Bag-of-Words Representation: Principles and Algorithms

Désiré Sidibé

Assistant Professor - Université de Bourgogne
LE2I - UMR CNRS 6306
dro-desire.sidibe@u-bourgogne.fr

25/06/2015

Outline

Outline

1. Introduction
2. BoW representation
3. Improvements & Extensions
4. Conclusion

A bit of history

- The Bag-of-Words (BoW) concept comes from text/documents retrieval community
- Assume you have to organize web pages into categories
  - Categories include **Sports**, **Movies**, **Cooking**
  - Your goal is to asssign each new webpage to one of these categories
  - You look for certain **words** in the webpages
  - For example, you might count how many times the word '*game*' appears in the webpage, or how many times the word '*recipe*' appears.
  - Then, you can assign a category based on the frequency of the words
- The set of words is called a **dictionary**
- And each webpage is represented by a **bag of words** from the dictionary

A bit of history

- Analysing a set of N documents, each represented by

$$\mathbf{x}^n = [x_1^n, \ldots, x_D^n]^T,$$

  where x_i^n counts how many times word i appears in document n
- D is typically very large and $\mathbf{x}$ will be very sparse
- The term-frequency (TF) is defiend as

$$tf_i^n = \frac{x_i^n}{\sum_i x_i^n}$$

- The inverse-document frequency (IDF)is given by

$$idf_i = \log \frac{N}{\text{\# of documents that contain term } i}$$

Bag-of-Words

A bit of history

- Analysing a set of N documents, each represented by
$$\mathbf{x}^n = [x_1^n, \ldots, x_D^n]^T,$$
where x_i^n counts how many times word i appears in document n
- The term-frequency - inverse document frequency (TF-IDF) is given by
$$x_i^n = tf_i^n \times idf_i$$
- TF-IDF gives high weight to terms that appear often in a document, but rarely amongst documents.

Latent Semantic Analysis

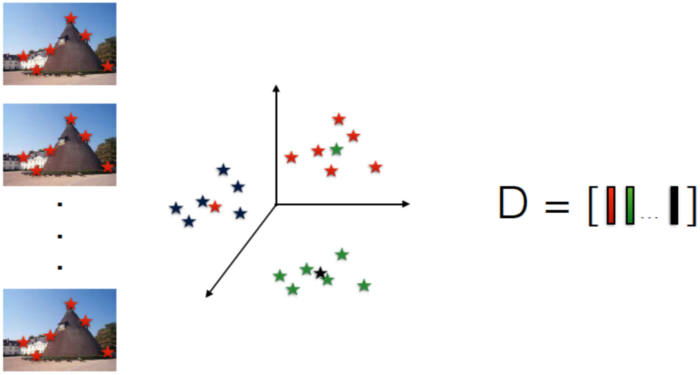
- Given a set of documents $\mathcal{D}$, the aim of LSA is to form a lower dimensional representation of each document
- An interpretation is that the principal directions define 'latent topics'

A bit of history

- This is the idea that was introduced to the computer vision community in the context of image category recognition
- The two seminal papers are :
  1. "*Video Google : a text retrieval approach to object matching in videos*", Sivic and Zisserman, ICCV 2003
  2. "*Visual categorization with bag of keypoints*", Csurka et al., ECCV Workshop 2004
- Paper 1 introduced the concept of visual vocabulary and used TF-IDF for retrieval
- Paper 2 introduced the concept of bag of features (later commonly used as BoW)

Key issues

- How to construct a visual dictionary ?



