Appearing in IEEE Conf. Comp. Vis. Pattern Recogn. 2011. This reprint differs from the original in pagination and typographic detail.

A Generalized Probabilistic Framework for Compact Codebook Creation

Lingqiao Liu[1], Lei Wang[1,2], Chunhua Shen[3,4]
[1] School of Engineering, Australian National University, ACT 0200, Australia
[2] School of Computer Science & Software Engineering, University of Wollongong, NSW 2522, Australia
[3] Australian Center Visual Technologies, University of Adelaide, SA 5005, Australia
[4] NICTA,* Canberra Research Laboratory, ACT 2061, Australia

Abstract

Compact and discriminative visual codebooks are preferred in many visual recognition tasks. In the literature, a few researchers have taken the approach of hierarchically merging visual words of a initial large-size codebook, but implemented this idea with different merging criteria. In this work, we show that by defining different class-conditional distribution functions and parameter estimation methods, these merging criteria can be unified under a single probabilistic framework. More importantly, by adopting new distribution functions and/or parameter estimation methods, we can generalize this framework to produce a spectrum of novel merging criteria. Two of them are particularly focused in this work. For one criterion, we adopt the multinomial distribution to model each object class, and for the other criterion we propose a large-margin based parameter estimation method. Both theoretical analysis and experimental study demonstrate the superior performance of the two new merging criteria and the general applicability of our probabilistic framework.

1. Introduction

In the past few years, the Bag-of-Word (BOW) model has gained its popularity in visual recognition thanks to its simplicity and efficiency [4, 7, 8, 18]. It normally works as follows: A set of local patches (for still images) or local spatial-temporal volumes (for videos) are extracted and represented by local descriptors. These descriptors are processed, for example, by k-means clustering [4], to form a collection of visual words, which in turn forms a visual codebook. By assigning each local descriptor to the closest visual word, a histogram indicating the number of occurrence of each visual word is created to characterize an image or video. Among all the factors of this model, the codebook size plays a pivotal role, determining how well a histogram approximates the true distribution of local descriptors in an image or video. Usually, a sufficiently large-size codebook (for example, up to thousands of visual words) has to be used to ensure good approximation and satisfactory recognition performance.

However, a large-size codebook can be unfavorable. In object localization, the computational load and memory requirement for obtaining the histogram of each candidate window is proportional to the codebook size [1]. In action recognition [10], pair-wise relationship among visual words is informative for modeling actions. A large codebook will quadratically increase the number of pairs to be considered. For classifiers commonly used in visual recognition, such as Support Vector Machines (SVMs) and Nearest Neighbor classifier, their training and test time often increases with the codebook size [23]. Besides, high dimensionality generally results in the "curse of dimensionality" problem. Hence, a compact visual codebook is preferred in many visual recognition tasks. However, simply reducing the value of k in k-means clustering will degrade recognition performance because this loses discriminative information.

In the literature, a variety of approaches have been proposed for designing better visual codebooks [9, 14, 22]. For building both compact and discriminative codebooks in a supervised setting, a dominating approach is to hierarchically merge visual words of a large initial codebook while minimizing the loss of discriminative information, as taken by [1, 23, 24]. Note that these three papers implemented this approach with different models and criteria. In [1, 10], the mutual information between words and class labels is used to identify the optimal pair of words to merge at each level of the hierarchy. In [23], the scatter-matrix-based class separability is used as a criterion to seek the optimal pair of words to merge. The work of [24] differs the previous work in that a more rigorous probabilistic model is used to merge visual words. In their work, the optimal pair is sought as the one after which is merged, the resulting histograms

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Center of Excellence program.

can maximize the posteriori probability of true class labels. Nevertheless, as reported in [1, 23], the merging criterion of [24] often produces results inferior to those in [1, 23]. This is in a sharp contrast to the expected power of a rigorous probabilistic model.

In this work, we follow the basic probabilistic model in [24] and discuss its two key factors: the class-conditional distribution function and parameter estimation method. The difference between our work and [24] is that the two key factors are kept fixed in [24] whereas they are treated as flexible components in our work. As will be seen, such a difference is critical because varying these two factors could bring forth remarkably different characteristic to the probabilistic model. By properly choosing different settings to the two factors, *we achieve a generalized probabilistic framework for merging visual words*. With our framework, we are not only able to unify the criteria in [1, 23, 24], but also able to produce a spectrum of new merging criteria. Two of them will be focused in this paper. In summary, our work has achieved the following results:

- By employing appropriate distribution functions and maximum likelihood estimation, our generalized probabilistic framework reproduces the criteria in [1] and [23] as special cases;
- With this framework, we propose a new criterion by modeling each class with a multinomial distribution function. It achieves better recognition performance than that originally proposed in [24];
- Based on this framework, we put forward a large-margin parameter estimation method, leading to another new criterion. It gives the overall highest recognition performance when compared with all the above word-merging criteria.

