In Defense of Soft-assignment Coding

Lingqiao Liu[1], Lei Wang[1,2], Xinwang Liu[1,3]
[1] School of Engineering, Australian National University, ACT 0200, Australia
[2] School of Computer Science & Software Engineering, University of Wollongong, NSW 2522, Australia
[3] School of Computer, National University of Defense Technology, 410073, ChangSha, Hunan, P.R.China
{lingqiao.liu, xinwang.liu}@cecs.anu.edu.au, leiw@uow.edu.au

Abstract

In object recognition, soft-assignment coding enjoys computational efficiency and conceptual simplicity. However, its classification performance is inferior to the newly developed sparse or local coding schemes. It would be highly desirable if its classification performance could become comparable to the state-of-the-art, leading to a coding scheme which perfectly combines computational efficiency and classification performance. To achieve this, we revisit soft-assignment coding from two key aspects: classification performance and probabilistic interpretation. For the first aspect, we argue that the inferiority of soft-assignment coding is due to its neglect of the underlying manifold structure of local features. To remedy this, we propose a simple modification to localize the soft-assignment coding, which surprisingly achieves comparable or even better performance than existing sparse or local coding schemes while maintaining its computational advantage. For the second aspect, based on our probabilistic interpretation of the soft-assignment coding, we give a probabilistic explanation to the magic max-pooling operation, which has successfully been used by sparse or local coding schemes but still poorly understood. This probability explanation motivates us to develop a new mix-order max-pooling operation which further improves the classification performance of the proposed coding scheme. As experimentally demonstrated, the localized soft-assignment coding achieves the state-of-the-art classification performance with the highest computational efficiency among the existing coding schemes.

1. Introduction

The Bag-of-Feature (BoF) approach [13, 4, 7] has now established itself as the state-of-the-art for generic image classification. It commonly consists of feature extraction, codebook creation, feature coding, and feature pooling. Recent research shows that given a visual codebook, how to code each local feature and how to pool the coding coefficient to obtain an image-level representation have significant impact to classification performance.

The simplest coding in the literature assigns a local feature to the closest visual word, giving one and only one nonzero coefficient. This "hard"-assignment does not consider codeword ambiguity and often introduces large quantization error. In [6, 14], a "soft"-assignment coding is proposed to alleviate this drawback by assigning a local feature to all visual words. The coding coefficient represents the membership of a local feature to different visual words. Recently, sparse and local coding schemes [16, 15, 18, 17] have been shown as a better choice. They optimize a linear combination of few visual words to approximate a local feature and code it with the optimized coefficients.

Given the coding result of each local feature, sum-pooling or average-pooling, which adds up the coefficient of all local features for a visual word, has been commonly used to obtain image-level representation. Recent work indicates that max-pooling that chooses the largest coefficient for a visual word can lead to better classification performance [15, 1, 16, 2]. The work in [2] suggests that having the max-pooling is even more crucial than employing a sparse coding scheme. Currently, the framework of using max-pooling with a sparse or local coding scheme is regarded as the state-of-the-art.

Compared with existing coding schemes, the soft-assignment coding in [6, 14] is conceptually simpler and computationally more efficient. Its coding process can be well understood as evaluating the membership of a local feature to different visual words. Also, it involves no optimization and only needs to compute the distance of a local feature to each word. However, the main drawback is that it cannot give excellent classification performance as the sparse or local coding counterparts.

This paper aims to identify the causes for the inferiority of the soft-assignment coding and improve its performance accordingly. We revisit this coding scheme from two key aspects: classification performance and probabilistic interpretation, and obtain the following interesting results:

978-1-4577-1102-2/11/$26.00 ©2011 IEEE

i) We find that the inferior performance of the soft-assignment coding is probably because it does not take the underlying manifold structure of local features into account. This makes the estimation of the membership to distant visual words unreliable. Rigidly employing the membership to all visual words degrades the classification performance of soft-assignment coding. To verify our analysis and improve this situation, we propose to only consider the k-nearest words in coding a local feature. With max-pooling, this "localized" soft-assignment coding[1] surprisingly catches up with or even outperforms the sparse or local coding schemes while maintaining its computational advantage.

ii) We extend its membership interpretation to show that soft-assignment coding essentially estimates the *posteriori* probability of a local feature to each visual word. This simple extension is critical in that it allows us to give a probabilistic explanation to the magic max-pooling operation in the context of soft-assignment coding. Formally, the largest soft-assignment coding coefficient can be proved as a lower bound of the probability of the presence of a visual word in an image. Using this largest coefficient, max-pooling can be viewed as a robust way to estimate such probability.

iii) Our probabilistic interpretation of soft-assignment coding further motivates us to develop a mix-order max-pooling operation. Besides estimating the probability of the presence of a visual word in an image, this mix-order max-pooling allows us to infer the probability of the "k-times" presence of a visual word in an image. This effectively incorporates the occurrence frequency information that is missed in existing max-pooling, attaining a more informative image-level representation.

We verify the effectiveness of our localized soft-assignment coding on multiple benchmark data sets. As demonstrated, it can achieve comparable or even better classification performance than the state-of-the-art coding schemes, and it has the highest ratio of classification performance to computational efficiency. Also, we test the proposed mix-order max-pooling and show that it can lead to better classification performance than the max-pooling.

