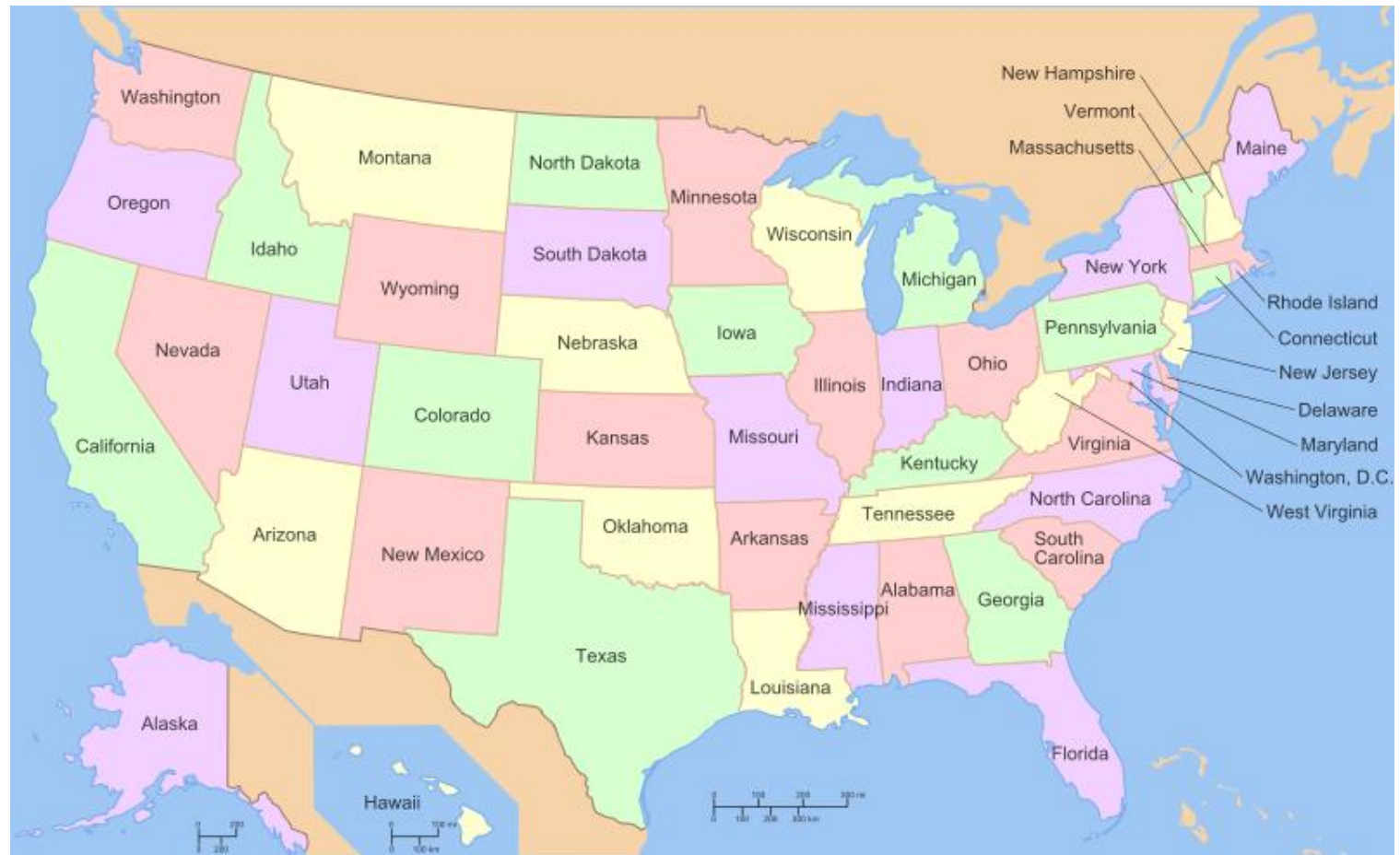
Discrete
Continuous

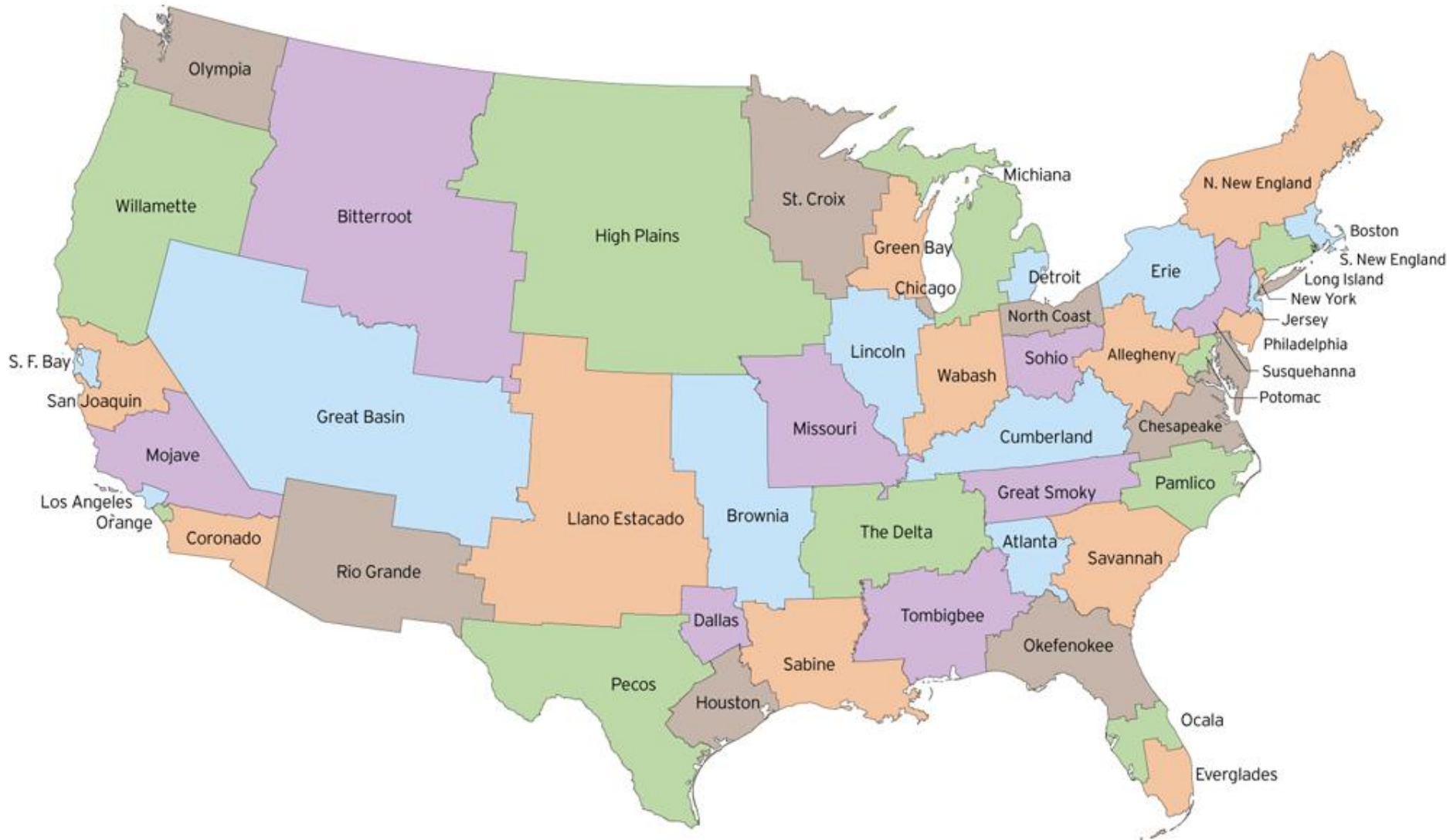
classification or
categorization

clustering

regression

dimensionality
reduction





- <http://fakeisthenewreal.org/reform/>

• <http://fakeisthenewreal.org/reform/>

- <http://fakeisthenewreal.org/reform/>

Clustering example: image segmentation

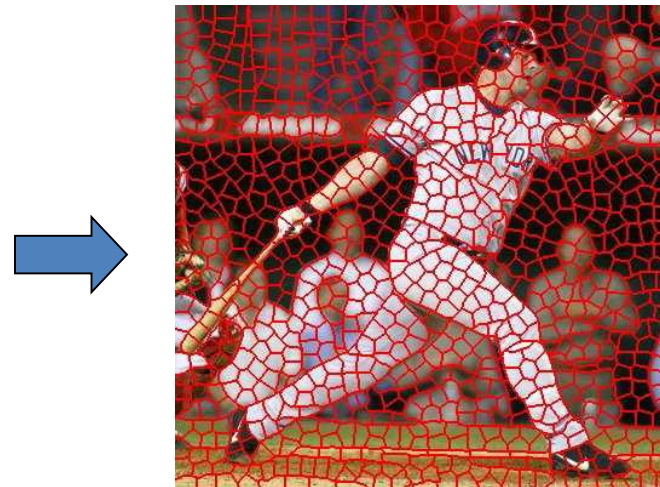
Goal: Break up the image into meaningful or perceptually similar regions



Segmentation for feature support or efficiency



[Felzenszwalb and Huttenlocher 2004]



[Shi and Malik 2001]