2. Related Work

This section reviews supervised compact codebook creation in [1, 23, 24], with the focus on [24] which inspires our work. As shown in [23], compact codebook creation can essentially be casted as a large-scale discrete optimization problem, subject to a criterion related to the discriminative power of the resulting compact codebook. Due to the difficulty of efficient and global optimization, hierarchically merging visual words is often adopted in the literature. That is, two words are identified at each level of the hierarchy such that merging them will optimize a given criterion. Let $\mathcal{B}^{t+1}$ denote a visual codebook consisting of $t+1$ words. Let $\mathcal{B}^t_{r,s}$ be the resulting codebook after merging the rth and sth words. The corresponding histogram for an image or video i is denoted by $\mathbf{h}^t_i$, and its jth bin is h^t_{ij}, where $1 \le i \le n, \;\; 1 \le j \le t$. Also, $c \in \{1, 2, \cdots, C\}$ is the class label of an image or video. In this paper, the criteria in [1, 23, 24] are termed AIB, CSM and UVD in short, respectively.

AIB: In [1], the mutual information, I, between $\mathcal{B}^t_{r,s}$ and class labels is used to measure its discriminative power as

$$I(\mathcal{B}^t_{r,s}, c) = \sum_{j=1}^{t}\sum_{c=1}^{C} P(v^t_j, c) \log \frac{P(v^t_j, c)}{P(v^t_j)P(c)}, \qquad (1)$$

where v^t_j denotes the jth word of $\mathcal{B}^t_{r,s}$ and $P(v^t_j, c)$ and $P(v^t_j)$ are estimated with the jth bins of training histograms. At each level, the words r and s whose merging maximizes $I(\mathcal{B}^t_{r,s}, c)$ are identified and merged. As noted in [1], this criterion can be related to agglomerative information bottleneck [19].

CSM: In [23], the scatter-matrix-based class separability, S, is used to measure the goodness of $\mathcal{B}^t_{r,s}$ as

$$S(r, s) = \mathrm{tr}(\mathbf{S}_w)/\mathrm{tr}(\mathbf{S}_t), \qquad (2)$$

where $\mathbf{S}_w$ and $\mathbf{S}_t$ are the within-class scatter matrix and the total scatter matrix, respectively. They are computed with $\mathbf{h}^t_1, \cdots, \mathbf{h}^t_n$. At each level, the words r and s whose merging minimize $S(r, s)$ are identified and merged [1].

UVD: In [24], the posteriori probability of true class labels conditioned on $\mathcal{B}^t_{r,s}$ is proposed to measure the discriminative power of $\mathcal{B}^t_{r,s}$. Let $\hat{\mathbf{c}} = \{c_1, \cdots, c_n\}$ be the label set of the n training samples. Let $\mathcal{H}^t = \{\mathbf{h}^t_1, \cdots, \mathbf{h}^t_n\}$ be the set of n histograms obtained with $\mathcal{B}^t_{r,s}$. Using Bayes' theorem, this posterior probability is computed as

$$P(\hat{\mathbf{c}}|\mathcal{H}^t) = \frac{P(\mathcal{H}^t|\hat{\mathbf{c}})P(\hat{\mathbf{c}})}{\sum_{\mathbf{c}'} P(\mathcal{H}^t|\mathbf{c}')P(\mathbf{c}')}, \qquad (3)$$

where $P(\mathcal{H}^t|\hat{\mathbf{c}})$ is the likelihood of the n training histograms conditioned on true label configuration $\hat{\mathbf{c}}$, and $P(\mathcal{H}^t|\mathbf{c}')$ is the likelihood conditioned on any one of C^n possible label configurations. Due to the difficulty of enumerating all possible configurations, [24] approximates the denominator with two configurations only: the true configuration $\hat{\mathbf{c}}$ and a special configuration $\mathbf{c}^{\text{same}}$ in which all training samples have a same class label. Assuming equal prior over these two configurations, it gives:

$$P(\hat{\mathbf{c}}|\mathcal{H}^t) \quad \approx \quad \frac{P(\mathcal{H}^t|\hat{\mathbf{c}})}{P(\mathcal{H}^t|\hat{\mathbf{c}}) + P(\mathcal{H}^t|\mathbf{c}^{\text{same}})} \qquad (4)$$