2. Related Work

This section reviews commonly used coding and pooling schemes. Let $\mathbf{b}_i$ ($\mathbf{b}_i \in \mathbb{R}^d$) denote a visual word or a basis vector, where d is the dimensionality of a local feature. The total number of visual words is n. A matrix $\mathbf{B}_{d\times n} = (\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n)$ denotes a visual codebook or a set of basis vectors. Let $\mathbf{x}_i$ ($\mathbf{x}_i \in \mathbb{R}^d$) be the ith local feature in an image. Let $\mathbf{u}_i$ ($\mathbf{u}_i \in \mathbb{R}^n$) be the coding coefficient vector of $\mathbf{x}_i$, with u_{ij} being the coefficient with respect to word $\mathbf{b}_j$.

Hard-assignment coding: For a local feature $\mathbf{x}_i$, there is one and only one nonzero coding coefficient. It corresponds to the nearest visual word subject to a predefined distance. When Euclidean distance is used,

$$u_{ij} = \begin{cases} 1 & \text{if } j = \arg \min_{j=1,\cdots,n} \|\mathbf{x}_i - \mathbf{b}_j\|_2^2; \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Soft-assignment coding: The jth coding coefficient represents the degree of membership of a local feature $\mathbf{x}_i$ to the jth visual word,

$$u_{ij} = \frac{\exp(-\beta\|\mathbf{x}_i - \mathbf{b}_j\|_2^2)}{\sum_{k=1}^{n} \exp(-\beta\|\mathbf{x}_i - \mathbf{b}_k\|_2^2)} \tag{2}$$

where β is the smoothing factor controlling the softness of the assignment. Note that all the n visual words are used in computing u_{ij}.

Sparse Coding: It represents a local feature $\mathbf{x}_i$ by a linear combination of a sparse set of basis vectors. The coefficient vector $\mathbf{u}_i$ is obtained by solving an ℓ_1-norm regularized approximation problem,

$$\mathbf{u}_i = \arg \min_{\mathbf{u}\in\mathbb{R}^n} \|\mathbf{x}_i - \mathbf{B}\mathbf{u}\|_2^2 + \lambda\|\mathbf{u}\|_1. \tag{3}$$

Locality-constrained Linear Coding (LLC): Unlike the sparse coding, LLC enforces locality instead of sparsity. This leads to smaller coefficient for the basis vectors farther away from a local feature $\mathbf{x}_i$. The coding coefficient is obtained by solving the following optimization:

$$\begin{aligned} \mathbf{u}_i &= \arg \min_{\mathbf{u}\in\mathbb{R}^n} \|\mathbf{x}_i - \mathbf{B}\mathbf{u}\|_2^2 + \lambda\|\mathbf{d}_i \odot \mathbf{u}\|_2^2 \\ s.t. &\quad \mathbf{1}^\top \mathbf{u}_i = 1, \end{aligned} \tag{4}$$

where $\mathbf{d}_i = \exp(\mathrm{dist}(\mathbf{x}_i, \mathbf{B})/\sigma)$ and $\mathrm{dist}(\mathbf{x}_i, \mathbf{B}) = (\mathrm{dist}(\mathbf{x}_i, \mathbf{b}_1), \mathrm{dist}(\mathbf{x}_i, \mathbf{b}_2), \cdots, \mathrm{dist}(\mathbf{x}_i, \mathbf{b}_n))^\top$ denotes the Euclidean distance between $\mathbf{x}_i$ and each $\mathbf{b}_j$. σ is a parameter controlling the weighting vector $\mathbf{d}_i$. In [15], a smart approximation is proposed to improve its computational efficiency in practice. Ignoring the second term in Eq.(4), it directly selects the k nearest basis vectors of $\mathbf{x}_i$ to minimize the first term by solving a much smaller linear system. This gives the coding coefficient for the selected k basis vectors and other coefficient are simply set to zero.

Given the coding coefficient of all local features in an image, a pooling operation is often used to obtain an image-level representation $\mathbf{p}$, where $\mathbf{p} \in \mathbb{R}^n$.

Sum-pooling / average-pooling: With sum-pooling, the ith component of $\mathbf{p}$ is $p_j = \sum_{i=1}^{l} u_{ij}$, where l is the total number of local features in an image. Dividing p_j by l

[1] We noticed in the literature of image retrieval [11] that a localized scheme has been empirically implemented because the number of codewords is so large that calculating the membership to each codeword is computational prohibitive. In this paper, our work focuses on image classification and proposes this localized scheme with a completely different motivation.

corresponds to average-pooling. Both operations have been widely used. The histogram of number of occurrence of visual words in an image is essentially obtained by applying sum-pooling to hard-assignment coding result.

Max-pooling: The ith component of $\mathbf{p}$ is defined as $p_j = \max_i u_{ij}$, where $i = 1, 2, \cdots, l$. The max-pooling often gives better classification than sum- and average-pooling. Working with hard-assignment coding scheme, max-pooling gives a binary histogram, indicating the presence or absence of each visual word in an image. However, when working with other coding schemes, its mechanism has not been fully understood in the literature.