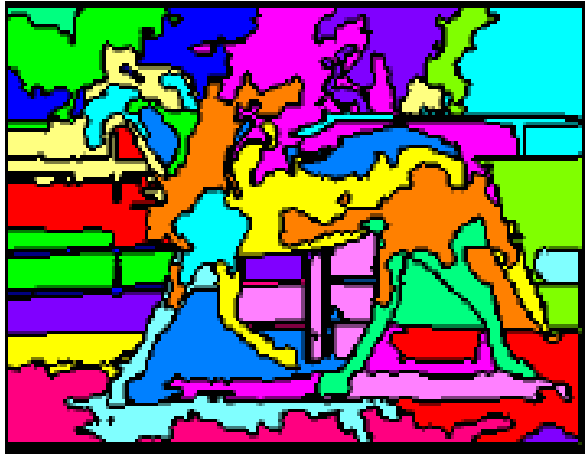
[Hoiem et al. 2005, Mori 2005]

Slide: Derek Hoiem

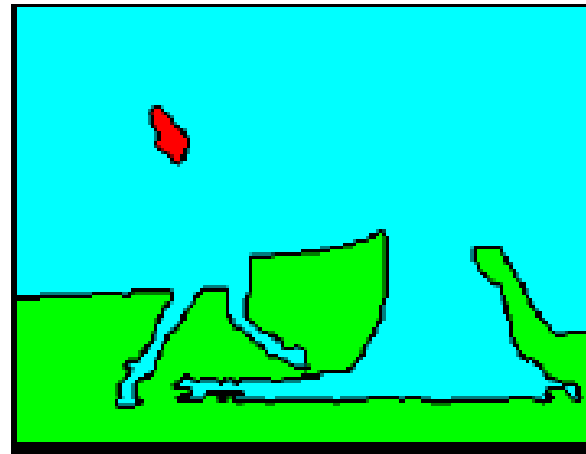
Segmentation as a result



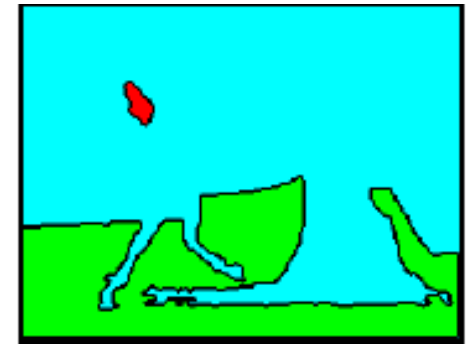
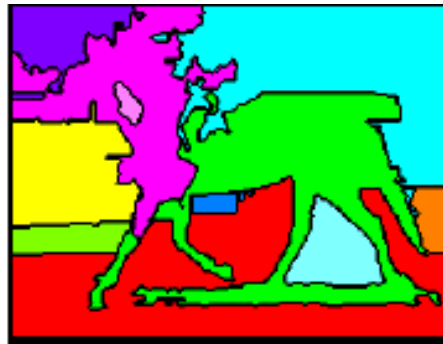
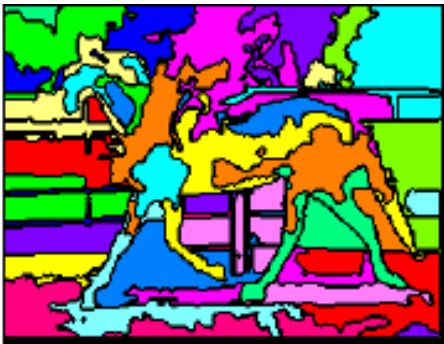
Types of segmentations



Oversegmentation



Undersegmentation



Multiple Segmentations

Clustering: group together similar points and represent them with a single token

Key Challenges:

- 1) What makes two points/images/patches similar?
- 2) How do we compute an overall grouping from pairwise similarities?

