From Bag of features to Bag of Words

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
4. Image signature computation: pooling
5. Whole recognition pipeline
Bag of Feature (BoF) Model

Image (features)
Bag of Words representation

- BoF
  - Local signatures: not a scalable representation
  - Not a semantic representation

- Model to represent images for categorization: «Bag of Words BoW»
- BoW model computed from BoF (Bag of features)
Bag of Words (BoW) model: basic explication with textual representation and color indexing

Of all the sensory inputs proceeding to the brain from our eyes, for a long time it was thought that nerve, image messages were the dominant one. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes. For a long time it was thought that nerve, image messages were the dominant one. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes. Our perception of the world around us is based essentially on the messages that reach the brain from our eyes.

Comparing 2 docs using visual/color/word occurrences

Slide credit L. Fei-Fei
Bag of Visual Words (BoW)

(Features)

BoW : histogram on visual dictionary

Questions:
1. Which dictionary?
2. How to project the BoF onto the dico
3. How to compute the histogram?
Outline

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
4. Image signature computation: pooling
5. Whole recognition pipeline

3 steps for BoWs
Step 1: dico computation

1. Extraction of local features (pattern/visual words) in images
   • Training dataset in classification
   • Image dataset in retrieval

2. Clustering of feature space

Training set but no labels => UNSUPERVISED Learning
Step 1 : dico computation

- Many algorithms for clustering:
  - K-Means
  - Vectorial Quantization
  - Gaussian Mixture Models
  - ...

![Diagram of dico computation with clustering algorithms examples]
Clustering with K clusters

Input: set of n points \( \{x_j\}_n \) in \( \mathbb{R}^d \)

Goal: find a set of K (K<<n) points \( w=\{w_k\}_K \)

that give a approximation of the n input points, ie. minimizing mean square error \( C(w) \):

\[
C(w) = \sum_{i=1}^{n} \min_k ||x_i - w_k||^2
\]

At k fixed, complexity is \( O(n^{(Kd+1) \log n}) \)

A lot of strategies to approximate the global optimization problem
Clustering with K clusters

\[ C(w) = \sum_{i=1}^{n} \min_{k} \|x_i - w_k\|^2 \]

**K-means Algorithm:**

Init K centers \((c_k)\) by sampling K points \(w_k\) in \(\mathbb{R}^d\)

1. (Re)assign each point \(x_i\) to the cluster \(s_i\) with the center \(w_{s_i}\) so that \(\text{dist}(x_i, w_{s_i})\) is less than \(\text{dist}\) from \(x_i\) to any other clusters

2. Move all \(w_k\) inside each cluster as the new barycenter from all the points assigned to the cluster \(k\) (equ. to minimize the corresponding mean square error)

3. Go to step 1 if some points changed clusters during the last iteration

Output: the set of the final K cluster centers \(\{c_k = w_k\}\)
K-means: why it is successful?

Consider an arbitrary cluster assignment $s_i$.

$$C(w) = \sum_{i=1}^{n} \min_k \|x_i - w_k\|^2$$

$$= \sum_{i=1}^{n} \|x_i - w_{s_i}\|^2 - \sum_{i=1}^{n} \|x_i - w_{s_i}\|^2 - \min_k \|x_i - w_k\|^2$$

$$\mathcal{L}(s,w) - \mathcal{D}(s,w) \geq 0$$

1. Change $s_i$ to minimize $\mathcal{D}$ leaving $C(w)$ unchanged.

2. Change $w_k$ to minimize $\mathcal{L}$. Meanwhile $\mathcal{D}$ can only increase.

© L. Botou
Clustering

• K-means:
  • Pros
    • Simplicity
    • Convergence (local min)
  • Cons
    • Memory-intensive
    • Depending on K
    • Sensitive to initialization
    • Sensitive to artifacts
    • Limited to spherical clusters
    • Concentration of clusters to areas with high densities of points (Alternatives: radial based methods)

• K-Means deeply used in practice
Clustering

• Uniform / K-means / radius-based:

• Radius-based clustering assigns all features within a fixed radius of similarity $r$ to one cluster.
Visual words

Centers = dico. Visual words

Extraction

Clustering

Formation du dico

Dico examples
Plan

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
4. Image signature computation: pooling
5. Whole recognition pipeline
Step 2: BoW image signature

- For each image:
  - For each local feature: find the closest visual word
  - Increase the corresponding bin in histogram

- Image signature (global Index):
  - Vector (histogram)
  - M=K dimension = dico size
  - Each term represents a Likelihood to get this visual word
Projection to dictionary

- Original BoW strategy: **hard assignment/coding**
  - Find the closest cluster for each feature
  - Assign a fix weight (e.g. 1)
Notations:

- Image data: \( X = \{ x_j \in \mathbb{R}^d \} , j \in \{1; N\} \)
- Centers: \( C = \{ C_m \}, \ m \in \{1; M\} \)
- Coding:

\[
  f : \mathbb{R}^d \rightarrow \mathbb{R}^M
  \]

\[
  x_j \rightarrow f(x_j) = \alpha_j = \{ \alpha_{m,j} \} , \quad m \in \{1; M\}
  \]

Hard coding: \( f = f_Q \) assigns a constant weight to its closest center:

\[
  f_Q(x_j)[m] = \begin{cases} 
  1 & \text{if} \ m = \arg\min_{k \in \{1; M\}} \| x_j - c_k \|^2 \\
  0 & \text{otherwise}
\end{cases}
  \]
\[
H = \begin{bmatrix}
  c_1 & \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\
  \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
  \alpha_{M,1} & \alpha_{M,j} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \\
\end{bmatrix}
\]

\[\Rightarrow g: \text{pooling}\]

\[\downarrow \quad f: \text{cooding}\]
Plan

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
4. **Image signature computation: pooling**
5. Whole recognition pipeline
Aggregating projections => global image index

- Global Index: image likelihood to get each visual word
- Several strategies to aggregate the projections: pooling

\[ g : \mathbb{R}^N \rightarrow \mathbb{R} \]

\[ \alpha_m = \{\alpha_{m,j}\}, j \in \{1; N\} \rightarrow g(\alpha_m) = z_m \]
\[ \mathbf{H} = \begin{bmatrix} c_1 & \cdots & c_M \\ \mathbf{\alpha}_1 & \cdots & \mathbf{\alpha}_M \end{bmatrix} \quad \Rightarrow g: \text{pooling} \]

\[ f: \text{cooding} \]
Aggregating projections => global image index

- **Sum pooling**: initial BoW strategy (just counting occurrences of words in the document)

Classical BoW = **hard coding + sum pooling**

1. Find the closest cluster for each feature
2. Assign a fix weight (e.g. 1) to this cluster
Aggregating projections => global image index

- BoW Sum pooling:

\[ z_m = g(\alpha_m) = \sum_{j=1}^{N} \alpha_{m,j} = \sum_{j=1}^{N} f_Q(x_j)[m] \]

\[ z_m = \sum_{j=1}^{N} \begin{cases} 
1 & \text{if } m = \arg\min_{k \in \{1; M\}} \|x_j - c_k\|^2 \\
0 & \text{otherwise}
\end{cases} \]
Plan

1. Introduction to Bag of Words
2. Dictionary computation
3. Coding of local descriptors
4. Image signature computation: pooling
5. **Whole recognition pipeline**
1. feature detection & representation

2. codewords dictionary

3. image representation
Learning and Recognition

1. feature detection & representation

2. codewords dictionary

3. ...

4. category models (and/or) classifiers

category decision