3. Soft-assignment coding revisit

3.1. Investigating the cause of inferiority

We formally express the coding coefficient of soft-assignment coding as the *posteriori* probability of a local feature $\mathbf{x}_i$ belonging to a visual word $\mathbf{b}_j$ by defining

$$P(\mathbf{b}_j|\mathbf{x}_i) = \frac{1}{Z} \exp\left(s(\mathbf{x}_i, \mathbf{b}_j)\right), \tag{5}$$

where normalization factor Z ensures $\sum_{j=1}^{n} P(\mathbf{b}_j|\mathbf{x}_i) = 1$. The function $s(\mathbf{x}_i, \mathbf{b}_j)$ measures the compatibility of $\mathbf{x}_i$ and $\mathbf{b}_j$. The term $\exp\left(s(\mathbf{x}_i, \mathbf{b}_j)\right)$ indicates the likelihood of $\mathbf{x}_i$ belonging to $\mathbf{b}_j$. In soft-assign coding, $\exp\left(s(\mathbf{x}_i, \mathbf{b}_j)\right)$ is set as $\exp\left(-\beta\|\mathbf{x}_i - \mathbf{b}_j\|_2^2\right)$[6, 14]. This implicitly assumes a spherical Gaussian distribution for the cluster of the ith word, which fits the assumption of the k-means clustering commonly used to generate the clusters. The mean of this Gaussian distribution is $\mathbf{b}_j$ and the covariance matrix is $\sigma^2\mathbf{I}$, where $\sigma^2 = (2\beta)^{-1}$.

The soft-assignment coding uniformly employs a same β value in all Gaussian distributions. This is a reasonable choice because β may not be reliably estimated for every Gaussian distribution, for example, when there are only a small number of samples in a cluster. As the sole parameter in the soft-assignment coding, the value of β becomes critical to the coding and classification performance. It determines the sensitivity of the likelihood to the variation of the distance $\|\mathbf{x}_i - \mathbf{b}_j\|_2^2$. To ensure a sufficiently good β value to be used, the work in [6, 14] tunes it via cross-validation.

Recently developed Locality-constrained Linear Coding (LLC) [15, 19] demonstrates excellent coding and classification performance. A fundamental assumption of LLC is that local features approximately reside on a lower-dimensional manifold in an ambient descriptor space. The soundness of the assumption has been supported by the success of LLC. The presence of a manifold structure implies that the Euclidean distance is only meaningful within a local region, in which it can approximate a geodesic distance well. Out of this region, two local features measured close by the Euclidean distance might be actually far from each other. This implies that the likelihood of $\mathbf{x}_i$ to $\mathbf{b}_j$ cannot be reliably estimated when $\|\mathbf{x}_i - \mathbf{b}_j\|_2^2$ is larger.

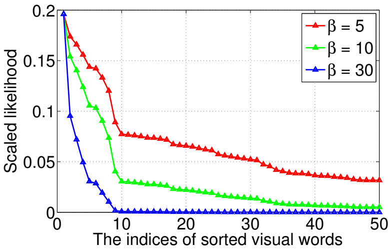


Figure 1: The likelihood of a local feature to a set of visual words when different β values are used. For better illustration, three curves are aligned to have a same largest value.

To avoid using unreliable likelihood, one solution for the soft-assignment coding could be to quickly decrease the likelihood for an increasing $\|\mathbf{x}_i - \mathbf{b}_j\|_2^2$ by carefully tuning β. However, this will encounter the following dilemma. To illustrate it, we plot the likelihood, $\exp\left(-\beta\|\mathbf{x}_i - \mathbf{b}_j\|_2^2\right)$, of $\mathbf{x}_i$ to a set of 1000 visual words by applying different β values. After sorting and aligning them, the largest 50 likelihood are displayed in Figure.1. As seen, to quickly decrease the likelihood, an adequately large β has to be used. However, a large β makes the likelihood sensitive to the variation of $\|\mathbf{x}_i - \mathbf{b}_j\|_2^2$. In Figure.1, the Euclidean distances of the local feature to the first and second visual words only have a difference about 0.02. Nevertheless, applying $\beta = 30$ drastically reduces the likelihood by 50%. After all, unlike the case of image matching, the local features in generic image classification are often similar rather than identical. Such high sensitivity will adversely affect the estimate of likelihood and in turn the coding result. Hence, simply tuning β cannot effectively address the issue.

3.2. "localized" Soft-assignment coding

To solve this problem, we propose to only consider the k visual words in the neighborhood of a local feature and conceptually set its distances to the remaining words as infinity. This strategy produces an "early cut-off" effect, which removes the adverse impact of unreliable longer distances even if a small β is used. In the literature, this strategy has been used in Euclidean embedding [12] and spectral clustering [10] to handle an underlying manifold structure in data, demonstrating excellent performance. Formally, recalling that $\mathbf{x}_i$ denotes a local feature and $\mathbf{b}_j$ denotes the jth visual word, the j coding coefficient of our "localized" soft-assignment coding can be written as follows:

$$
\begin{aligned}
u_{ij} &= \frac{\exp\left(-\beta \hat{d}\left(\mathbf{x}_i, \mathbf{b}_j\right)\right)}{\sum_{l=1}^{n} \exp\left(-\beta \hat{d}\left(\mathbf{x}_i, \mathbf{b}_l\right)\right)}, \\
\hat{d}\left(\mathbf{x}_i, \mathbf{b}_l\right) &= \begin{cases} d\left(\mathbf{x}_i, \mathbf{b}_l\right) & \text{if } \mathbf{b}_l \in \mathcal{N}_k(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases}
\end{aligned} \tag{6}
$$

where $\hat{d}(\mathbf{x}_i, \mathbf{b}_l)$ is the localized version of the original distance $d(\mathbf{x}_i, \mathbf{b}_l)$ and $\mathcal{N}_k(\mathbf{x}_i)$ denotes the k-nearest neighborhood of $\mathbf{x}_i$ defined by the distance $d(\mathbf{x}_i, \mathbf{b}_l)$, where $l = 1, 2, \cdots, n$. In this paper, we follow the original soft-assignment coding scheme [6, 14] to define $d(\mathbf{x}_i, \mathbf{b}_l)$ as squared Euclidean distance but other distances can certainly be used. The β value can be tuned by cross-validation as usual.

Note that this "early cut-off" strategy is very simple to implement. It excellently maintains the computational advantage of soft-assignment coding over the state-of-the-art sparse coding and locality-constrained linear coding. Hence, if this new soft-assignment coding can achieve classification performance comparable to the state-of-the-art ones, it will have the highest ratio of performance to computational load, which is an attractive characteristic for practical applications. We will experimentally demonstrate that it can achieve comparable or even better classification performance.

3.3. Interpretation of max-pooling

Max-pooling has demonstrated higher classification performance than sum-pooling and average-pooling [2]. However, its mechanism has not been fully understood so far, which hinders us from developing new pooling operations to obtain more efficient image-level representation. Based on the probabilistic interpretation of soft-assignment coding, we now give max-pooling a probabilistic explanation.

In hard-assignment coding, max-pooling results in a binary histogram, indicating the absence or presence of a visual word $\mathbf{b}_j$ in an image. Now let $P(\mathbf{b}_j|\mathcal{I})$ denote the probability of the presence of word $\mathbf{b}_j$ in an image $\mathcal{I}$. With the Bag-of-Feature (BoF) model, $\mathcal{I}$ is characterized by the set of all local features in this image as $\mathcal{I} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{|\mathcal{I}|}\}$, where $|\mathcal{I}|$ is the cardinality of $\mathcal{I}$.

As shown in 3.1, we interpret soft-assignment coding as computing the *posteriori* probability of a local feature $\mathbf{x}_i$ belonging to a visual word $\mathbf{b}_j$, denoted by $P(\mathbf{b}_j|\mathbf{x}_i)$. Firstly, let us assume that all the local features in an image $\mathcal{I}$ are independent of each other. In this case, it is not difficult to prove that:

$$
P(\mathbf{b}_j|\mathcal{I}) = 1 - \prod_{i=1}^{|\mathcal{I}|} \left(1 - P(\mathbf{b}_j|\mathbf{x}_i)\right) \geq \max_{i=1,\cdots,|\mathcal{I}|} P(\mathbf{b}_j|\mathbf{x}_i) \tag{7}
$$

That is, in the context of soft-assignment coding, the result of max-pooling is actually a lower bound of the probability of the presence of a visual word in an image. To illustrate this relation, we compare $P(\mathbf{b}_j|\mathcal{I})$ and $\max_i P(\mathbf{b}_j|\mathbf{x}_i)$ by using the soft-assignment coding coefficients from a real image in Figure 2. As seen, $\max_i P(\mathbf{b}_j|\mathbf{x}_i)$ is indeed a lower bound and it is tight for many visual words.

Evidently, the independence assumption is inappropriate when the local features are densely sampled from a lattice of pixels. Due to the Markovian property of an image, spatially close local features are highly correlated. In this case, we can partition image $\mathcal{I}$ into a set of homogeneous regions $\mathcal{R}_1, \cdots, \mathcal{R}_r$, for example, by superpixel technique. Now we assume that the local features sampled from the pixels within a same region are identical and thus have same probability $P(\mathbf{b}_j|\mathbf{x} \in \mathcal{R})$[2]. Thus, we rewrite Eq. (7) as

$$
\begin{aligned}
P(\mathbf{b}_j|\mathcal{I}) &= 1 - \prod_{r=1}^{R} \left(1 - P(\mathbf{b}_j|\mathbf{x} \in \mathcal{R}_r)\right) \\
&\geq \max_{r=1,\cdots,R} P(\mathbf{b}_j|\mathbf{x} \in \mathcal{R}_r) = \max_{i=1,\cdots,|\mathcal{I}|} P(\mathbf{b}_j|\mathbf{x}_i)
\end{aligned} \tag{8}
$$

Now the number of terms in the product reduces from $|\mathcal{I}|$ to R. Noting that all the terms are always between 0 and 1, this means the product becomes larger and then $P(\mathbf{b}_j|\mathcal{I})$ becomes smaller. Because the lower bound $\max_i P(\mathbf{b}_j|\mathbf{x}_i)$ is fixed for a given image, this means that it now becomes even tighter than the case shown in Figure 2. In practice, it is better to use $\max_i P(\mathbf{b}_j|\mathbf{x}_i)$ to approximately estimate $P(\mathbf{b}_j|\mathcal{I})$. The term $1 - \prod_{r=1}^{R} (1 - P(\mathbf{b}_j|\mathbf{x} \in \mathcal{R}_r))$ cannot be reliably obtained because it could be affected by the noise from any one of the R probabilities.