How do we cluster?

- K-means
 - Iteratively re-assign points to the nearest cluster center
- Agglomerative clustering
 - Start with each point as its own cluster and iteratively merge the closest clusters
- Mean-shift clustering
 - Estimate modes of pdf
- Spectral clustering
 - Split the nodes in a graph based on assigned links with similarity weights

Clustering for Summarization

Goal: cluster to minimize variance in data given clusters

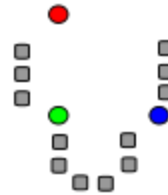
- Preserve information

$$\mathbf{c}^*, \boldsymbol{\delta}^* = \underset{\mathbf{c}, \boldsymbol{\delta}}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij} \left(\underset{\substack{\text{Cluster center} \\ \nwarrow}}{\mathbf{c}_i} - \underset{\substack{\text{Data} \\ \nwarrow}}{\mathbf{x}_j} \right)^2$$

\nearrow
Whether \mathbf{x}_j is assigned to \mathbf{c}_i

K-means algorithm

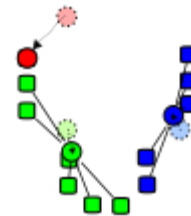
1. Randomly select K centers



2. Assign each point to nearest center

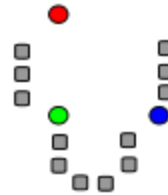


3. Compute new center (mean) for each cluster

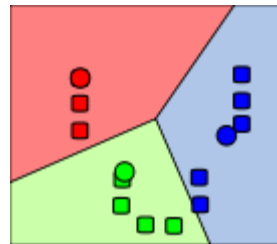


K-means algorithm

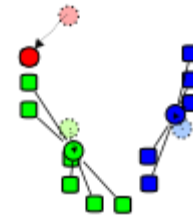
1. Randomly select K centers



2. Assign each point to nearest center



3. Compute new center (mean) for each cluster



Back to 2



K-means

1. Initialize cluster centers: \mathbf{c}^0 ; $t=0$

2. Assign each point to the closest center

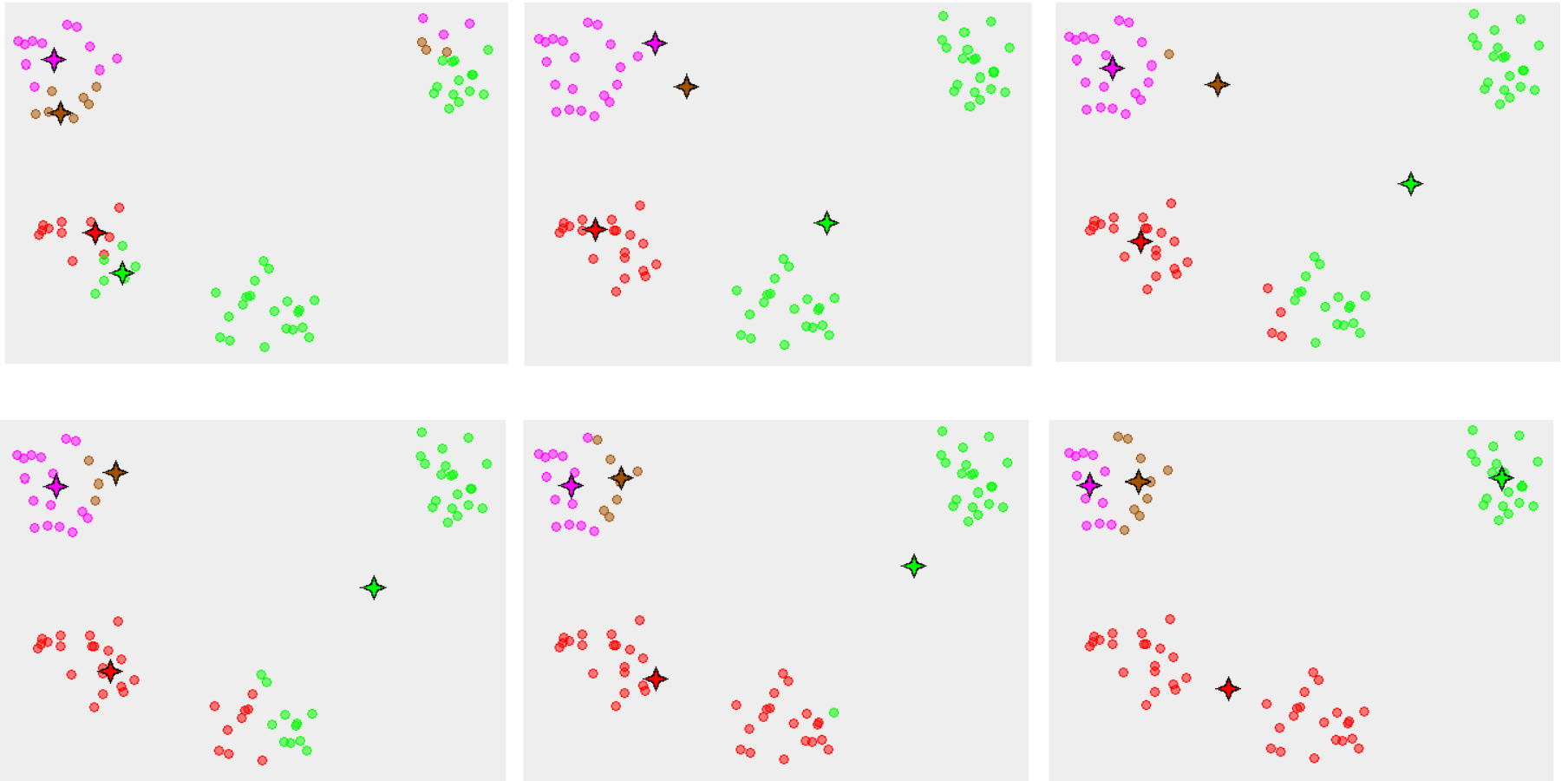
$$\delta^t = \underset{\delta}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij} \left(\mathbf{c}_i^{t-1} - \mathbf{x}_j \right)^2$$

3. Update cluster centers as the mean of the points

$$\mathbf{c}^t = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^t \left(\mathbf{c}_i - \mathbf{x}_j \right)^2$$

4. Repeat 2-3 until no points are re-assigned ($t=t+1$)

K-means converges to a local minimum



K-means: design choices

- Initialization
 - Randomly select K points as initial cluster center
 - Or greedily choose K points to minimize residual
- Distance measures
 - Traditionally Euclidean, could be others
- Optimization
 - Will converge to a *local minimum*
 - May want to perform multiple restarts

K-means clustering using intensity or color

Image



Clusters on intensity



Clusters on color



How to evaluate clusters?