Thus, maximizing $P(\hat{\mathbf{c}}|\mathcal{H}^t)$ is (approximately) equivalent to maximizing $\frac{P(\mathcal{H}^t|\hat{\mathbf{c}})}{P(\mathcal{H}^t|\mathbf{c}^{\text{same}})}$. The likelihood $P(\mathcal{H}^t|\mathbf{c})$ is computed as

$$P(\mathcal{H}^t|\mathbf{c}) = \prod_{c=1}^{C} \int \prod_{\mathbf{h}^t_i \in \mathcal{D}_c} P(\mathbf{h}^t_i|\boldsymbol{\theta}_c)P(\boldsymbol{\theta}_c)d\boldsymbol{\theta}_c \qquad (5)$$

[1] To facilitate the subsequent analysis, we use the minimization of $\mathrm{tr}(\mathbf{S}_w)/\mathrm{tr}(\mathbf{S}_t)$ here. Because of $\mathrm{tr}(\mathbf{S}_t) = \mathrm{tr}(\mathbf{S}_b) + \mathrm{tr}(\mathbf{S}_w)$, it is essentially equivalent to [23] which maximizes $\mathrm{tr}(\mathbf{S}_b)/\mathrm{tr}(\mathbf{S}_t)$.

where $P(\mathbf{h}_i^t|\boldsymbol{\theta}_c)$ is the class-conditional distribution for class c, $\boldsymbol{\theta}_c$ its parameter set, and $\mathcal{D}_c$ the set of all training samples in class c. In [24], $P(\mathbf{h}_i^t|\boldsymbol{\theta}_c)$ is modeled as a Gaussian distribution[2]. A conjugate Gaussian-gamma prior is defined over $\boldsymbol{\theta}_c$ as $P(\boldsymbol{\theta}_c|\mu, \lambda, a, b)$, where μ, λ, a, and b are the hyper-parameters. Assuming the independence of different bins and i.i.d samples in each class, the above likelihood is obtained as

$$P(\mathcal{H}^t|\mathbf{c}) = \prod_{c=1}^{C}\prod_{j=1}^{t}\int \prod_{\mathbf{h}_i^t\in\mathcal{D}_c} P(h_{ij}^t|\theta_{cj})P(\theta_{cj})d\theta_{cj} \qquad (6)$$

where h_{ij}^t is the jth bin of the histogram $\mathbf{h}_i^t$, and θ_{ci} is the parameter set (mean and variance) for the jth bin in class c. Since $P(\theta_{cj})$ is the conjugate prior of $P(h_{ij}^t|\theta_{cj})$, the integral can be analytically worked out. At each level of the hierarchy, the pair of words r and s whose merging maximizes $P(\mathcal{H}^t|\hat{\mathbf{c}})/P(\mathcal{H}^t|\mathbf{c}^{\text{same}})$ is identified and merged.

3. Our Generalized Probabilistic Framework

In this paper, we take the basic formulation in Eq. (4) and develop it to a general framework (Note that we define $\mathcal{J} = \log P(\mathcal{H}^t|\hat{\mathbf{c}})/P(\mathcal{H}^t|\mathbf{c}^{\text{same}})$ and use it throughout the following sections). Any algorithm taking such a formulation needs to determine *two key factors: i) how to model the class-conditional distribution* $P(\mathbf{h}_i|\boldsymbol{\theta}_c)$ [3]*; ii) how to handle the model parameter* $\boldsymbol{\theta}_c$. As shown in Section 2, UVD [24] models $P(\mathbf{h}_i|\boldsymbol{\theta}_c)$ with a Gaussian distribution and take the Bayesian method to marginalize out the model parameter $\boldsymbol{\theta}_c$. The effect of $\boldsymbol{\theta}_c$ is averaged with a Gaussian-gamma prior and their values are not explicitly estimated.

By choosing different settings to the two factors, our framework not only accommodates the existing criteria, but also produces a matrix of new criteria. UVD [24], AIB [23] and CSM [23] are merely three entries corresponding to specific settings of the two factors, and there are more criteria to be explored. Two of them, called MLT and MME in short, are focused in this paper. We demonstrate that they can create more efficient compact codebooks. In Section 3.1, we first propose the criterion MLT, which replaces the Gaussian distribution in UVD with a multinomial distribution. Then, we derive AIB and CSM as two special cases in Section 3.2 and 3.3 respectively. Finally, we propose in Section 3.4 another criterion MME which estimates model parameters through a discriminative approach.

3.1. MLT: Multinomial distribution + Bayesian Method (Dirichlet prior)

In the literature, the BOW model originates from document analysis, in which a histogram of words is usually modeled by a multinomial distribution [2]. In the first criterion derived from our generalized framework, we propose to use the multinomial distribution and Dirichlet prior to replace the Gaussian distribution and the Gaussian-gamma prior in [24]. As will be seen in the experiment, this simple change can bring forth significant improvement.

