Machine Learning

Applied to Computer Vision

Adín Ramírez Rivera

adin@ic.unicamp.br

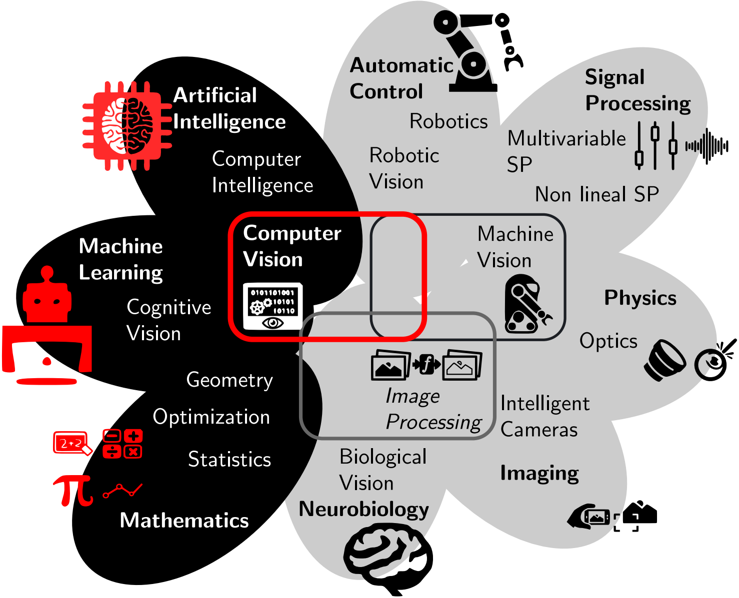
2nd. Semester 2016

Adapted from J. Hays, I. Guyon, E. Sudderth, M. Johnson, D. Hoiem

Overview

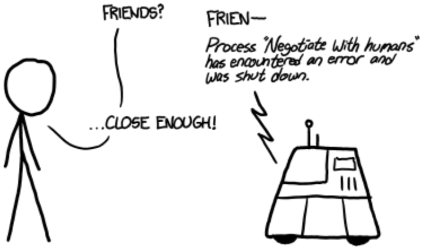
- Principal idea: make predictions or decisions through data
- We won't be getting much into the advanced details
- We will see the use of machine learning as tools (in general)

Computer Vision and Related Fields



Brief History of Computer Vision

- 1960: interpretation of synthetic worlds
- 1966: Minsky assigns a homework to a student, plug the camera to the computer and make it interpret what it is seeing
- 1970: progress interpreting selected images
- 1980: ANNs come and go, there is a tendency towards geometry and math
- 1990: face recognition and statistical analysis
- 2000: broader recognition, several databases appear, and we start processing videos
- 2010: deep learning
- 2030: robot revolution?

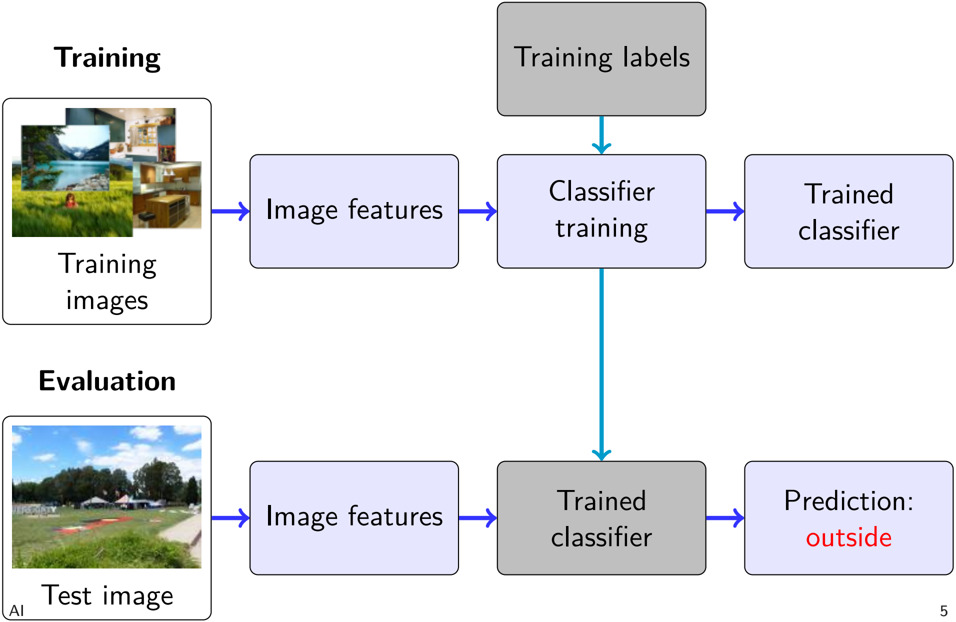


https://what-if.xkcd.com/5/

Machine Learning Impact

- It is the major export from computing to other fields of science
- Some fields that use it
  - High energy physics
  - Market analysis
  - Systems diagnostics
  - Bio-informatics
  - Text classification
  - Machine vision
  - ...