local features extraction clustering in feature space dictionary

Bag-of-Words

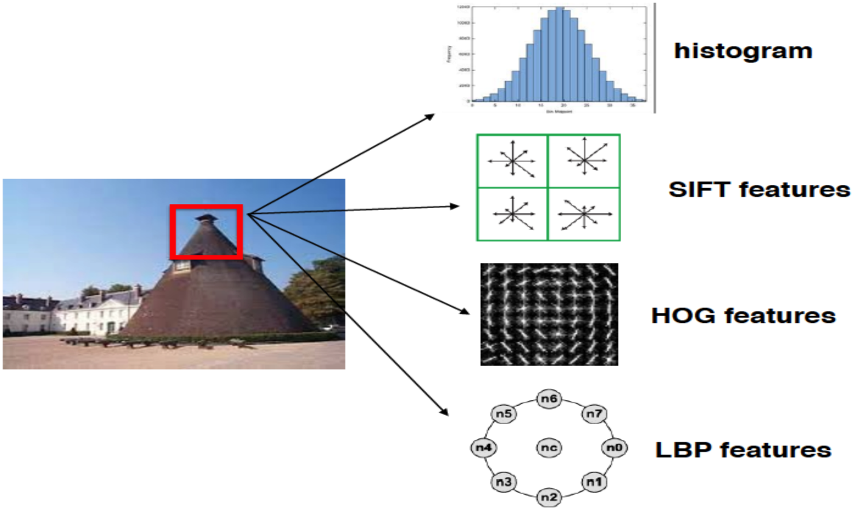
Key issues

- Vocabulary size ?
- Sampling strategy ?
- Clustering/Quantization ?
- Unsupervised vs Supervised ?

Outline

Local Features

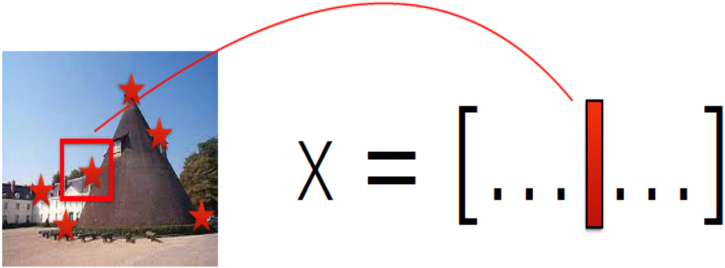
Many local features can be used



Sampling strategy

Keypoints detection

- Detect a set of keypoints (Harris, SIFT, etc)
- Extract local descriptors around each keypoint

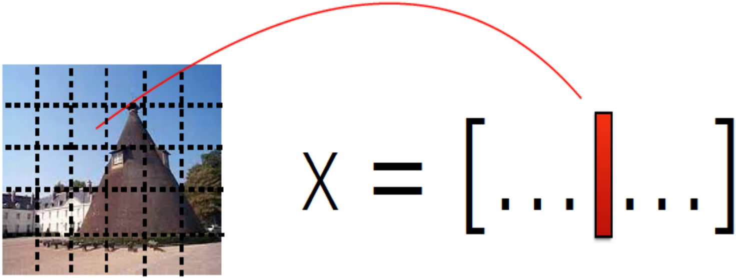


BoW representation

Sampling strategy

Dense sampling

- Divide image into local patches
- Extract local features from each patch



BoW representation

Clustering/Quantization

- For each image I_i we extract a set of low level descriptors and represent them as a feature matrix $\mathbf{X}_i$:

$$\mathbf{X}_i = \begin{bmatrix} | & | & & | \\ \mathbf{f}_i^1 & \mathbf{f}_i^2 & \dots & \mathbf{f}_i^{N_i} \\ | & | & & | \end{bmatrix},$$

  where $\mathbf{f}_i^1, \dots, \mathbf{f}_i^{N_i}$ are the N_i descriptors extracted from I_i.
- We then put together all descriptors from all training images to form a big training matrix $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_N \end{bmatrix}.$$

  $\mathbf{X}$ is a matrix of size $d \times M$, with $M = \sum_{i=1}^{N} N_i$ and d the dimension of the descriptor.

Clustering/Quantization

- To simplify the notation, we will just write the set of descriptors from the training images as

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{f}_1 & \mathbf{f}_2 & \dots & \mathbf{f}_M \\ | & | & & | \end{bmatrix}.$$

- Create a dictionary by solving the following optimization problem

$$\min_{\mathbf{D}} \sum_{m=1}^{M} \min_{k=1\dots K} ||\mathbf{f}_m - \mathbf{d}_k||^2,$$

  where $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_K]$ are the K clusters centers to be found and $||.||$ is the L_2 norm of vectors.

- **D** is the visual dictionary or codebook.

Clustering/Quantization

- The optimization problem

$$\min_{\mathbf{D}} \sum_{m=1}^{M} \min_{k=1 \ldots K} ||\mathbf{f}_m - \mathbf{d}_k||^2,$$

is solved iteratively with K-means algorithm.