In MLT, $P(\mathcal{H}|\mathbf{c})$ is still modeled as Eq.(5), but the likelihood and the prior terms become:

$$\begin{aligned} P(\mathbf{h}_i|\boldsymbol{\theta}_c) &= \prod_{j=1}^{t} P(v_j|c)^{h_{ij}} \\ P(\boldsymbol{\theta}_c) &= \frac{1}{B(\boldsymbol{\alpha})}\prod_{j=1}^{t} P(v_j|c)^{\alpha_j-1}, \end{aligned} \qquad (7)$$

where v_j denotes the jth word and $P(v_j|c)$ is the model parameter, which represents the likelihood of word v_j occurring in class c. $B(\boldsymbol{\alpha})$ is the multinomial Beta function and $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_t)$ is the hyper-parameter. We set $\alpha_j = 0.1$ for all j and all classes in the experiment. Substituting (7) into (5), We can obtain:

$$P(\mathcal{H}|\mathbf{c}) = \cdots = \prod_{c=1}^{C}\frac{B(\boldsymbol{\alpha}+\bar{\mathbf{h}}_c)}{B(\boldsymbol{\alpha})} \qquad (8)$$

where we define $\bar{\mathbf{h}}_c = (\bar{h}_{c1}, ..., \bar{h}_{ct})$ for class c and $\bar{h}_{cj} = \sum_{\{i|\mathbf{h}_i\in\mathcal{D}_c\}} h_{ij}$. Note that the integral in (5) can be analytically worked out because the Dirichlet distribution is a conjugate prior of a multinomial distribution. Thus, our MLT criterion is obtained as

$$\mathcal{J} = \sum_{c=1}^{C}\log B(\boldsymbol{\alpha}+\bar{\mathbf{h}}_c) - \log B(\boldsymbol{\alpha}+\sum_{c=1}^{C}\bar{\mathbf{h}}_c) + const. \qquad (9)$$

Recall that $\mathcal{J} = \log P(\mathcal{H}|\hat{\mathbf{c}})/P(\mathcal{H}|\mathbf{c}^{\text{same}})$. At each level of the hierarchy, the pair of words r and s whose merging maximizes $\mathcal{J}$ is identified and merged.

3.2. AIB [1]: Multinomial distribution + Maximum Likelihood Estimation

The above UVD and MLT use Bayesian method to handle the model parameters. The performance of the Bayesian method highly depends on the choice of prior distribution and its hyper-parameters. In practice, for the feasibility of calculation, the hyper-parameters are often empirically set, say, using the same hyper-parameters for all classes. Consequently, the Bayesian method does not necessarily outperform the way that straightforwardly estimates model parameters from training data. In the following, we maintain the multinomial distribution and use the maximum likelihood estimate (MLE). By doing so, our probabilistic framework will produce the AIB criterion in [1].

In this setting, $P(\mathcal{H}|\mathbf{c})$ is computed as:

$$P(\mathcal{H}|\mathbf{c}) = \prod_{\{i|\mathbf{h}_i\in\mathcal{D}_c\}} P(\mathbf{h}_i|\boldsymbol{\theta}_c) = \prod_{\{i|\mathbf{h}_i\in\mathcal{D}_c\}}\prod_{j=1}^{t} P(v_j|c)^{h_{ij}}. \quad (10)$$

[2] As advised in [24], the square root of each bin of $\mathbf{h}$ is used to better fit the Gaussian distribution assumption.

[3] In this section, we drop the superscript t in $\mathbf{h}_i^t$. All the calculation is at the level t unless indicated otherwise.

Thus, the merging criterion becomes:

$$\begin{aligned}\mathcal{J} &= \sum_{c=1}^{C}\sum_{j=1}^{t}\bar{h}_{cj}\log P(v_j|c) - \sum_{j=1}^{t}\left(\sum_{c=1}^{C}\bar{h}_{cj}\right)\log P(v_j)\\ &= \sum_{c=1}^{C}\sum_{j=1}^{t}\bar{h}_{cj}\log\frac{P(v_j|c)}{P(v_j)}.\end{aligned}\tag{11}$$

With training samples, it is not difficult to obtain the MLE of the model parameters as

$$P(v_j|c) = \frac{\bar{h}_{cj}}{\sum_{j=1}^{t}\bar{h}_{cj}}, \quad P(v_j|\mathbf{c}^{\text{same}}) = \frac{\sum_{c=1}^{C}\bar{h}_{cj}}{\sum_{j=1}^{t}\sum_{c=1}^{C}\bar{h}_{cj}}. \tag{12}$$

