Machine Learning Crash Course

Computer Vision
James Hays

Photo: CMU Machine Learning Department protests G20

Slides: Isabelle Guyon, Erik Sudderth, Mark Johnson, Derek Hoiem
Recap: Multiple Views and Motion

• Epipolar geometry
  – Relates cameras in two positions
  – Fundamental matrix maps from a point in one image to a line (its epipolar line) in the other
  – Can solve for F given corresponding points (e.g., interest points)

• Stereo depth estimation
  – Estimate disparity by finding corresponding points along epipolar lines
  – Depth is inverse to disparity

• Motion Estimation
  – By assuming brightness constancy, truncated Taylor expansion leads to simple and fast patch matching across frames
  – Assume local motion is coherent
  – “Aperture problem” is resolved by coarse to fine approach

\[ \nabla I \cdot [u \ v]^T + I_t = 0 \]
Machine learning: Overview

• Core of ML: Making predictions or decisions from Data.

• This overview will not go in to depth about the statistical underpinnings of learning methods. We’re looking at ML as a tool.

• Take a machine learning course if you want to know more!
Impact of Machine Learning

• Machine Learning is arguably the greatest export from computing to other scientific fields.
Machine Learning Problems

<table>
<thead>
<tr>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>classification or categorization</td>
<td>clustering</td>
</tr>
<tr>
<td>regression</td>
<td>dimensionality reduction</td>
</tr>
</tbody>
</table>
Dimensionality Reduction

- **PCA, ICA, LLE, Isomap**

- 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

<table>
<thead>
<tr>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td></td>
</tr>
<tr>
<td>classification or categorization</td>
<td>clustering</td>
</tr>
<tr>
<td>Continuous</td>
<td></td>
</tr>
<tr>
<td>regression</td>
<td>dimensionality reduction</td>
</tr>
</tbody>
</table>
• http://fakeisthenewreal.org/reform/
The United States redrawn as Fifty States with Equal Population

Legend

- state borders
- international border
- American Indian reservation
- National Park
- Natural Earth, US Census
- Natural Earth, Global
- Cities
- Cheyenne, 50,000 to 100,000
- Shreveport, 100,000 to 500,000
- Baltimore, 500,000 and over

Hawaiian Islands part of Shasta

http://fakeisthenewrealt.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]

[Hoiem et al. 2005, Mori 2005]

[Shi and Malik 2001]
Segmentation as a result

Rother et al. 2004
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?
Why do we cluster?

• Summarizing data
  – Look at large amounts of data
  – Patch-based compression or denoising
  – Represent a large continuous vector with the cluster number

• Counting
  – Histograms of texture, color, SIFT vectors

• Segmentation
  – Separate the image into different regions

• Prediction
  – Images in the same cluster may have the same labels
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

\[ \mathbf{c}^*, \delta^* = \arg\min_{\mathbf{c}, \delta} \frac{1}{N} \sum_{j} \sum_{i} \delta_{ij} (c_i - x_j)^2 \]

Whether \( x_j \) is assigned to \( c_i \)

Cluster center

Data
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

K-means

1. Initialize cluster centers: \( c^0 \); \( t=0 \)

2. Assign each point to the closest center

\[
\delta^t = \arg\min_\delta \frac{1}{N} \sum_j \sum_i \delta_{ij} \left( c_i^{t-1} - x_j \right)^2
\]

3. Update cluster centers as the mean of the points

\[
c^t = \arg\min_c \frac{1}{N} \sum_j \sum_i \delta_{ij} \left( c_i - x_j \right)^2
\]

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

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

1. Say “Every point is its own cluster”
Agglomerative clustering

1. Say “Every point is its own cluster”
2. Find “most similar” pair of clusters
Agglomerative clustering

1. Say “Every point is its own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster
Agglomerative clustering

1. Say “Every point is its own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster
4. Repeat
Agglomerative clustering

1. Say “Every point is its own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster
4. Repeat
Agglomerative clustering

How to define cluster similarity?
- Average distance between points, maximum distance, minimum distance
- Distance between means or medoids

How many clusters?
- Clustering creates a dendrogram (a tree)
- Threshold based on max number of clusters or based on distance between merges
Conclusions: Agglomerative Clustering

Good
• Simple to implement, widespread application
• Clusters have adaptive shapes
• Provides a hierarchy of clusters

Bad
• May have imbalanced clusters
• Still have to choose number of clusters or threshold
• Need to use an “ultrametric” to get a meaningful hierarchy
Mean shift segmentation


• Versatile technique for clustering-based segmentation
Mean shift algorithm

- Try to find *modes* of this non-parametric density
Kernel density estimation function

\[ \hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right) \]

Gaussian kernel

\[ K \left( \frac{x - x_i}{h} \right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}. \]
Mean shift
Mean shift

Slide by Y. Ukrainitz & B. Sarel
Mean shift

Region of interest
Center of mass
Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean shift

Region of interest

Center of mass

Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean shift

Region of interest
Center of mass
Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean shift

Region of interest
Center of mass
Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean shift

Region of interest
Center of mass

Slide by Y. Ukrainitz & B. Sarel
Simple Mean Shift procedure:
- Compute mean shift vector
- Translate the Kernel window by $\mathbf{m}(\mathbf{x})$

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^{n} \mathbf{x}_i g \left( \frac{||\mathbf{x} - \mathbf{x}_i||^2}{h} \right)}{\sum_{i=1}^{n} g \left( \frac{||\mathbf{x} - \mathbf{x}_i||^2}{h} \right)}$$

$g(\mathbf{x}) = -k'(\mathbf{x})$
Attraction basin

- **Attraction basin**: the region for which all trajectories lead to the same mode
- **Cluster**: all data points in the attraction basin of a mode
Attraction basin
Mean shift clustering

• The mean shift algorithm seeks *modes* of the given set of points

1. Choose kernel and bandwidth

2. For each point:
   a) Center a window on that point
   b) Compute the mean of the data in the search window
   c) Center the search window at the new mean location
   d) Repeat (b,c) until convergence

3. Assign points that lead to nearby modes to the same cluster
Segmentation by Mean Shift

- Compute features for each pixel (color, gradients, texture, etc)
- Set kernel size for features $K_f$ and position $K_s$
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that are within width of $K_f$ and $K_s$
Mean shift segmentation results

Comaniciu and Meer 2002
Mean shift pros and cons

• Pros
  – Good general-practice segmentation
  – Flexible in number and shape of regions
  – Robust to outliers

• Cons
  – Have to choose kernel size in advance
  – Not suitable for high-dimensional features

• When to use it
  – Oversegmentation
  – Multiple segmentations
  – Tracking, clustering, filtering applications
Spectral clustering

Group points based on links in a graph
Cuts in a graph

Normalized Cut

• a cut penalizes large segments
• fix by normalizing for size of segments

\[
N_{\text{cut}}(A, B) = \frac{\text{cut}(A, B)}{\text{volume}(A)} + \frac{\text{cut}(A, B)}{\text{volume}(B)}
\]

• \(\text{volume}(A) = \) sum of costs of all edges that touch \(A\)

Source: Seitz
Normalized cuts for segmentation
Which algorithm to use?

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

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

http://www.cs.berkeley.edu/~arbelaez/UCM.html
Clustering

Key algorithm

• K-means
Machine Learning Problems

<table>
<thead>
<tr>
<th>Supervised Learning</th>
<th>Unsupervised Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td>Continuous</td>
</tr>
<tr>
<td>classification or categorization</td>
<td>clustering</td>
</tr>
<tr>
<td>regression</td>
<td>dimensionality reduction</td>
</tr>
</tbody>
</table>
To be continued