The above analysis also gives another interesting interpretation to max-pooling. We can deem visual words as a bank of "feature detectors" and view a coding process as running these detectors on various locations in an image. The best response of each detector is then recorded by a max-pooling of coding coefficients. As can be expected from this interpretation, the denser the sampled local features, the more reliable the responses. This agrees well with the observation in [2].

3.4. A new "mix-order" max-pooling

As shown above, max-pooling only estimates the probability of the presence of a visual word $\mathbf{b}$ in an image $\mathcal{I}$. It ignores the frequency of the occurrence of the word. In the following, we propose a "mix-order" max-pooling to incorporate this information.

For a given word $\mathbf{b}$, let $\mathcal{T} = \{t_1, t_2, \cdots, t_{\mathcal{I}}\}$ be a set of binary ("1" or "0") indicator showing the presence or absence of this word at each local feature $\mathbf{x}_i$ in an image $\mathcal{I}$. By

[2] Certainly, this assumption may not be true when local features cross regions.

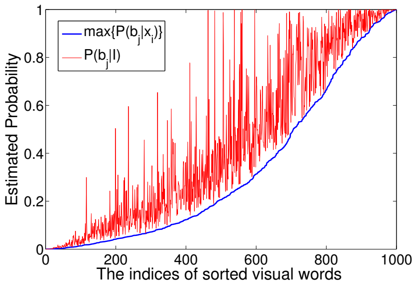


Figure 2: Demonstration of the relationship of $P(\mathbf{b}_j|\mathcal{I})$ to $\max_i P(\mathbf{b}_j|\mathbf{x}_i)$ in Eq. (7) by using the soft-assignment coding coefficients of a real image. For better illustration, we sort $\max_i P(\mathbf{b}_j|\mathbf{x}_i)$ first and use the obtained index to sort $P(\mathbf{b}_j|\mathcal{I})$.

definition, $P(t_i = 1)$ is equivalent to $P(\mathbf{b}|\mathbf{x}_i)$ discussed in last section. Now, we use $P(\#\mathbf{b} \geq k)$ to denote the probability of "word $\mathbf{b}$ is present in an image $\mathcal{I}$ no less than k times". Let $\mathcal{K} = \{t_{i_1}, t_{i_2}, \cdots, t_{i_k}\}$ be a subset of $\mathcal{T}$ and its cardinality $|\mathcal{K}| = k$. In particular, we define $\mathcal{K}^\star$ as a special $\mathcal{K}$ which contains the binary indicators whose $P(t_i = 1)$ values are the k larger ones, where $i = 1, 2, \cdots, |\mathcal{I}|$. With these, we can show that

$$\begin{aligned} P(\#\mathbf{b} \geq k) &= P\left(\sum_{i=1}^{|\mathcal{I}|} t_i \geq k\right) \\ &= \sum_{\mathcal{K} \subset \mathcal{T}} P(t_{i_1} = 1, t_{i_2} = 1, \cdots, t_{i_k} = 1) \\ &= \sum_{\mathcal{K} \subset \mathcal{T}} \prod_{t_{i_j} \in \mathcal{K}} P(t_{i_j} = 1) \geq \max_{\mathcal{K} \subset \mathcal{T}} \left\{ \prod_{t_{i_j} \in \mathcal{K}} P(t_{i_j} = 1) \right\} \\ &= \prod_{t_{i_j} \in \mathcal{K}^\star} P(t_{i_j} = 1) \geq \left(\min_{t_{i_j} \in \mathcal{K}^\star} P(t_{i_j} = 1) \right)^k \\ &= \left(\min_{t_{i_j} \in \mathcal{K}^\star} P(\mathbf{b}|\mathbf{x}_{i_j}) \right)^k . \end{aligned} \tag{9}$$

Thus, similar to the case in Eq. (7), we can now use $(\min_{t_{i_j} \in \mathcal{K}^\star} P(\mathbf{b}|\mathbf{x}_{i_j}))^k$ to approximate $P(\#\mathbf{b} \geq k)$. The term $\min_{t_{i_j} \in \mathcal{K}^\star} P(\mathbf{b}|\mathbf{x}_{i_j})$ can be conveniently obtained by sorting all $P(\mathbf{b}|\mathbf{x}_i)$ values (obtained as the coding coefficient of the soft-assignment coding) and choosing the k larger ones. In practice, to avoid the numerical issue of underflow, we approximate $\sqrt[k]{P(\#\mathbf{b} \geq k)}$ instead and thus do not need to apply the power of k to $\min_{t_{i_j} \in \mathcal{K}^\star} P(\mathbf{b}|\mathbf{x}_{i_j})$. For a given k value, we can produce a new image-level representation. It is an n-dimensional vector indicating the probability of $P(\#\mathbf{b}_j \geq k)$, where $j = 1, 2, \cdots, n$. Max-pooling is a special case when $k = 1$. By varying k, we can obtain a set of vectors, forming a richer description on the word occurrence frequency in an image. These vectors can be concantenated to train a classifier. Also, considering that they may have different discriminative power, a better may be to optimally combine them by using multiple kernel learning.