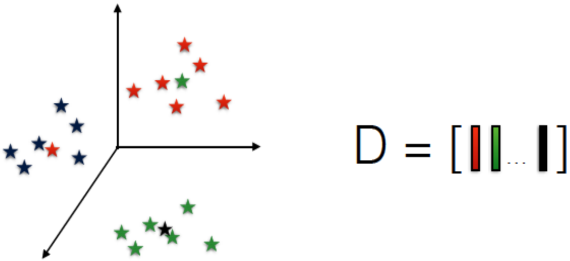
K-means

1. Initialize the K centers (randomly)
2. Assign each data point to one of the K centers
3. Update the centers
4. Iterate until convergence

Clustering/Quantization

- K-means algorithm results in a set of K cluster centers which form the dictionary

$$\mathbf{D} = \begin{bmatrix} | & | & & | \\ \mathbf{d}_1 & \mathbf{d}_2 & \dots & \mathbf{d}_K \\ | & | & & | \end{bmatrix}_{d \times K}$$



BoW representation

Features coding

- Given the dictionary $\mathbf{D}$
- Given a set of low-level features $\mathbf{X}_i$ from image I_i

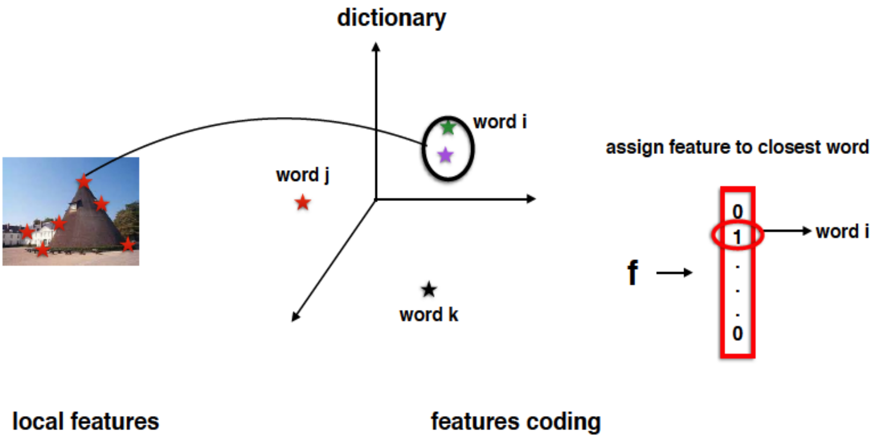
$$\mathbf{X}_i = \begin{bmatrix} | & | & & | \\ \mathbf{f}_i^1 & \mathbf{f}_i^2 & \dots & \mathbf{f}_i^{N_i} \\ | & | & & | \end{bmatrix}$$

- Encode each local descriptor $\mathbf{f}_i^l$ using the dictionary $\mathbf{D}$
  - Find $\mathbf{a}_l$ such that

$$\min_{\mathbf{a}_l} ||\mathbf{f}_i^l - D\mathbf{a}_l||^2 \ s.t. \ ||\mathbf{a}_l||_0 = 1, \mathbf{a}_l \geq 0$$

Features coding

- Encode each local descriptor $\mathbf{f}_i^I$ using the dictionary $\mathbf{D}$



Features pooling

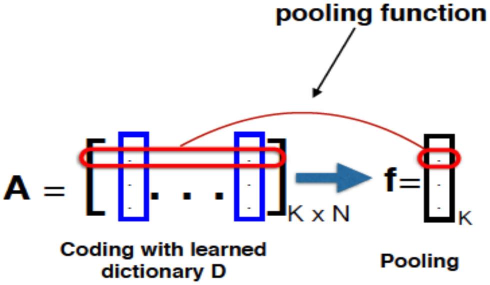
- The coding of image I_i results in a matrix of codes $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_K \\ | & | & & | \end{bmatrix}_{K \times N_i} ,$$

  where each $\mathbf{a}_l$ satisfies $||\mathbf{a}_l||_0 = 1,\ \mathbf{a}_l \geq 0$
- The pooling step transforms $\mathbf{A}$ into a single signature vector $\widehat{\mathbf{x}}_i$

$$\widehat{\mathbf{x}}_i = \text{pooling}(\mathbf{A})$$

Features pooling



- A popular choice for pooling is to compute a histogram

$$\widehat{\mathbf{x}}_i = \frac{1}{N_i} \sum_{l=1}^{N_i} \mathbf{a}_l$$

- The final vector just encodes the frequency of occurrence of each visual words.

