Learning Weights for Codebook in Image Classification and Retrieval

Hongping Cai, Fei Yan and Krystian Mikolajczyk
Centre for Vision, Speech & Signal Processing, University of Surrey, UK
{hongping.cai, f.yan, k.mikolajczyk}@surrey.ac.uk

I. Motivation

WHAT Optimization of the codebook in bag-of-words models with machine learning.
WHY Some codewords can be more informative than others thus more discriminative for classification and retrieval.
HOW Weighted similarity between same labeled images should be larger than that between differently labeled images.

Optimize sub-vectors of \( \mathbf{w} \) in a recursive way: the sub-vector in the white box is the item being optimized with all the remaining sub-vectors in shadow boxes fixed.

\[ s(i_j, l_{i_j}) > s(l_{i_j}, l_k) \]

Ideally, \( s(i_j, l_{i_j}) > s(l_{i_j}, l_k) \)

Normal bag-of-words model: \( s(i_j, l_{i_j}) = \mathbf{T}^T \mathbf{s}_{i_j} \)
Weighted bag-of-words model: \( s(i_j, l_{i_j}) = \mathbf{w}^T \mathbf{s}_{i_j} \)

\( s_{i_j} = (s_{i_j}(1), s_{i_j}(2), ..., s_{i_j}(M))^T \): intersection of the \( m \)-th codeword occurrence frequency

II. Algorithm

Learning weight vector \( \mathbf{w} \in \mathbb{R}^M \) is formulated as a constrained convex quadratic programming:

\[
\begin{align*}
\min_{\mathbf{w}, \xi_{i,j,k}} & \quad \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i,j,k} \xi_{i,j,k} \\
\text{s.t.} & \quad \mathbf{w}^T \mathbf{s}_{i,j} - \mathbf{w}^T \mathbf{s}_{k} \geq 1 - \xi_{i,j,k}, \forall (i,j,k) \in \mathcal{T} \\
& \quad \mathbf{w} \geq 0, \xi_{i,j,k} \geq 0, \forall (i,j,k) \in \mathcal{T}
\end{align*}
\]

\[ \mathcal{T} = \{(i,j,k)| y_i = y_j, y_j \neq y_k \} \]
\[ y_i; \text{the class label for image } I_i \]

III. Implementation

To solve \( \mathbf{w} \) efficiently, two issues have to be addressed:
- How to choose the triplets when the training set \( \mathcal{T} \) is large?
- How to optimize the speed and memory usage for large codebooks?

\[ \mathcal{T}^T = \{(i,j,k)| \mathbf{w}_q^T \mathbf{s}_{i,j} < \mathbf{w}_{q,k}^T \mathbf{s}_{k} \} \]

Optimize sub-vectors of \( \mathbf{w} \) in a recursive way: the sub-vector in the white box is the item being optimized with all the remaining sub-vectors in shadow boxes fixed.

IV. Performance

Multi-class classification on Scene-15

Image retrieval on Oxford5K

V. Learnt weights

VI. Conclusions

- The proposed efficient optimization approach for learning approach can improve NN-based classification and retrieval performance, especially when the training examples are insufficient for a large size codebook.
- The approach can also be used for building compact codebooks or for object localization.