4. Experimental Result

This experiment aims to verify that i) the "early cut-off" strategy can improve the classification performance of soft-assignment coding; ii) the proposed localized soft-assignment coding can produce comparable or even better classification performance than sparse coding and locality-constrained linear coding (LLC), which are the state-of-the-art; iii) the proposed mix-order max-pooling can achieve better classification performance than max-pooling.

Three benchmark data sets, Scene-15, Caltech-101, and UIUC 8-Sport, are tested. Unless indicated otherwise, all the experimental settings follow the literature to ensure consistency. For Scene-15 and Caltech-101, images are first resized to keep the maximum size of height and width no more than 300 pixels. For UIUC 8-sport, the maximum size is set as 400 because its images have higher resolutions. Dense SIFT features [8] are extracted from all data sets from a single scale of 16×16 patches with the step size of 8 pixels. To incorporate spatial information, the linear version of Spatial Pyramid Matching (SPM) kernel [8, 16] with three levels of 1×1, 2×2 and 4×4 is adopted. To be able to compare with [15], we use k-means clustering to create a visual codebook. The codebook size is set as 1000 for the three data sets. Following the work of sparse coding and LLC, we use a linear SVM. We implement it with the LibSVM toolbox [3]. We randomly split the whole data set into 10 pairs of training/test subsets and the average classification accuracy is reported. To achieve fair and comprehensive comparison, we compare our method to both i) the soft-assignment coding and LLC implemented by ourselves [3] and ii) different versions of the soft-assignment coding and sparse coding reported in the literature. By comparing with our own implementations, we can ensure a same experimental setting, such as image resize ratio, SIFT extraction parameters and training/test splits, to be shared. Meanwhile, comparing with other implementations provides a reference to evaluate the performance of our method. For our proposed localized soft-assignment coding, we fix the neighborhood size as 5 and β as 10 when comparing with other methods. For the standard soft-assignment coding [6, 14], the β is optimally tuned. The impact of the two parameters will be discussed in detail afterwards. Also, max-pooling is used in our implementations.

[3] For LLC, the code of local feature coding is provided by the authors [15].

4.1. Comparison to existing coding schemes

Caltech-101. We first conduct the comparison on the widely used Caltech-101 data set which contains 101 object classes and a background class. All 102 classes are used in this experiment. Following the standard experimental setting, we use 30 images per class for training while leaving the remaining for test. Classification accuracy is compared in Table 1. The table is divided into two sections. The top one lists the proposed localized soft-assignment coding, as well as the soft-assignment coding and the LLC implemented by ourselves. As shown, our method outperforms the original soft assignment coding, indicating the effectiveness of the "early cut-off" strategy. Moreover, it attains higher accurracy than LLC, which is very encouraging because LLC is one of the state-of-the-art coding schemes. The bottom section lists different versions of the existing coding schemes reported in the literature. We list two versions of hard-assignment coding implemented in [8] and [2], respectively. Both of them use SPM, but the former employes a nonlinear SVM while the latter uses a linear SVM. Also, we list two versions of soft-assignment coding, for which sum-pooling is used in [6] while max-pooling is used in [2]. Two versions of sparse coding are listed too. According to the description in [16][2], they share very similar experimental setting, although the reported performance is a bit different. This discrepancy may be due to the implementation of linear SVM or the local minima of the optimization in codebook learning. Comparatively, the implementations in [2, 16] are more comparable to ours. As seen, our localized soft-assignment coding shows better performance than different versions of hard-assignment coding and soft-assignment coding in the bottom section of Table 1. This again validates the effectiveness of our strategy. Moreover, it even outperforms the sparse coding. Actually, merely achieving a performance comparable to sparse coding has made our method more attractive because of its much lower computational overhead. For LLC, its performance is reported as 73.44% in [15]. In that work, the SIFT features are extracted from three scales rather than a single scale in our work. To concretly verify our advantage, we compare our implementation of the localized soft assignment coding and LLC by using SIFT features extracted from three scales. Our localized soft assignment coding now achieves $76.48 \pm 0.71\%$, while LLC only gives $75.28 \pm 0.79\%$.

Scene-15 It contains 4485 images of 15 classes, with class size varying from 200 to 400. Following the standard setting, we use 100 images per class for training and the remaining is reserved for test. Classification accuracy is compared in Table 2, which is again divided into two sections. As seen, our proposed localized soft assignment coding still performs best among the three methods implemented by ourselves. Compared with those reported in the literature, our method still shows higher classification accuracy than the hard-assignment coding and soft-assignment coding. For sparse coding, our result is evidently better than that reported in [16] and comparable to that in [2]. However, our method has clear computational advantage in coding features. Also, it works on the visual codebook created by simple k-mean clustering instead of the more sophisticated dictionary learning used by sparse coding.