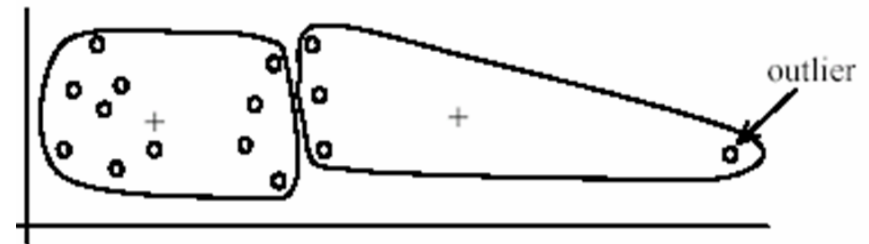
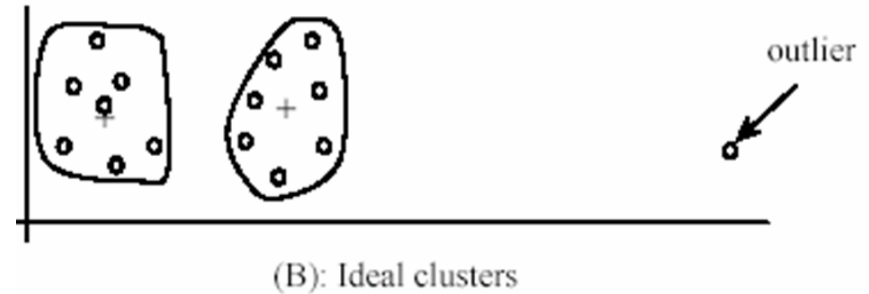
- Generative
 - How well are points reconstructed from the clusters?
- Discriminative
 - How well do the clusters correspond to labels?
 - Purity
 - Note: unsupervised clustering does not aim to be discriminative

How to choose the number of clusters?

- Validation set
 - Try different numbers of clusters and look at performance
 - When building dictionaries (discussed later), more clusters typically work better

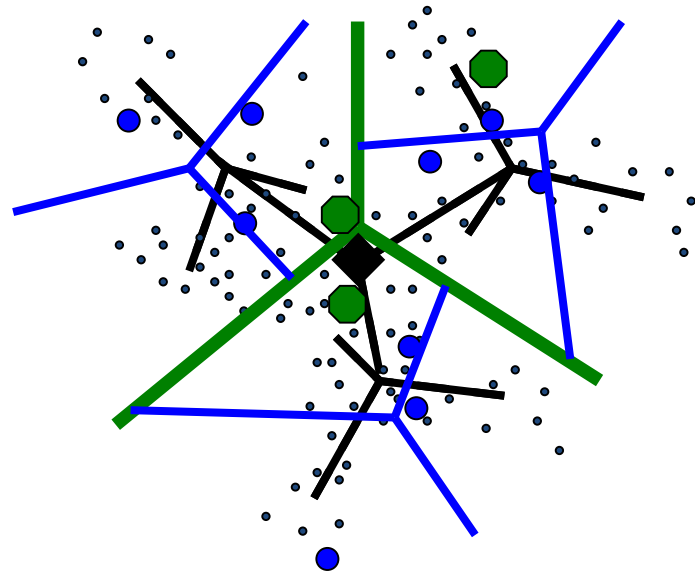
K-Means pros and cons

- Pros
 - Finds cluster centers that minimize conditional variance (good representation of data)
 - Simple and fast*
 - Easy to implement
- Cons
 - Need to choose K
 - Sensitive to outliers
 - Prone to local minima
 - All clusters have the same parameters (e.g., distance measure is non-adaptive)
 - *Can be slow: each iteration is $O(KNd)$ for N d-dimensional points
- Usage
 - Rarely used for pixel segmentation