Image Classification



Examples

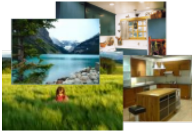
Scene classification

- ¿Is this a kitchen?



Image Features

Training



Training images

Image features

Training labels

Classifier training

Trained classifier

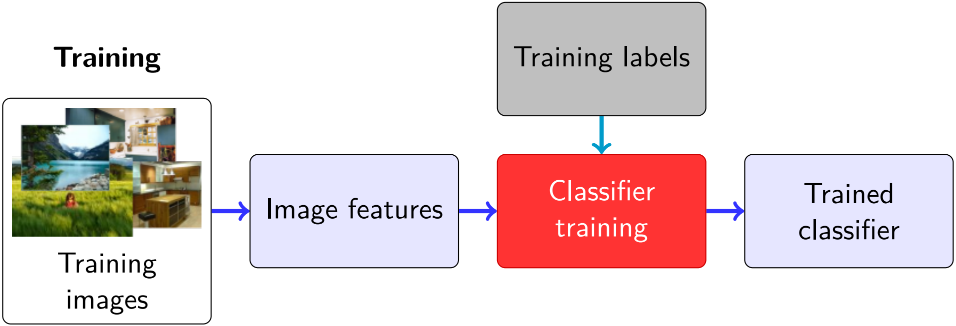
General Principles of Representation

- Coverage
  - Lets make sure that all the relevant information is covered and captured
- Concise
  - Minimize the number of features without sacrificing coverage
- Direct
  - The ideal features are used in prediction

Image Representation

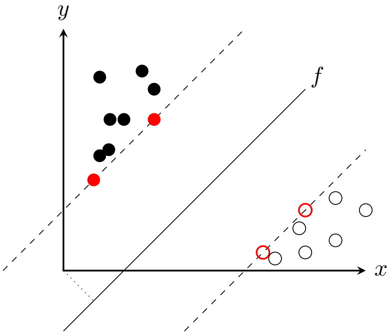
- Templates
  - Intensity
  - Gradients
  - etc.
- Histograms
  - Color
  - Texture
  - SIFT
  - Descriptors
  - etc.

Classifiers



Learn a Classifier

- Given a set of features with their corresponding labels
- Learn a function f that predicts the labels of the features



Many Classifiers

- Support Vector Machines (SVM)
- Neural Networks
- Naïve Bayes
- Bayesian Networks
- Logistic Regression
- Random Forests
- Boosted Decision Trees
- k-Nearest Neighbors
- etc.
- Which one is the best?

One way of thinking about them

- Training labels identify the examples that are equal or different (according to the classification problem)
- Features and the distance measures define a visual similarity
- Classifiers try to learn the weights or parameters for the features and the distance measures such that the visual similarity predicts the similarity of the labels
- The decision of using machine learning methods is more important than the particular method to use

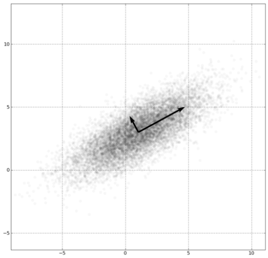
Machine Learning Problems

	Supervised Learning	Unsupervised Learning
Discrete	Classification or categorization	Clustering
Continuous	Regression	Dimensionality Reduction

Machine Learning Problems

	Supervised Learning	Unsupervised Learning
Discrete	Classification or categorization	Clustering
Continuous	Regression	Dimensionality Reduction

Reduction



- **PCA**, ICA, LLE, Isomap
- PCA (Principal Component Analysis) is a the most common technique
- Takes advantage of the correlation in the dimensions to produce a new representation in lower dimensional space
- PCA must be used to reduce the dimensionality of data, not to discover patterns or predict
- Do not assign a semantic value to the new base

Machine Learning Problems

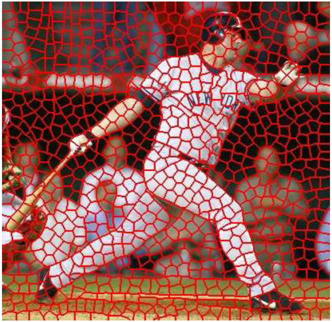
	Supervised Learning	Unsupervised Learning
Discrete	Classification or categorization	Clustering
Continuous	Regression	Dimensionality Reduction