Table 1: Comparison on Caltech-101 data set

Algorithm	Classification Accuracy
Localized soft-assignment coding	74.21 ± 0.81
Soft-assignment coding	72.56 ± 0.65
LLC	71.25 ± 0.98
Hard-assignment coding [8]	64.6 ± 0.80
Hard-assignment coding [2]	64.3 ± 0.9
Soft-assignment coding [6]	64.1 ± 1.2
Soft-assignment coding [2]	69.0 ± 0.8
Sparse coding [16]	73.2 ± 0.55
Sparse coding [2]	71.5 ± 1.1
LLC [15]	73.44 ± -

Table 2: Comparison o Scene-15 data set

Algorithms	Classification Accuracy
Localized soft-assignment coding	82.70 ± 0.39
Soft-assignment coding	81.09 ± 0.43
LLC	81.53 ± 0.65
Hard-assignment coding [8]	81.4 ± 0.50
Hard-assignment coding [2]	80.1 ± 0.6
Soft-assignment coding [6]	76.67 ± 0.39
Soft-assignment coding [2]	81.4 ± 0.6
Sparse coding [16]	80.28 ± 0.93
Sparse coding [2]	83.1 ± 0.6

UIUC 8-Sport This data set is collected in [9] for image-based event classification. It contains 8 sport categories of badminton, bocce, croquet, polo, rock climbing, rowing, sailing and snow boarding. There are 1792 images in total, and the size of each category ranges from 137 to 250. Following the common experimental setting on this data set, we randomly select 70 training images and 60 test images from each class. Classification accuracy is compared in Table 3. Again, the proposed localized soft-assignment coding shows better performance than the original soft-assignment coding and the LLC. Its classification accuracy is also comparable to that obtained by using the sparse coding, as reported in [5]. For the UIUC 8-Sport data set, we notice that original soft-assignment coding has been able to achieve very good performance. Our localized version still slightly

Table 3: Comparison on UIUC 8-Sport data set

Algorithms	Classification Accuracy
Local Soft Assignment	82.29 ± 1.84
Soft-assignment coding	82.04 ± 2.37
LLC	81.41 ± 1.84
Sparse Coding [5]	82.74 ± 1.46

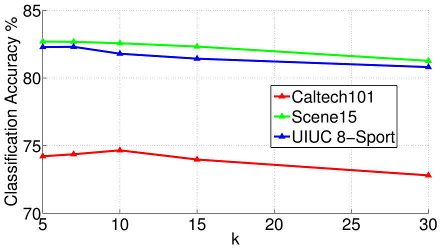


Figure 4: The impact of the neighborhood size on the classification performance for localized soft-assignment coding.

improves the classification accuracy and well reduces its variance to be comparable to the other methods.

4.2. Impact of algorithmic parameters

In this section we investigate the impact of β on the original soft-assignment coding and the proposed localized variant. Also, we show the impact of the neighborhood size k on the latter. As plotted in Figure 3(a), the original soft-assignment coding attains its highest performance on Caltech101 at $\beta = 30$. Also, it can be seen that a relatively larger β ($\beta > 20$) often gives better classification. This demonstrates that quickly decreasing the likelihood $\exp\left(-\beta\|\mathbf{x}_i - \mathbf{b}_j\|_2^2\right)$ when the Euclidean distance $\|\mathbf{x}_i - \mathbf{b}_j\|_2$ increases is necessary, which is consistent with our analysis in Section 3.1. In the meantime, such necessity hinders the original soft-assignment coding from exploiting the property of smaller β values. In the proposed localized soft-assignment coding, this restriction is removed by the "early cut-off" strategy, allowing it to access smaller β values and achieve higher performance. This experimental result provides a support to our argument about the manifold structure of local features. Similar results can be obtained from sub-figure(b) and (c). Since a smaller β works better in the proposed localized soft-assignment coding, we conduct an interesting experiment—simply setting $\beta = 0$ in our method. This means the likelihood of a local feature will simply be "1" for the k nearest visual words and "0" for the other words. As seen in Figure 3(a)(b), this still produces classification performance better than the best of the original soft-assignment coding. On Caltech101, it is even higher than the accuracy of the LLC ($71.25 \pm 0.98\%$) in Table 1, which is implemented by ourself and shares the same experimental setting. This interesting observation seems to suggest that in some cases local features share a similar membership to the visual words fall into a nearby region.

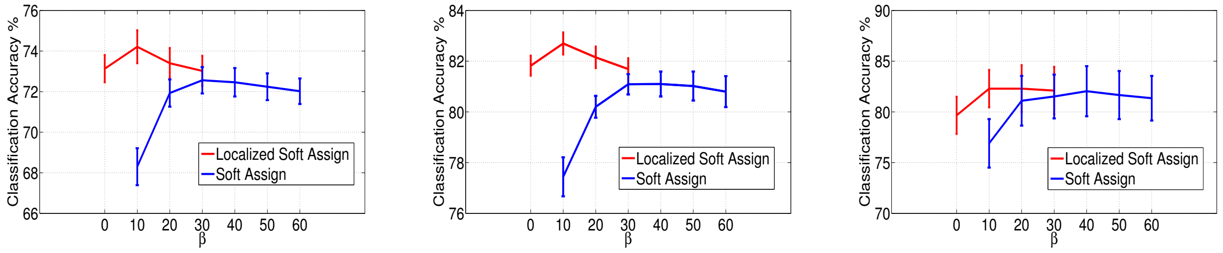
The impact of the neighborhood size to classification accuracy of the localized soft-assignment coding is presented in Figure 4. It is clear that the accuracy gradually decreases with an expanding neighborhood. This again supports our argument that incorporating larger distances out of a local region adversely affects the soft-assignment coding scheme.