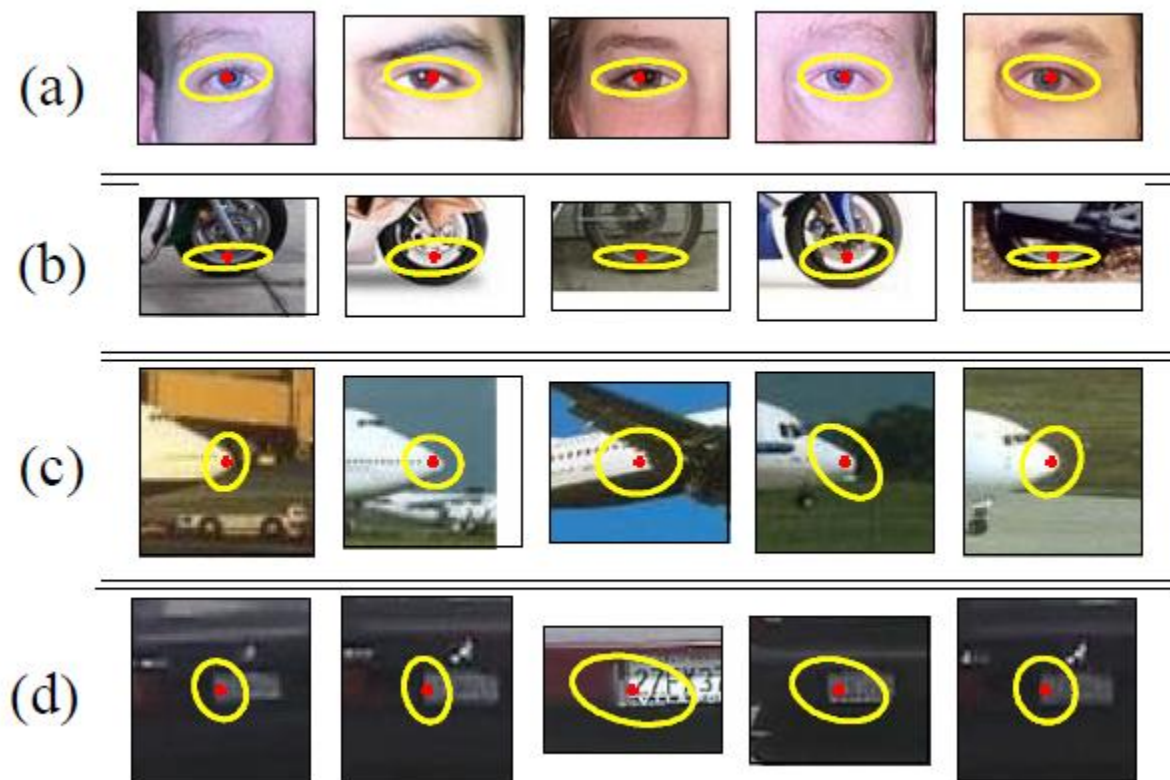


Building Visual Dictionaries

1. Sample patches from a database
 - E.g., 128 dimensional SIFT vectors
2. Cluster the patches
 - Cluster centers are the dictionary
3. Assign a codeword (number) to each new patch, according to the nearest cluster



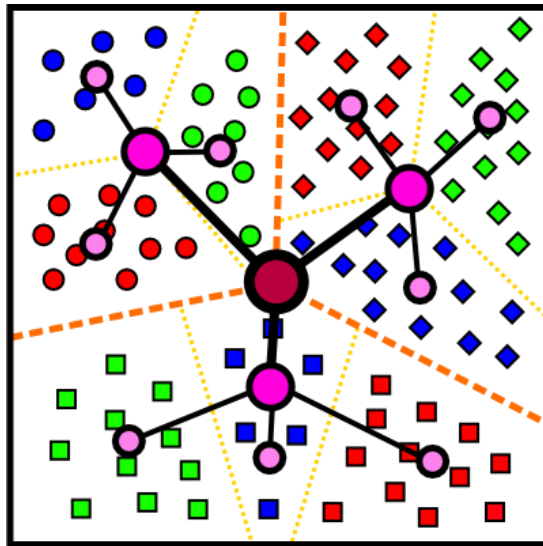
Examples of learned codewords



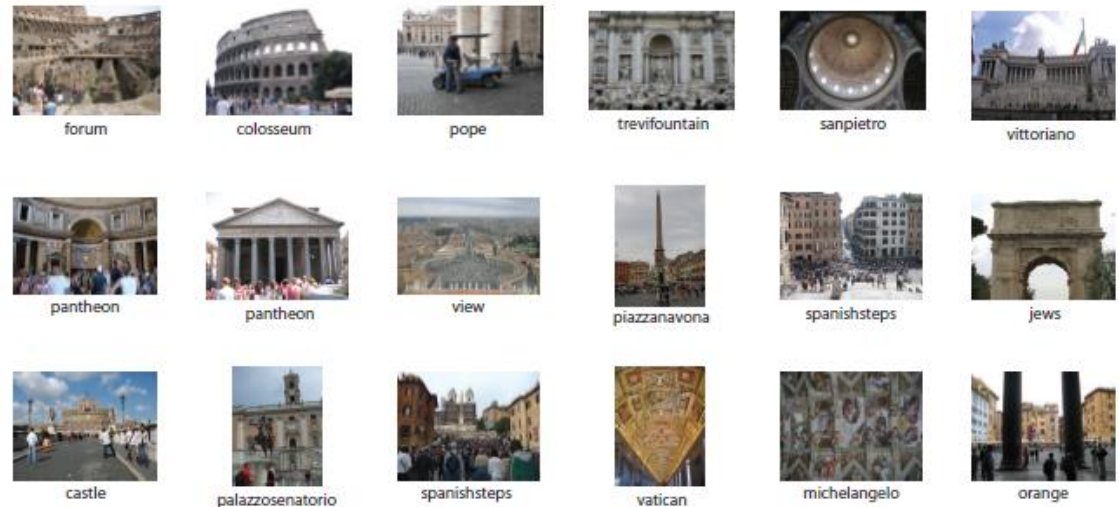
Most likely codewords for 4 learned “topics”
EM with multinomial (problem 3) to get topics

Which algorithm to use?

- Quantization/Summarization: K-means
 - Aims to preserve variance of original data
 - Can easily assign new point to a cluster



Quantization for
computing histograms

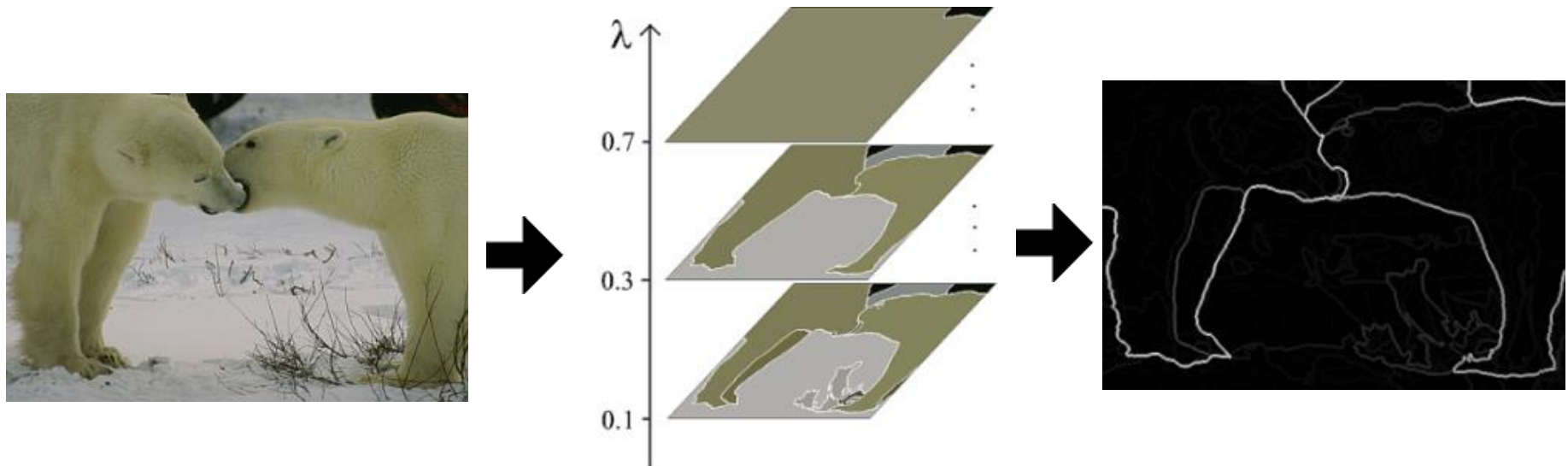


Summary of 20,000 photos of Rome using
“greedy k-means”