Note that $P(v_j|\mathbf{c}^{\text{same}}) = P(v_j)$ because all samples are assumed to be in a same class in the $\mathbf{c}^{\text{same}}$ configuration. In AIB [1], these two terms are computed in the same way [4]. Moreover, AIB computes the joint probability as:

$$P(v_j, c) = \frac{\bar{h}_{cj}}{\sum_{j=1}^{t}\sum_{c=1}^{C}\bar{h}_{cj}}. \tag{13}$$

Note that the denominator $\sum_{j=1}^{t}\sum_{c=1}^{C}\bar{h}_{cj}$ keeps constant when merging different words at level t. Substitute $\bar{h}_{cj} = P(v_j, c)\sum_{j=1}^{t}\sum_{c=1}^{C}\bar{h}_{cj}$ into Eq.(11) and dropping constant $\sum_{j=1}^{t}\sum_{c=1}^{C}\bar{h}_{cj}$, we produce AIB criterion in [1] as,

$$\text{Eq.}(11) \propto \sum_{c=1}^{C}\sum_{j=1}^{t}P(v_j, c)\log\frac{P(v_j, c)}{P(v_j)P(c)} = \text{AIB}. \tag{14}$$

3.3. CSM [23]: Gaussian distribution + Maximum Likelihood Estimation

If we use a Gaussian distribution to model $P(\mathbf{h}_i|\boldsymbol{\theta}_c)$ and estimate the mean with MLE, Eq.(4) will lead to the CSM criterion in [23], as shown below.

$$\begin{aligned}P(\mathcal{H}|\mathbf{c}) &= \prod_{\{i|\mathbf{h}_i\in\mathcal{D}_c\}} P(\mathbf{h}_i|\boldsymbol{\theta}_c)\\ \propto |\boldsymbol{\Sigma}_c|^{-\frac{N_c}{2}} &\exp\left(-\frac{1}{2}\sum_{\{i|\mathbf{h}_i\in\mathcal{D}_c\}}(\mathbf{h}_i-\boldsymbol{\mu}_c)^{\top}\boldsymbol{\Sigma}_c^{-1}(\mathbf{h}_i-\boldsymbol{\mu}_c)\right),\end{aligned}\tag{15}$$

where $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ denote the mean and covariance matrix for class c, respectively. N_c is the number of samples in class c. Then $\mathcal{J} = \log P(\mathcal{H}|\hat{\mathbf{c}})/P(\mathcal{H}|\mathbf{c}^{\text{same}})$ becomes:

$$\begin{aligned}\mathcal{J} &= \text{const.} + \sum_{i=1}^{n}(\mathbf{h}_i-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\mathbf{h}_i-\boldsymbol{\mu})\\ &\quad - \sum_{c=1}^{C}\sum_{\{i|\mathbf{h}_i\in\mathcal{D}_c\}}(\mathbf{h}_i-\boldsymbol{\mu}_c)^{\top}\boldsymbol{\Sigma}_c^{-1}(\mathbf{h}_i-\boldsymbol{\mu}_c),\end{aligned}\tag{16}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the total mean and covariance matrix for all data, respectively. Now, we treat $\boldsymbol{\Sigma}_c$ and $\boldsymbol{\Sigma}$ as predetermined constants. Considering a special case of $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \ldots = \boldsymbol{\Sigma}_C = \text{diag}(\sigma_1^2, .., \sigma_1^2)$ and $\boldsymbol{\Sigma} = \text{diag}(\sigma_0^2, .., \sigma_0^2)$, Eq.(16) reduces to

$$\begin{aligned}\mathcal{J} &= \frac{1}{\sigma_0^2}\sum_{i=1}^{n}\|\mathbf{h}_i-\boldsymbol{\mu}\|^2 - \frac{1}{\sigma_1^2}\sum_{c=1}^{C}\sum_{\{i|\mathbf{h}_i\in\mathcal{D}_c\}}\|\mathbf{h}_i-\boldsymbol{\mu}_c\|^2\\ &\propto -\left([\text{tr}(\mathbf{S}_w)] - (\sigma_1^2/\sigma_0^2)\,[\text{tr}(\mathbf{S}_t)]\right)\end{aligned}\tag{17}$$

where $\mathbf{S}_w$ and $\mathbf{S}_t$ are exactly the within-class scatter matrix and the total scatter matrix defined in [23]. The criterion $\text{tr}(\mathbf{S}_w) - (\sigma_1^2/\sigma_0^2)\text{tr}(\mathbf{S}_t)$ strongly connects to $\text{tr}(\mathbf{S}_w)/\text{tr}(\mathbf{S}_t)$ used in [23]. This is because minimizing $\text{tr}(\mathbf{S}_w)/\text{tr}(\mathbf{S}_t)$, which is a fractional programming problem, can be effectively solved by the Dinkelbach's algorithm [17]. It iteratively minimizing $\text{tr}(\mathbf{S}_w) - \lambda\text{tr}(\mathbf{S}_t)$, where λ is the ratio of $\text{tr}(\mathbf{S}_w)$ to $\text{tr}(\mathbf{S}_t)$ at the last iteration.