4.3. Advantage of the mix-order max-pooling

Finally, we compare the proposed mix-order max-pooling strategy with the commonly used max-pooling. The localized soft-assignment coding is used, but any other coding method can be applied too. As described in Section 3.4, given a k value, mix-order max-pooling produces a vector, in which each component is the kth largest coding coefficient for a visual word. In this experiment, we set k as $1, 2, \cdots, 9$ and obtain 9 vectors. Considering that they may have different discriminative power, we use Multiple Kernel Learning (MKL) technique to combine them for classification. More specifically, we use the linear kernel of each vector as a base kernel and utilize the MKL package in to learn an optimal linear combination of them. Note that the obtained MKL classifier is still a linear SVM, which preserves the computational efficiency in test phase.

We use the whole Scene-15 and UIUC 8-Sport data sets as previous experiments. For Caltech-101 which has 102 classes, multiple kernel learning becomes time-consuming. Here we select the classes whose size is larger than 100, forming a "smaller" Caltech-101[4]. The comparison result is presented in Table 4. As seen, the proposed mix-order max-pooling achieves better classification performance than max-pooling on Scene-15 and UIUC 8-Sport data sets and comparable performance on Caltech11. This shows that incorporating the occurrence frequency of visual words in each image helps classification, at least in classifying scenes and events. Moreover, it validates the effectiveness of our theoretical analysis in Section 3.4, which suggests using the kth largest coding coefficient to more reliably estimate the probability of k-time occurrence. Besides, compared with all the methods listed in Table 2 and 3, with this mix-order strategy, our proposed localized soft-assignment coding actually demonstrates the highest classification performance on Scene-15 and UIUC 8-Sport data sets. Since the mix-order max-pooling is rooted in our probability interpretation of soft-assignment coding, the above result again shows the flexibility and extendability of the soft-assignment coding scheme.

[4] It containing 11 classes of "BACKGROUND_Google", "Faces", "Faces_easy", "Leopards", "Motorbikes", "airplanes", "bonslai", "car_side", "chandelier", "ketch" and "watch".



(a) Caltech101

(b) Scene-15

(c) UIUC 8-Sport

Figure 3: The impact of β on the classification performance for soft-assignment coding and localized soft-assignment coding.

Table 4: Comparison of mix-order max-pooling and max-pooling

Data set	Max-pooling	Mix-order max-pooling
Caltech-101	90.87 ± 1.46 %	90.47 ± 0.46 %
Scene-15	82.70 ± 0.39 %	83.76 ± 0.59 %
UIUC 8-Sport	82.29 ± 1.84 %	84.56 ± 1.5 %

5. Conclusion

A feature coding scheme with both high computational efficiency and excellent classification performance is of great importance for generic image classification, especially at a time when the number of classes and images quickly increases. Also, a high-quality pooling operation is critical to form an image-level representation to facilitate classifier design. This paper addresses both issues by revisiting soft-assignment coding scheme. We demonstrate that our proposed localized soft-assignment coding scheme not only has high computational efficiency but also produces excellent classification performance, being very attractive and competitive for practical applications. Moreover, we discuss the probabilistic essence of max-pooling and further develop a mix-order max-pooling strategy. With such strategy, our localized soft-assignment coding scheme achieves top classification performance on benchmark data sets.

References

[1] Y. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in vision algorithms. In *ICML'10*, 2010. 1

[2] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. *CVPR 2010*, 0. 1, 4, 6

[3] C.-C. Chang and C.-J. Lin. *LIBSVM:*. http://www.csie.ntu.edu.tw/~cjlin/libsvm. 5

[4] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV*, 2004. 1

[5] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely : Laplacian sparse coding for image classification. *CVPR*, 2010. 6, 7

[6] J. C. V. Gemert, J. mark Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *ECCV 2008*, pages 696–709, 2008. 1, 3, 4, 5, 6

[7] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV 2005*, 2005. 1

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2:2169–2178, 2006. 5, 6

[9] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. *CVPR*, 2007. 6

[10] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, 2001. 3

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 2

[12] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000. 3

[13] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, Oct. 2003. 1

[14] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual Word Ambiguity. *IEEE TPAMI*, 99, 2009. 1, 3, 4, 5

[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *CVPR 2010*. 1, 2, 3, 5, 6

[16] J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*. IEEE, 2009. 1, 5, 6

[17] J. Yang, K. Yu, and T. Huang. Supervised translation-invariant sparse coding. *CVPR*, 0. 1

[18] J. Yang, K. Yu, and T. Huang. Efficient highly over-complete sparse coding using a mixture model. In *ECCV 2010*, 2010. 1

[19] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *NIPS*, 2009. 3