Image representations

Subhransu Maji

CMPSCI 670: Computer Vision

October 25/27, 2016
Administrativia

- Has everyone submitted a project abstract?
  - I’ll take a look at these over the weekend
  - Expect some comments if you have not talked to me already
Recall: Steps

Training

Training Images

Image Features

Training Labels

Training

Learned model

Testing

Test Image

Image Features

Prediction

Learned model

Slide credit: D. Hoiem
What is an image feature?

- Any **transformation** of an image into a **new representation**
- **Example:** transform an **image** into a **binary edge map**

*Image source: wikipedia*
Goals of a feature map

- Introduce invariance: illumination, deformations, position
- Preserve useful properties: shape, texture, color
- Make the subsequent learning easier
  - Ability to learn from a few examples
  - Can use simpler classifiers (prevent overfitting)

Figure 1.3: Variation in appearance due to a change in illumination

Image: [Fergus05]
The importance of good features

- Most learning methods are invariant to feature permutation
  - E.g., patch vs. pixel representation of images

Can you recognize the digits?

permute pixels

bag of pixels

permute patches

bag of patches
In general the optimal feature depends on:

- the nature of the recognition task
- the choice of subsequent classifier
  - “Shallow” learning — hand-crafted features + simple classifiers
  - “Deep” learning — end-to-end mapping of pixels to labels

Two families of features that work well with simple classifiers:

- Histogram of oriented gradients — captures overall \textit{shape}
- Bag of visual words — captures \textit{local shape} and \textit{texture}
Recall the feature matching step in image alignment

Problem with pixel values as a feature representation
  - illumination changes, small deformations

How can we design a feature that is robust to these changes?
SIFT features

- **Descriptor computation:**
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions
  - Additional step: normalize the descriptor to unit length

SIFT features

- **Descriptor computation:**
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions
  - Additional step: normalize the descriptor to unit length

- **Advantage over raw vectors of pixel values**
  - Gradients less sensitive to illumination change
  - Pooling of gradients over the sub-patches achieves robustness to small shifts, but still preserves some spatial information

Histogram of Oriented Gradients

- Can apply the same idea to the whole image
  - Preserves the overall structure of the image
  - Provides robustness to illumination and small deformations

- Introduced by Dalal and Triggs (CVPR 2005) for pedestrian detection
HOG feature: basic idea

- Divide the image into blocks
- Compute histograms of gradients for each regions
HOG feature: additional steps

- Additional steps for more invariance
  - Logarithm of the intensity values
  - Local contrast normalization

**Effect of bin-size**

- **Smaller bin-size**: better spatial resolution
- **Larger bin-size**: better invariance to deformations
- Optimal value depends on the object category being modeled
  - e.g. rigid vs. deformable objects
Works well for template matching

- Compute the HOG feature map for the image
- Convolve the template with the feature map to get score
  - Do this across scales (since we don’t know the size of the person)
- Find peaks of the response map (non-max suppression)
Example pedestrian detections

We will discuss object detection in detail later
Hand-crafting features

- Two families of features that work well with simple classifiers
  - Histogram of oriented gradients — captures overall shape
  - Bag of visual words — captures local shape and texture
Origin and motivation of the “bag of words” model

Algorithm pipeline

- Extracting local features
- Learning a dictionary — clustering using k-means
- Encoding methods — hard vs. soft assignment
- Spatial pooling — pyramid representations

Figure from Chatfield et al., 2011
Image as a “bag of patches”

Properties:
- Spatial structure is not preserved
- Invariance to large translations

Compare this to the HOG feature
Texture is characterized by the repetition of basic elements.

For stochastic textures, it is the identity of these elements, not their spatial arrangement, that matters.

Origin 1: Texture recognition

Origin 2: Bag-of-words models

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary  
  Salton & McGill (1983)
Origin 2: Bag-of-words models

Origin 2: Bag-of-words models

Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from Chatfield et al., 2011
Local feature extraction

- Regular grid or interest regions
Local feature extraction

Choices of descriptor:
- SIFT
- The patch itself
- ...