3.4. MME: Multinomial distribution + Max-Margin Parameter Estimation

The maximum likelihood estimation of model parameters still presents some potential drawbacks. Due to its generative nature, it prevents us from using more information in training data. For example, when the multinomial distribution is employed, the MLE of its parameters are only determined by the average histogram per class. In the literature, this phenomenon is known as *exchangeable property* [2]. One solution may be to adopt more complex distribution, say, the multivariate Polya distribution [13]. However, this will lead to intractable computation because there is normally no analytical MLE for the parameters in these complex models. Another disadvantage of MLE is that the estimation can be noisy when training samples are scarce or many less discriminative visual words exist. It cannot effectively identify the discriminative words since the parameters are estimated based on the data from a same class only. This limits the performance of the created compact codebooks.

To improve this situation, we propose to employ a Max-Margin parameter Estimation (MME) scheme. The idea is to seek the model parameters that can *maximize the margin of posterior probability ratio of the true class label to all other possible labels* under certain regularization. The aforementioned disadvantages of MLE can be avoided because (i) the parameter estimation is now based on all available training samples; (ii) the max-margin principle emphasizes discriminative features. Still modeling $P(\mathbf{h}_i|\boldsymbol{\theta}_c)$ by a multinomial distribution, we define the ratio for each training sample as:

$$\begin{aligned}R_{i,c} &= \log\frac{P(c_i|\mathbf{h}_i)}{P(c|\mathbf{h}_i)} = \log\frac{P(\mathbf{h}_i|c_i)P(c_i)}{P(\mathbf{h}_i|c)P(c)}\\ &= \sum_{j=1}^{t}h_{ij}\log\frac{P(v_j|c_i)}{P(v_j|c)} + \log\frac{P(c_i)}{P(c)}, \; c \neq c_i;\end{aligned}\tag{18}$$

[4] This can be seen in the code provided by the author [20].

where c_i is the truth label of sample i and c is one of the other possible labels. Note that this ratio takes a form of linear classifier if we treat $\log \frac{P(v_j|c_i)}{P(v_j|c)}$ and $\log \frac{P(c_i)}{P(c)}$ as parameters, although the variables are $P(v_j|c_k)$ and $P(c_k)$, $k = 1,..,C$, $j = 1,..,t$. Inspired by the margin definition in SVM [6], we define the margin of posterior probability ratio as:

$$\gamma = \min_{\substack{i=1,\cdots,n\\c=1,\cdots,C}} \frac{R_{i,c}}{\sum_{c_p,c_q}\left[\sum_{j=1}^{t}\left(\log\frac{P(v_j|c_p)}{P(v_j|c_q)}\right)^2+\left(\log\frac{P(c_p)}{P(c_q)}\right)^2\right]}$$
$$= \frac{\min_{\substack{i=1,\cdots,n\\c=1,\cdots,C}} R_{i,c}}{\sum_{c_p,c_q}\left[\sum_{j=1}^{t}\left(\log\frac{P(v_j|c_p)}{P(v_j|c_q)}\right)^2+\left(\log\frac{P(c_p)}{P(c_q)}\right)^2\right]}$$
$$\forall\, p \neq q,\;\; p,q = 1,2,\cdots,C; \tag{19}$$

where the denominator acts as a regularization term which smoothes the estimation of $P(v_j|c_k)$ and $P(c_k)$ over different k ($k = 1,\cdots,C$). Maximizing the margin leads to an optimization problem which is similar to SVM [5](Note that we also add the slack variables to handle the non-separable case):

$$\begin{aligned}
\min\;\; & \sum_{c_p,c_q}\left[\sum_{j=1}^{t}\left(\log\frac{P(v_j|c_p)}{P(v_j|c_q)}\right)^2+\left(\log\frac{P(c_p)}{P(c_q)}\right)^2\right]+\lambda\sum_{i,c}\xi_{i,c}\\
\text{s.t.}\;\; & \sum_{j=1}^{t} h_{ij}\left(\log\frac{P(v_j|c_i)}{P(v_j|c)}\right)+\left(\log\frac{P(c_i)}{P(c)}\right)\geq 1-\xi_{i,c},\\
& \xi_{i,c}\geq 0,\;\; \forall\, c\neq c_i;\;\; \forall\, i=1,2,\cdots,n;\\
& \forall\, p\neq q,\;\; p,q,c=1,2,\cdots,C;
\end{aligned} \tag{20}$$

