Words & Pictures

Clustering and Bag of Words

Many slides adapted from Svetlana Lazebnik, Fei-Fei Li, Rob Fergus, and Antonio Torralba
The paper bag is a remarkable contrivance. It serves us constantly and inconspicuously. It folds flat, yet opens into a structure that can stand open upon the table while we eat our sandwiches from it and chat with friends.

If we take the bag apart, we find it's made from a single paper cylinder. One end of the cylinder has been folded into a complex 3-dimensional pattern and finished off with a bit of paste. It would be, and once was, costly to make, because each fragile cylinder had to be folded manually into that hardy sack.
Bag-of-features models

Many slides adapted from Fei-Fei Li, Rob Fergus, and Antonio Torralba
Origin: Bag-of-words models

Origin: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  
  Salton & McGill (1983)
Origin: Bag-of-words models

Bags of features for image classification

1. Extract features
Bags of features for image classification

1. Extract features
2. Learn “visual vocabulary”
Bags of features for image classification

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
Bags of features for image classification

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies of “visual words”
1. Feature extraction

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005
1. Feature extraction

- Regular grid
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005

- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005
1. Feature extraction

- **Regular grid**
  - Vogel & Schiele, 2003
  - Fei-Fei & Perona, 2005

- **Interest point detector**
  - Csurka et al. 2004
  - Fei-Fei & Perona, 2005
  - Sivic et al. 2005

- **Other methods**
  - Random sampling (Vidal-Naquet & Ullman, 2002)
  - Segmentation-based patches (Barnard et al. 2003)
1. Feature extraction

- Compute SIFT descriptor [Lowe’99]
- Normalize patch
- Detect patches
  - [Mikołajczyk and Schmid ’02]
  - [Mata, Chum, Urban & Pajdla, ’02]
  - [Sivic & Zisserman, ’03]

Slide credit: Josef Sivic
1. Feature extraction
2. Learning the visual vocabulary
2. Learning the visual vocabulary

Slide credit: Josef Sivic
2. Learning the visual vocabulary
Clustering

- The assignment of objects into groups (called clusters) so that objects from the same cluster are more similar to each other than objects from different clusters.

- Often similarity is assessed according to a distance measure.

- Clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.
Clustering

- Clustering systems:
  - Unsupervised learning
  - Detect patterns in unlabeled data
    - E.g. group emails or search results
    - E.g. find categories of customers
    - E.g. detect anomalous program executions
  - Useful when don’t know what you’re looking for
  - Requires data, but no labels
  - Often get gibberish
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns

- What could “similar” mean?
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns

- What could “similar” mean?
  Any of the similarity metrics we talked about before (SSD, angle between vectors)
Feature Clustering

Clustering is the process of grouping a set of features into clusters of similar features.

Features within a cluster should be similar.

Features from different clusters should be dissimilar.
K-Means

- An iterative clustering algorithm
  - Pick K random points as cluster centers (means)
  - Alternate:
    - Assign data instances to closest mean
    - Assign each mean to the average of its assigned points
  - Stop when no points’ assignments change

source: Dan Klein
K-means clustering

- Want to minimize sum of squared Euclidean distances between points $x_i$ and their nearest cluster centers $m_k$

$$D(X, M) = \sum_{\text{cluster } k} \sum_{\text{point } i \text{ in cluster } k} (x_i - m_k)^2$$

source: Svetlana Lazebnik
K-means clustering

- Want to minimize sum of squared Euclidean distances between points $x_i$ and their nearest cluster centers $m_k$

$$D(X, M) = \sum_{\text{cluster } k} \sum_{\text{point } i \text{ in cluster } k} (x_i - m_k)^2$$

source: Svetlana Lazebnik
K-means

1. Ask user how many clusters they’d like.
   *(e.g. k=5)*
K-means

1. Ask user how many clusters they’d like. *(e.g. k=5)*

2. Randomly guess k cluster Center locations
K-means

1. Ask user how many clusters they'd like. (e.g. $k=5$)

2. Randomly guess $k$ cluster Center locations

3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)
K-means

1. Ask user how many clusters they’d like. *(e.g. k=5)*

2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it’s closest to.