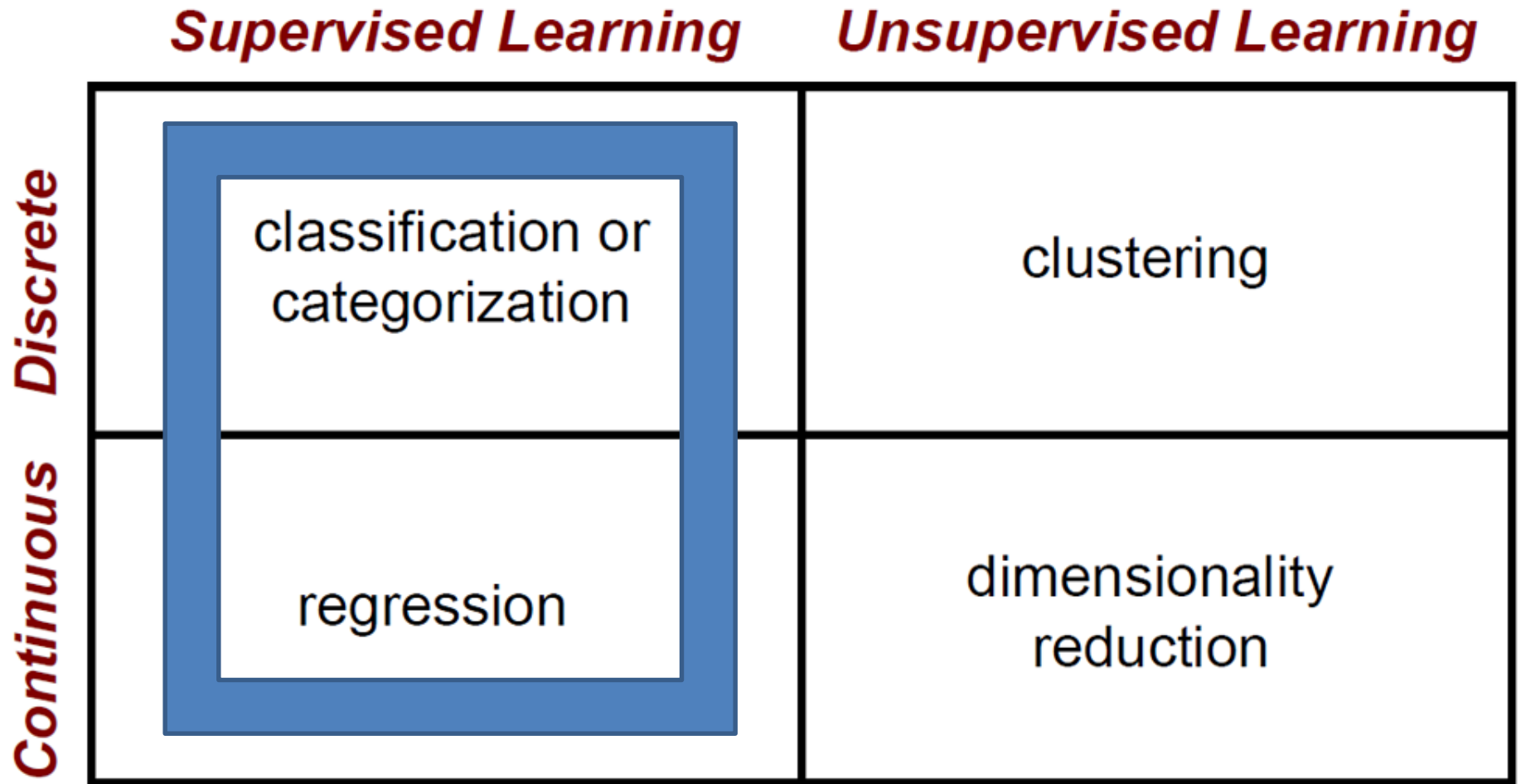
<http://grail.cs.washington.edu/projects/canonview/>

Which algorithm to use?

- Image segmentation: agglomerative clustering
 - More flexible with distance measures (e.g., can be based on boundary prediction)
 - Adapts better to specific data
 - Hierarchy can be useful