Clustering Examples

- Goal: decompose and image into its similar parts that are relevant

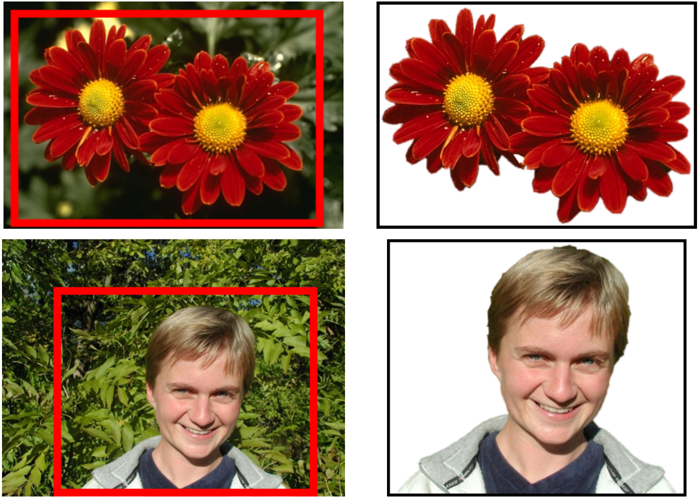
Segmentation for Efficiency
Superpixels


Felzenswalb and Huttenlocher 2004

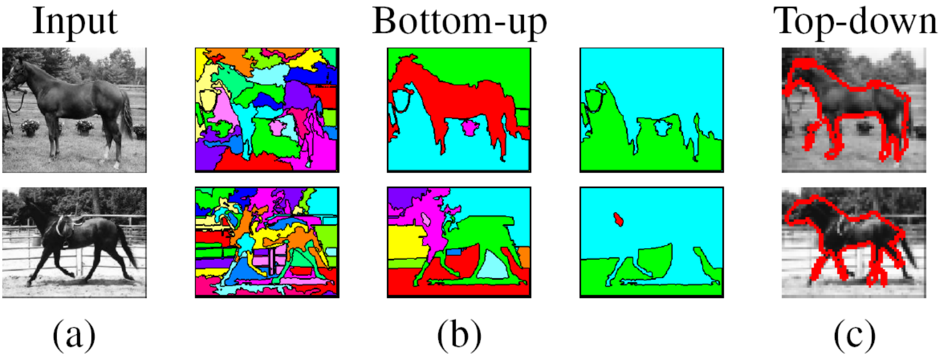



Shi and Malik 2001

Segmentation as Result



Types of Segmentation



Objectives

- The *clustering* objectives are grouping similar points and representations into the same *token*
- Key challenges
  - What makes two points, images, or patches similar?
  - How do we compute a global grouping from the similar pairs?

Why do we group?

- Summarize the data
  - Look at big volumes of data
  - Compression or elimination of noised based on data (patches)
  - Representation of a continuous vector (and probably big) with a cluster number
- Count
  - Histograms of texture, color, feature vectors (SIFT)
- Segment
  - Separate the images into several regions
- Predict
  - Images of the same cluster can have same labels

How do we generate the clusters?

- k-means
  - Iteratively re assign the points to the closer cluster according to the center of mass of the cluster
- Agglomerative Clustering
  - Start with each point as its own cluster
  - Iteratively mix the closer clusters
- Mean-shift Clustering
  - Estimate the modes of the pdf
- Spectral Clustering
  - Divide the nodes of a graph based on the similarity weights of the edges

Clustering in a Nutshell

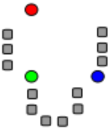
- Goal: create clusters to minimize the variance between the given data
- Preserve information

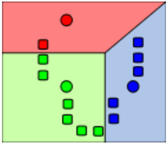
$$\mathbf{c}^*, \boldsymbol{\delta}^* = \arg\min_{\mathbf{c},\boldsymbol{\delta}} \frac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \boldsymbol{\delta}_{ij} \left(\mathbf{c}_i - \mathbf{x}_j\right)^2$$

  - $\mathbf{c}_i$ is the ith cluster center
  - $\boldsymbol{\delta}_{ij}$ whether $\mathbf{x}_j$ must be assigned to $\mathbf{c}_i$
  - $\mathbf{x}_j$ is the jth datum