4. Each Center finds the centroid of the points it owns
K-means

1. Ask user how many clusters they’d like. (e.g. $k=5$)
2. Randomly guess $k$ cluster Center locations
3. Each datapoint finds out which Center it’s closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!

Copyright © 2001, 2004, Andrew W. Moore
K-means continues

...
K-means continues
K-means continues
K-means continues

...
K-means continues...
K-means terminates
K-Means Getting Stuck

- A local optimum:
Initialization

- K-means is non-deterministic
  - Requires initial means
  - It does matter what you pick!
  - What can go wrong?
  - Various schemes for preventing this kind of thing: variance-based split / merge, initialization heuristics
Flat vs. Hierarchical clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning of docs into groups
  - Refine iteratively
  - Main algorithm: $K$-means

- Hierarchical algorithms
  - Create a hierarchy
  - Bottom-up, agglomerative
  - Top-down, divisive

Source: Hinrich Schutze
Hierarchical clustering

Our goal in hierarchical clustering is to create a hierarchy like the one we saw earlier in Reuters:

![Hierarchical clustering diagram]

We want to create this hierarchy **automatically**. We can do this either **top-down** or **bottom-up**. The best known bottom-up method is **hierarchical agglomerative clustering**.

Source: Hinrich Schutze
Hierarchical clustering strategies

- **Agglomerative clustering**
  - Start with each point in a separate cluster
  - At each iteration, merge two of the “closest” clusters

- **Divisive clustering**
  - Start with all points grouped into a single cluster
  - At each iteration, split the “largest” cluster

source: Svetlana Lazebnik
Agglomerative Clustering

- **Agglomerative clustering:**
  - First merge very similar instances
  - Incrementally build larger clusters out of smaller clusters

- **Algorithm:**
  - Maintain a set of clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there's only one cluster left

- Produces not one clustering, but a family of clusterings represented by a *dendrogram*

source: Dan Klein
Agglomerative Clustering

- How should we define “closest” for clusters with multiple elements?
- Many options
  - Closest pair (single-link clustering)
  - Farthest pair (complete-link clustering)
  - Average of all pairs
  - Distance between centroids

- Different choices create different clustering behaviors

source: Dan Klein
Divisive Clustering

• Top-down (instead of bottom-up as in Agglomerative Clustering)
• Start with all docs in one big cluster
• Then recursively split clusters
• Eventually each node forms a cluster on its own.

Source: Hinrich Schutze
Flat or hierarchical clustering?

• For high efficiency, use flat clustering (e.g. k means)
• For deterministic results: hierarchical clustering
• When a hierarchical structure is desired: hierarchical algorithm
• Hierarchical clustering can also be applied if K cannot be predetermined (can start without knowing K)

Source: Hinrich Schutze
2. Learning the visual vocabulary
2. Learning the visual vocabulary

Visual vocabulary

Clustering

Slide credit: Josef Sivic
From clustering to vector quantization

• Clustering is a common method for learning a visual vocabulary or codebook
  – Unsupervised learning process
  – Each cluster center produced by k-means becomes a codevector
  – Codebook can be learned on separate training set
  – Provided the training set is sufficiently representative, the codebook will be “universal”

• The codebook is used for quantizing features
  – A vector quantizer takes a feature vector and maps it to the index of the nearest codevector in a codebook
  – Codebook = visual vocabulary
  – Codevector = visual word
Example visual vocabulary

Fei-Fei et al. 2005
Image patch examples of visual words

Sivic et al. 2005
Visual vocabularies: Issues

• How to choose vocabulary size?
  – Too small: visual words not representative of all patches
  – Too large: quantization artifacts, overfitting

• Generative or discriminative learning?

• Computational efficiency
  – Vocabulary trees
    (Nister & Stewenius, 2006)
3. Image representation
Image classification (later)

- Given the bag-of-features representations of images from different classes, how do we learn a model for distinguishing them?