Detect patches

Compute descriptor

Normalize patch

Slide credit: Josef Sivic
Local feature extraction

Extract features from many images

Slide credit: Josef Sivic
Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from Chatfield et al., 2011
Learning a dictionary

Slide credit: Josef Sivic
Learning a dictionary

Clustering

Slide credit: Josef Sivic
Learning a dictionary

Visual vocabulary

Clustering

Slide credit: Josef Sivic
Clustering

- **Basic idea:** group together similar instances
- **Example:** 2D points
Basic idea: group together similar instances

Example: 2D points

What could similar mean?

- One option: small Euclidean distance (squared)

\[ \text{dist}(x, y) = \|x - y\|_2^2 \]

- Clustering results are crucially dependent on the measure of similarity (or distance) between points to be clustered
Clustering using k-means

- Given \((x_1, x_2, \ldots, x_n)\) partition the \(n\) observations into \(k (\leq n)\) sets \(S = \{S_1, S_2, \ldots, S_k\}\) so as to minimize the within-cluster sum of squared distances.

- The objective is to minimize:

\[
\arg \min_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2
\]
Lloyd’s algorithm for k-means

- Initialize k centers by picking k points randomly among all the points
- Repeat till convergence (or max iterations)
  - Assign each point to the nearest center (assignment step)
    \[\text{arg min}_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2\]
  - Estimate the mean of each group (update step)
    \[\text{arg min}_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2\]
k-means in action

![Image of k-means clustering in action](http://simplystatistics.org/2014/02/18/k-means-clustering-in-a-gif/)
k-means for image segmentation

Grouping pixels based on **intensity** similarity

feature space: intensity value (1D)
Example codebook

Source: B. Leibe
Another codebook

Source: B. Leibe
Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from Chatfield et al., 2011
Encoding methods

- Assigning words to features

Visual vocabulary

Also called hard assignment

partition of space
Encoding methods

- Assigning words to features

Visual vocabulary

![Diagram showing partition of space and hard assignment with large quantization error.]

- Different words: 1, 0, 0
- Similar features: 0, 0, 1

Large quantization error
Assigning words to features

- Visual vocabulary

Partition of space

Soft assignment

\[ \alpha_i \propto e^{-f(d(x, c_i))} \]

Assign high weights to centers that are close in practice non-zero to only k-nearest neighbors
Encoding methods

- Assigning words to features

**Visual vocabulary**

**soft assignment**

\[ \alpha_i \propto e^{-f(d(x,c_i))} \]

**partition of space**

**similar features**

**soft assignment**

<table>
<thead>
<tr>
<th>0.6</th>
<th>0</th>
<th>0.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0</td>
<td>0.6</td>
</tr>
</tbody>
</table>

**hard assignment**

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Encoding considerations

- What should be the size of the dictionary?
  - Too small: doesn’t capture the variability of the data (underfitting)
  - Too large: too few points per cluster (overfitting)

- Speed of embedding
  - Exact nearest neighbor is slow if the dictionary is large
  - Approximate nearest neighbor techniques
    - Search trees — organize data in a tree
    - Hashing — create buckets in the feature space
Lecture outline

- Origin and motivation of the “bag of words” model
- Algorithm pipeline
  - Extracting local features
  - Learning a dictionary — clustering using k-means
  - Encoding methods — hard vs. soft assignment
  - Spatial pooling — pyramid representations

Figure from Chatfield et al., 2011
Spatial pyramids

**pooling:** sum embeddings of local features within a region

Lazebnik, Schmid & Ponce (CVPR 2006)
Spatial pyramids

**pooling:** sum embeddings of local features within a region

Same motivation as **SIFT** — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)
Spatial pyramids

**pooling**: sum embeddings of local features within a region

Same motivation as **SIFT** — keep coarse layout information

Lazebnik, Schmid & Ponce (CVPR 2006)
Two families of features that work well with simple classifiers

- Histogram of oriented gradients — captures overall shape
- Bag of visual words — captures local shape and texture