Besides, we need to ensure the variables $P(v_j|c_k)$ and $P(c_k)$ bounded because only their pairwise ratio presents in the constraints and object function. Hence, we impose two more constraints through the total probability rule and probability property,

$$\sum_{c=1}^{C} P(v_j|c)P(c) = P(v_j), \quad \sum_{c=1}^{C} P(c) = 1. \tag{21}$$

where $P(v_j)$ is estimated via Eq.(12). Thus the parameter estimation can be performed in two steps: i) obtain the log-ratios by solving the problem in Eq.(20) which is a QP problem; and ii) recover the exact values of $P(v_j|c_k)$ and $P(c_k)$ combining a linear equation derived from Eq.(21). Particularly, in binary classification, there are only two types of class labels, $+1$ and -1. The problem in Eq.(20) turns out to be a standard QP problem in binary SVMs by defining

$$w^j_{+1,-1} = \log\frac{P(v_j|c=1)}{P(v_j|c=-1)};\;\; b_{+1,-1} = \log\frac{P(c=1)}{P(c=-1)}. \tag{22}$$

This equivalence provides us with the advantage of simply using the off-the-shelf SVM solver to perform parameter estimation. Once we estimate these model parameters, we apply them to the multinomial distribution to compute $\mathcal{J}$ to identify the optimal pair of words to merge.

Estimating $P(v_{rs}|c)$ might be a computational issue, where v_{rs} denotes the new visual word formed by merging words r and s. If strictly following the max-margin parameter estimation, we have to re-estimate $P(v_{rs}|c)$ by solving the QP problem for each possible pair of r and s, which is at the order of $\mathcal{O}(t^2)$ at level t. Even the SVM solver is highly efficient, this will still be too time-consuming. In practice, we adopt a compromised scheme: the max-margin estimation is only carried out once at each level after the optimal pair of words is identified. In the course of identifying the optimal pair, the updating formula $P(v_{rs}|c) = P(v_r|c) + P(v_s|c)$ is used. Experimental study shows that this strategy works well in practice.

Multi-class extension For a multi-class case, the optimization in Eq.(20) is not equivalent to the optimization problem in SVM anymore because many extra constraints are to be introduced. Extending the definition of w in Eq.(22) to define $w^j_{c_p,c_q} = \log\frac{P(v_j|c_p)}{P(v_j|c_q)}$, a set of extra constraints: $w^j_{c_p,c_q} = w^j_{c_p,c_k} + w^j_{c_k,c_q}$, $w^j_{c_p,c_q} + w^j_{c_q,c_p} = 0$ need to be added to enforce the fact that $\log\frac{P(v_j|c_p)}{P(v_j|c_q)} = \log\frac{P(v_j|c_p)}{P(v_j|c_k)} + \log\frac{P(v_j|c_k)}{P(v_j|c_q)}$ $\log\frac{P(v_j|c_p)}{P(v_j|c_q)} + \log\frac{P(v_j|c_q)}{P(v_j|c_p)} = 0$ (Similar definition and constraints for b_{c_p,c_q}). As a result, the optimization problem becomes:

$$\begin{aligned}
\min\;\; & \sum_{c_p,c_q}^{C}\left(\|\mathbf{w}_{c_p,c_q}\|^2 + b^2_{c_p,c_q}\right)+\lambda\sum_{i,c}\xi_{i,c}\\
\text{s.t.}\;\; & \mathbf{w}^{\top}_{c_i,c}\mathbf{h}_i + b_{c_i,c}\geq 1-\xi_{i,c};\quad \forall i=1,2,\cdots,n;\\
& w^j_{c_p,c_k}+w^j_{c_k,c_q}=w^j_{c_p,c_q};\;\; w^j_{c_p,c_q}+w^j_{c_q,c_p}=0;\\
& b_{c_p,c_k}+b_{c_k,c_q}=b_{c_p,c_q};\;\; b_{c_p,c_q}+b_{c_q,c_p}=0;\\
& \forall p\neq k, k\neq q, q\neq p,\;\; p,k,q,c=1,2,\cdots,C;\\
& \xi_{i,c}\geq 0,\;\; c\neq c_i,\;\forall\, j=1,2,\cdots,t;
\end{aligned} \tag{23}$$

where $\mathbf{w}_{c_p,c_q} = (w^1_{c_p,c_q}, w^2_{c_p,c_q},\cdots,w^t_{c_p,c_q})^{\top}$. Examining this problem shows that it is still QP. However, we cannot leverage a highly efficient SVM solver anymore since it is not equivalent to any multi-class SVM formulation. Its number of variables and constraints gradually rises with the increasing number of classes, words and training samples. Consequently, this problem will become difficult to handle on a data set having a large number of classes, many training samples and a large-size initial codebook. In this paper, we test this multi-class extension on the data sets with a smaller number of classes and words to preliminarily demonstrate its effectiveness. More efficient solutions will be explored in our future work. The problem in Eq.(23) is currently solved by CVX [5] package.