Machine Learning Problems



The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

$$f(\text{apple image}) = \text{"apple"}$$

$$f(\text{tomato image}) = \text{"tomato"}$$

$$f(\text{cow image}) = \text{"cow"}$$

The machine learning framework

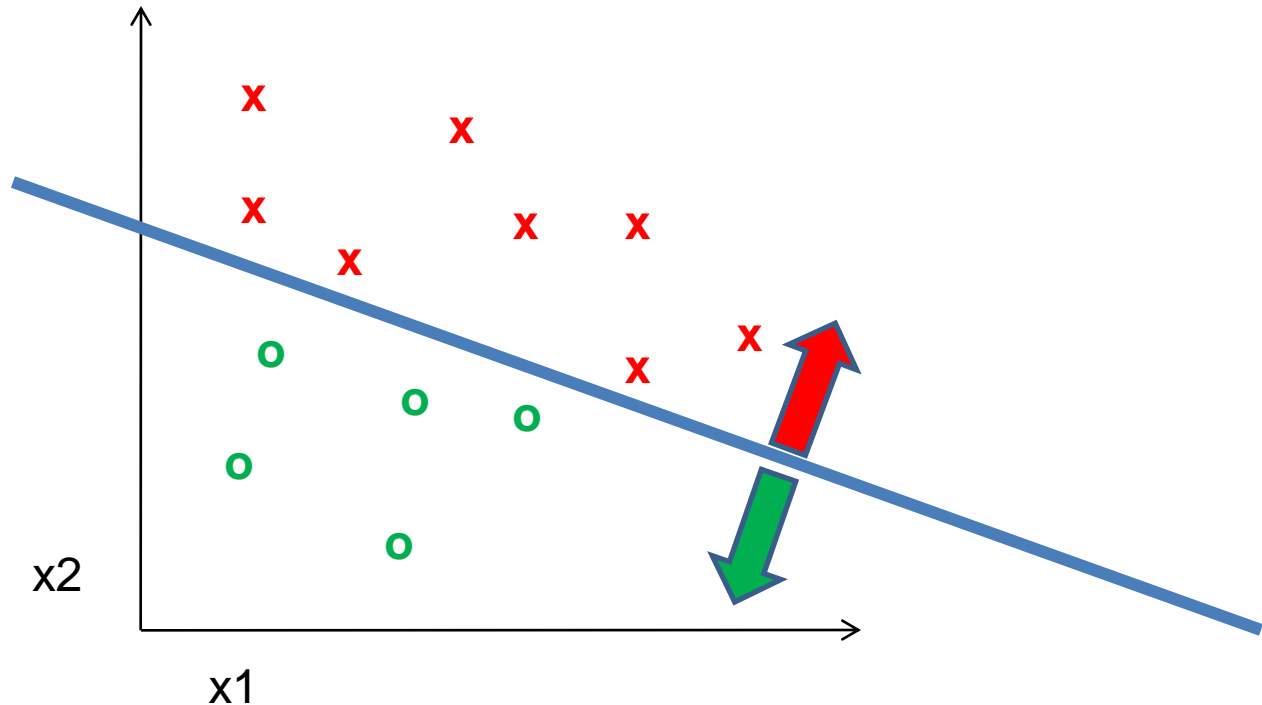
$$y = f(\mathbf{x})$$

output prediction function Image feature

- **Training:** given a *training* set of labeled examples $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* \mathbf{x} and output the predicted value $y = f(\mathbf{x})$

Learning a classifier

Given some set of features with corresponding labels, learn a function to predict the labels from the features



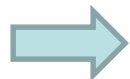
Steps

Training

Training
Images



Image
Features



Training
Labels



Training



Learned
model

Testing



Test Image



Image
Features



Learned
model



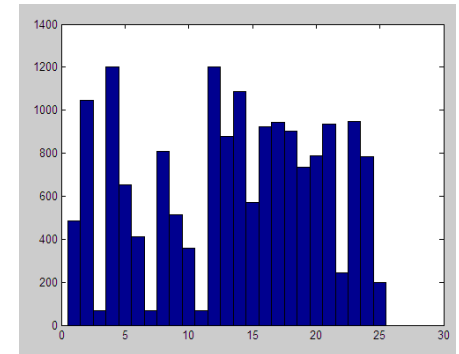
Prediction

Features

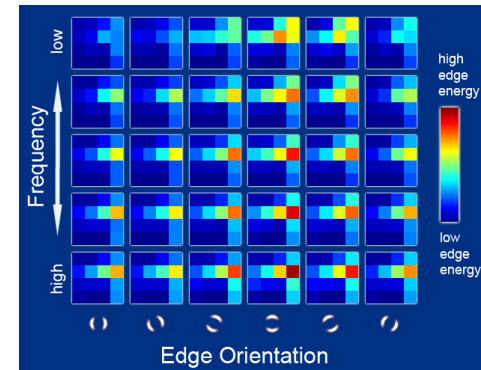
- Raw pixels



- Histograms



- GIST descriptors



- ...

One way to think about it...

- Training labels dictate that two examples are the same or different, in some sense
- Features and distance measures define visual similarity
- Classifiers try to learn weights or parameters for features and distance measures so that visual similarity predicts label similarity

Many classifiers to choose from

- SVM
- Neural networks
- Naïve Bayes
- Bayesian network
- Logistic regression
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- RBMs
- Deep Convolutional Network
- Etc.

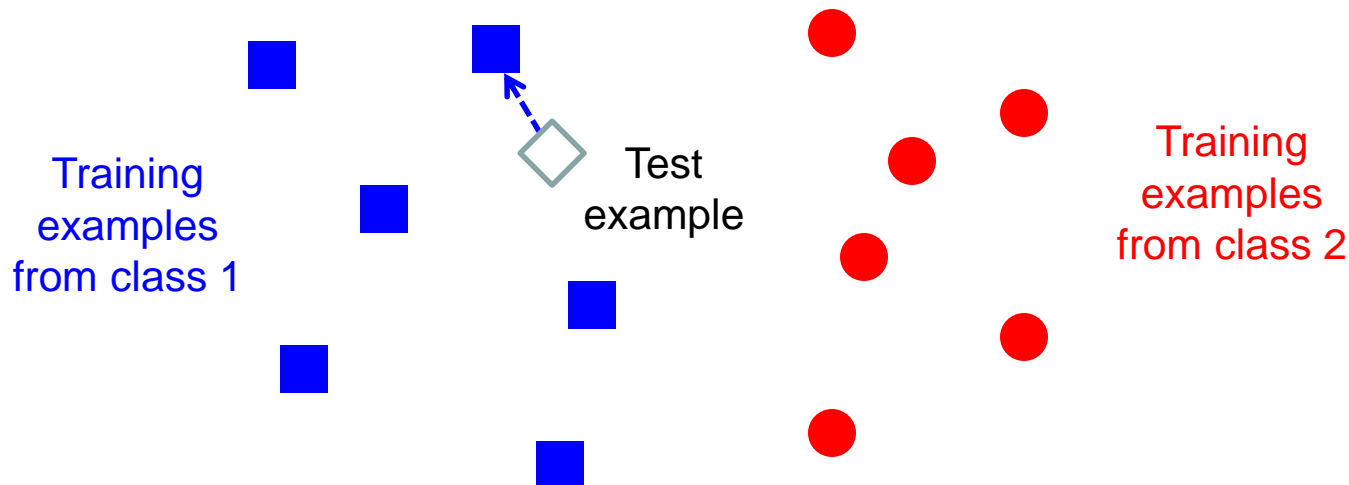
Which is the best one?

Claim:

The decision to *use* machine learning is more important than the choice of a *particular* learning method.

*Deep learning seems to be an exception to this, at the moment, probably because it is learning the feature representation.