Summary : Basic BoW framework

1. Extract a set of local features from all images

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{f}_1 & \mathbf{f}_2 & \dots & \mathbf{f}_M \\ | & | & & | \end{bmatrix}_{d \times M}$$

2. Create a visual dictionary by clustering of the set of local features

$$\mathbf{D} = \begin{bmatrix} | & | & & | \\ \mathbf{d}_1 & \mathbf{d}_2 & \dots & \mathbf{d}_K \\ | & | & & | \end{bmatrix}_{d \times K}$$

3. Given $\mathbf{D}$, encode each local feature from an image I_i, by assigning it to its closest word : $\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_K \\ | & | & & | \end{bmatrix}_{K \times N_i}$

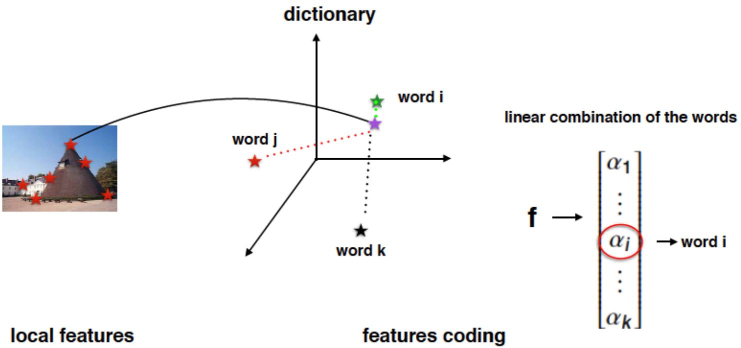
4. Finally, compute the final representation of I_i : $\widehat{\mathbf{x}}_i = \frac{1}{N_i} \sum_{l=1}^{N_i} \mathbf{a}_l$

Outline

Features coding

- Represent each local feature $\mathbf{f}_i^l$ as a linear combination of the words.

$$\mathbf{f}_i^l = \sum_{p=1}^{K} \alpha_i^p \mathbf{d}_p \qquad s.t. \sum_{p=1}^{K} \alpha_i^p = 1,\ \alpha_i^p \geq 0.$$



BoW representation

Features coding

- **Hard assignment**
  - Assign each local feature $\mathbf{f}_i^l$ to its closest word

$$\mathbf{a}_l = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \qquad \sum_p \mathbf{a}_l^p = 1$$

- **Soft assignment**
  - Write each local feature $\mathbf{f}_i^l$ as a linear combination (weighted sum) of the words

$$\mathbf{a}_l = \begin{bmatrix} \alpha_l^1 \\ \vdots \\ \alpha_l^p \\ \vdots \\ \alpha_l^K \end{bmatrix}, \qquad \sum_p \alpha_l^p = 1, \; \alpha_l^p \geq 0.$$

Features pooling

- **average**

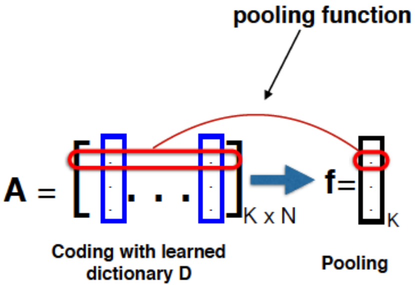
$$\widehat{\mathbf{x}}_i = \frac{1}{N_i} \sum_{l=1}^{N_i} \mathbf{a}_l$$

- **max**

$$\widehat{\mathbf{x}}_i^j = \max_j (\mathbf{a}_l^j)$$

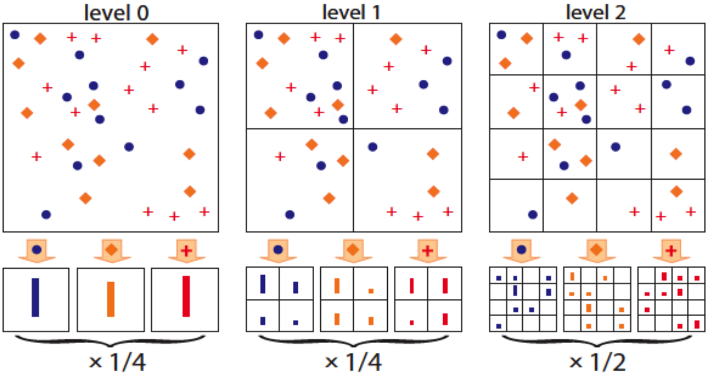
- **mean absolute value**

$$\widehat{\mathbf{x}}_i = \frac{1}{N_i} \sum_{l=1}^{N_i} |\mathbf{a}_l|$$



Including spatial information

- BoW model ignores the spatial layout of the features in the image
- Does not take into account the regularities in image composition