K-means

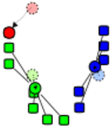
1. Pick k centers randomly

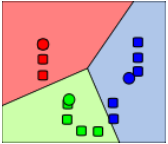




2. Assign each point to the closest center

3. Compute new centers





Iterate until convergence

K-means

1. Initialize the centers to the clusters $\mathbf{c}^0$, $t = 0$
2. Assign each point to the closest cluster

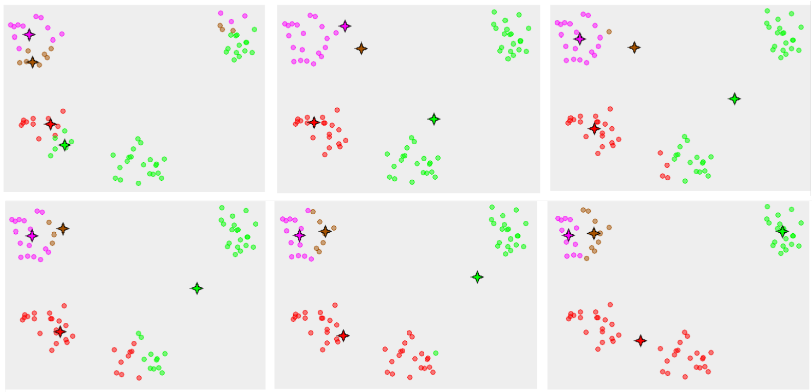
$$\boldsymbol{\delta}^t = \arg\min_{\boldsymbol{\delta}} \frac{1}{N} \sum_j^N \sum_i^K \boldsymbol{\delta}_{ij}^{t-1} \left(\mathbf{c}_i^{t-i} - \mathbf{x}_j \right)^2$$

3. Update the cluster centers using the mean of the points

$$\mathbf{c}_i^t = \frac{1}{N} \sum_j^N \boldsymbol{\delta}_{ij}^t \mathbf{x}_j$$

4. Repeat 2–3 until there is no point reassignment

K-means convergence to a local minima



Design decisions

- Initialization
  - Pick k random points as initial centers
  - Or greedily select k points to minimize the residual
- Distance measures
  - Traditionally, Euclidean ($\updownarrow_2$), but we can use others
- Optimization
  - Will converge to a local minima
  - We can do several runs

How to evaluate clusters?

- Generative
  - How good are the reconstructed points from the clusters?
- Discriminative
  - How good does the clusters correspond to the tags?
  - Purity
  - Note that non supervised clustering does not try to be discriminative

How do we pick the number of clusters?

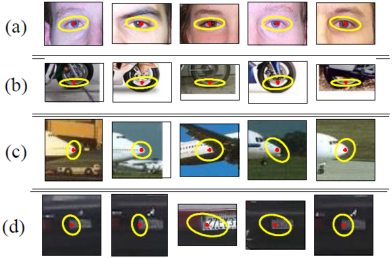
- We use a validation set
- Try different number of clusters and watch the performance
- When constructing dictionaries, the more the merrier

K-means advantages and disadvantages

- Advantages
  - Fin cluster centers that minimize the conditional variance (good representation of the data)
  - Simple and fast
  - Easy to implement
- Disadvantages
  - Need to pick K
  - Sensitive to outliers
  - Prone to local minima
  - Every cluster has the same parameters (e.g., the distance measure is not adaptive)
  - Can be slow: each iteration $O(KNd)$, N points of d dimensions
- Use
  - They are not used for pixel segmentation

Visual Dictionaries

- Samples of patches from a database (e.g., 128-dimensional vectors)
- Generate clusters from the patches
- The cluster centers are the dictionary
- Assign each *codeword* (number) to each new patch according to the nearest cluster



Sivic et al. ICCV 2005
http://www.robots.ox.ac.uk/~vgg/publications/papers/sivic05b.pdf

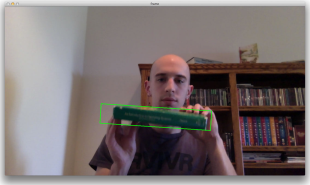
Important points

- Many classifiers, knowing which and what we need is important
- Decision of using a type of method is more important than the method itself
- Consider data, examples, previous knowledge, etc.
- Reduction problems, solution PCA (and friends)
- Clustering problems, solution k-means (and friends)

Next class

	Supervised Learning	Unsupervised Learning
Discrete	Classification or categorization	Clustering
Continuous	Regression	Dimensionality Reduction

Homework



- Install OpenCV
- Get demo camshift
  https://github.com/opencv/opencv/blob/master/samples/cpp/camshiftdemo.cpp
- Understand what the demo is doing
- Write **one page** report
  - Due on **one week**

Report

```
#include <iostream>
using namespace std;

int fib(int x) {
  if (x == 0)
  return 0;

  if (x == 1)
  return 1;

  return fib(x-1)+fib(x-2);
}

int main() {
  int n;
  cin >> n;
  cout << fib(n) << endl;
}
```

Bad example

The code reads an integer. Then calls a recursive function and computes the sum of two calls of the same function summing them by reducing the input by one and two, respectively.

Good example

A Fibonacci number, f_n, is computed through a recursive expression

$$f_n = f_{n-1} + f_{n-2},$$

where the initial values of the sequence are $f_0 = 0$ and $f_1 = 1$. This recursive equation is implemented as such through the function `fib`.