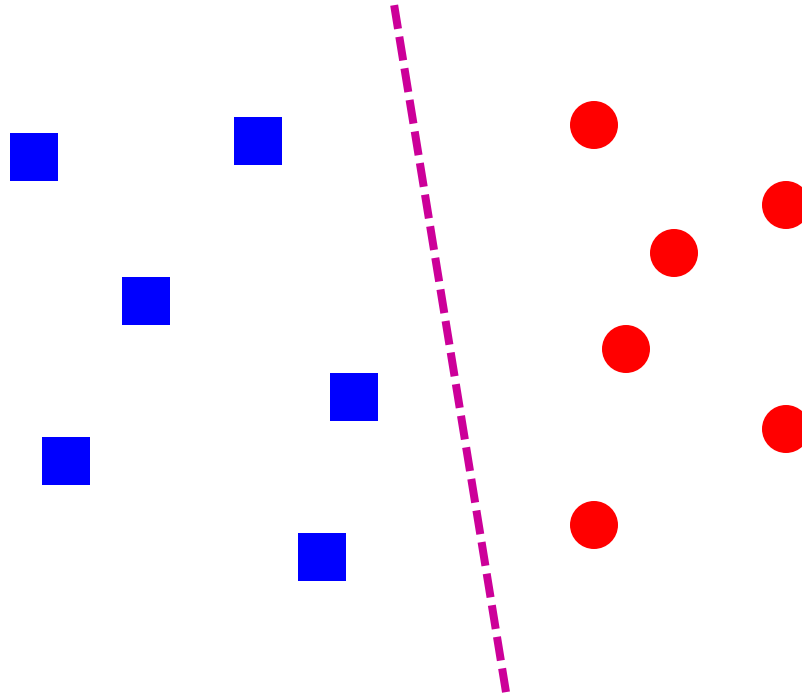
Classifiers: Nearest neighbor



$f(\mathbf{x}) = \text{label of the training example nearest to } \mathbf{x}$

- All we need is a distance function for our inputs
- No training required!

Classifiers: Linear



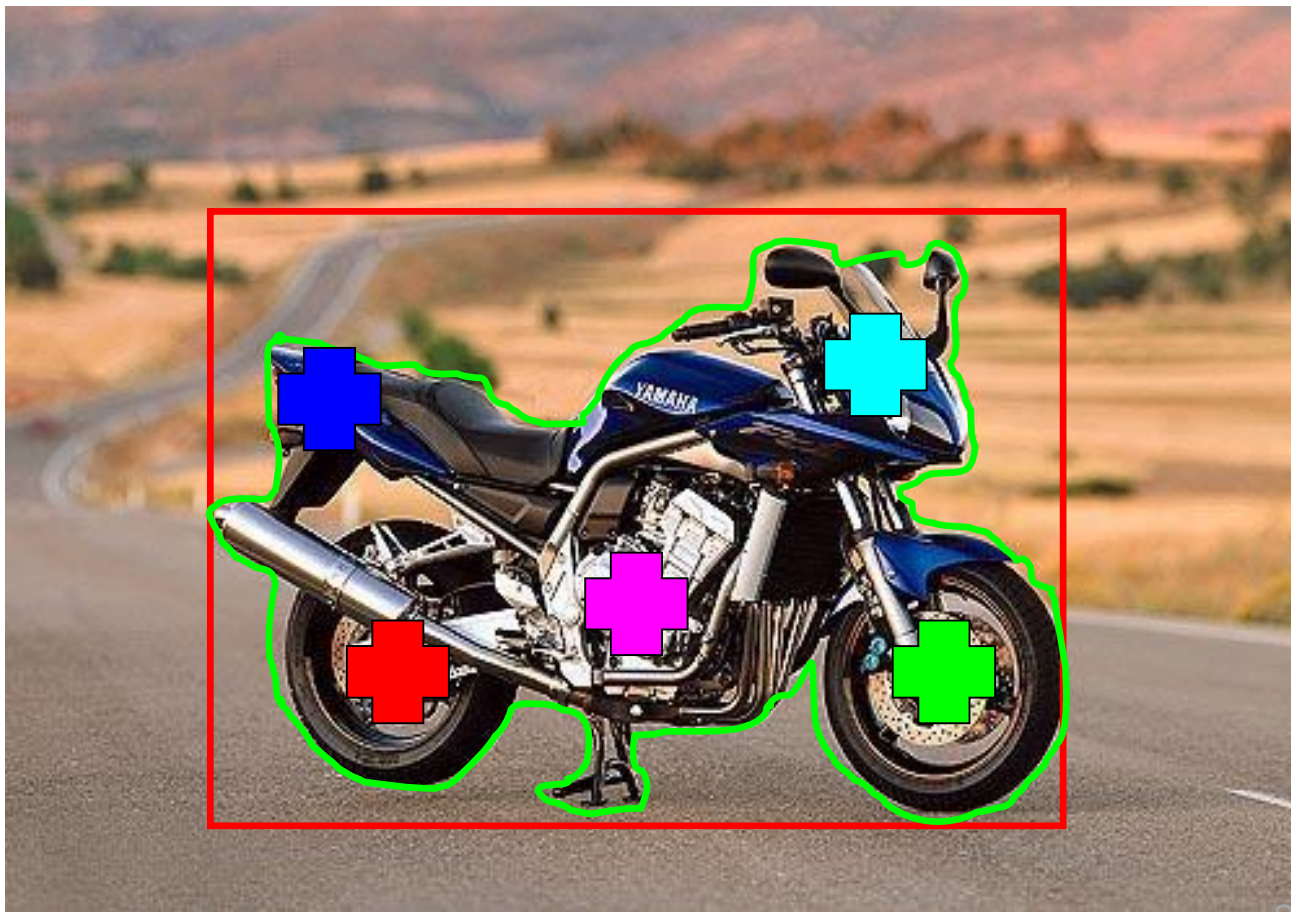
- Find a *linear function* to separate the classes:

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b)$$

Recognition task and supervision

- Images in the training set must be annotated with the “correct answer” that the model is expected to produce

Contains a motorbike



Spectrum of supervision

Less

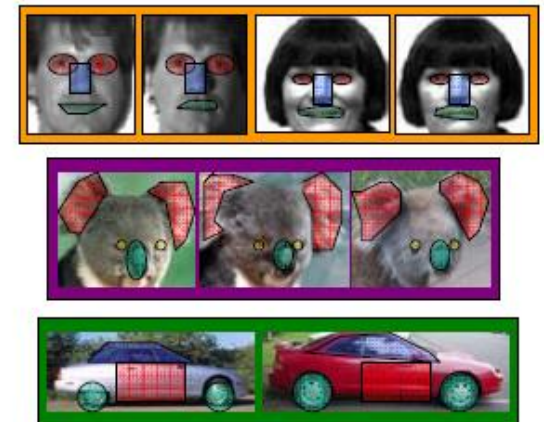
More



Unsupervised



“Weakly” supervised



Fully supervised

Definition depends on task

Generalization



Training set (labels known)



Test set (labels unknown)

- How well does a learned model generalize from the data it was trained on to a new test set?

To be continued...