Spatial pyramid : Lazebnik et al. CVPR 2006

Another view

Sparse coding

The objective of sparse coding is to reconstruct an input vector (e.g. an image patch) as a **linear combination** of a **small number of vectors** picked from a large **dictionary**

$$\underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{d}_1 & \mathbf{d}_2 & \dots & \mathbf{d}_K \\ | & | & & | \end{bmatrix}}_{\text{Dictionary}} \begin{bmatrix} \\ \alpha \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \mathbf{x} \\ \\ \end{bmatrix}$$

- Every column of **D** is called an atom
- The vector α is the representation of **x** w.r.t. **D**
- α has few non-zero elements (sparsity)

- Every signal is built as a linear combination of few atoms from **D**

Sparse coding

- Sparse coding can be seen as a soft-assignment
- But, each feature is represented as a linear combination of only a limited number of words.

$$\min_{\alpha} ||\alpha||_0^0 \quad \text{s.t.} \quad ||\mathbf{D}\alpha - \mathbf{x}||_2^2 < \epsilon^2$$

- Solving this optimization problem is hard (NP hard)
  - We approximate it : relaxation or greddy approaches
  - Refer to last year seminar of sparse representations

Dictionary learning

Our goal is to solve

$$\min_{\mathbf{A},\mathbf{D}} \sum_{j=1}^{P} \|\mathbf{D}\alpha_j - \mathbf{x}_j\|_2^2 \quad \text{s.t.} \quad \forall j \ \|\alpha_j\|_0^0 \leq L$$

The K-SVD [1] algorithm is one effective technique for dictionary learning

- It is an unsupervised dictionary learning technique
- It is a generalization of K-means clustering method

1. Aharon, et al., "The K-SVD : An Algorithm for Designing of Overcomplete Dictionaries for Sparse Representation", IEEE Trans. On Signal Processing, 54(11), pp. 4311-4322, 2006.

K-SVD vs K-means

K-means

- Initialize the K centers
- Assign each data point to one of the K centers
- Update the centers
- Iterate

K-SVD

- Initialize the K atoms of $\mathbf{D}$
- Sparse code each example with $\mathbf{D}$
- Update the dictionary $\mathbf{D}$
- Iterate

A word about PCA

- PCA can also be viewed as an unsupervised dictionary learning technique
- Given a set of features $\mathbf{X}$, we find a set of vectors (the dictionary) $\mathbf{V}$ such that the data is un-correlated when represented in $\mathbf{V}$

$$\mathbf{V} = \begin{bmatrix} | & | & & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_K \\ | & | & & | \end{bmatrix}_{d \times K}$$

- In general, $K \ll d$, so that we reduce the dimensionality of the data
- Each feature $\mathbf{f}_i$ is represented by $\mathbf{V}^T \mathbf{f}_i$

What's about PCA

A word about PCA

- PCA finds a set of K vectors such that $K \leq d$
  - When $K < d$, we say that we have an **under-complete** dictionary
  - When $K = d$, we say that we have a **complete** dictionary

- With the BoW approach, we will usually have large dictionaries, $K > d$
  - When $K > d$, we say that we have an **over-complete** dictionary

Conclusions

From a broader perspective

Matrix factorization

Decomposing each input example as a linear combination of basis vectors

$$\mathbf{X} \approx \mathbf{DA}$$

PCA	variance maximization
ICA	non-Gaussianity (kurtosis) maximization
NMF	non-negativity constraints
Sparse coding	sparsity constraints
...	

TABLE : Different approaches

Outline

Conclusions

- The BoW approach is an efficient image representation technique
- It is inspired by ideas from text/documents retrieval community
- Many extensions and improvements have been proposed
  - Including spatial layout : spatial pyramid
- It falls within a more general framework