[5] Strictly speaking, the margin defined in Eq.19 is invariant after multiplying a scaling factor to $\log\frac{P(v_j|c_i)}{P(v_j|c)}$ and $\log\frac{P(c_i)}{P(c)}$. To make a simple analysis, here we set this scaling factor to 1. But other choices are also acceptable. In fact, this factor will give an extra tuning parameter for estimating $P(v_j|c_i)$ and $P(c_i)$.

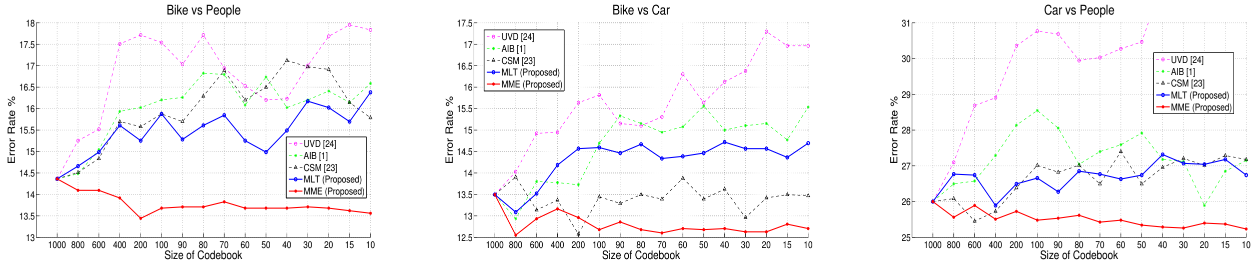


Figure 1: Comparison of five compact codebook creation algorithms on the Graz02 data set.

4. Experimental Result

This experiment aims to verify the effectiveness of our generalized probabilistic framework. The proposed new criteria, MLT and MME, will be compared with the existing criterion UVD [24], AIB [1], and CSM [23]. We implement UVD by following [24] and use AIB and CSM codes provided by their authors. Two indexes are used to evaluate the quality of compact codebooks. One is the classification error rate versus codebook size, used in [1, 10, 23, 24]. For a same codebook size, the lower the error rates, the better the criterion. The other one is the "stability" of the classification performance of the created codebooks. A better criterion will show smaller fluctuation with the decreasing codebook size. This will mitigate the issue of choosing the optimal codebook size. Here, we use the variance of classification error rate to measure the stability. A linear SVM classifier is used. Its regularization parameter is equally optimized via 5-fold cross-validation for each algorithm to ensure fair comparison. Three benchmark data sets, Graz02 [15], 15-Scenes [9], and KTH [16], are used, corresponding object classification, scene classification, and action classification tasks, respectively. With the binaries provided by VGG group[6], we use the Harris-Affine detector [12] to locate interest regions, and then represent them by the SIFT descriptor [11]. Each comparison will be conducted on ten pairs of randomly split training/test sets, and the average result is used. A large initial codebook is created by applying k-means to the local descriptors from training samples only. This experiment mainly focuses on binary classification cases by using one-vs-one and one-vs-rest settings. This allows us to verify the correspondence of the optimization problems in MME and binary SVMs, as identified in section 3.4. We use the off-the-shelf SVM solver (libsvm [3]) to estimate parameters for MME. Meanwhile, we investigate the multi-class extension of MME and preliminarily demonstrate its effectiveness on multi-class classification tasks. In the experiment, the following two points will be verified: *i) if the proposed MME is the overall best; ii) if the proposed MLT is better than UVD.*

4.1. Object classification on the Graz02 data set

Graz02 is a challenging data set because each object can appear in an image with different location, position, size, and view angle. Also, the background can occupy a large portion of an image, making the histograms contain many "noisy bins". There are three object classes, namely, Car, Bicycle and Person. We use one-vs-one setting to form three binary classification problems. An initial codebook with the size of 1000 is used to create compact codebooks.

As presented in Figure 1, the MME criterion produces the lowest error in all classification tasks, and clearly outperforms the others. Also, it shows clear improvement over AIB which only differs to MME at the parameter estimation method. This can be understood as that the max-margin-based parameter estimation is able to emphasize more on discriminative bins and handle noisy bins more effectively. Another important observation is that by using MME, the created compact codebooks can even achieve the classification performance better than that using the initial large codebook. Also, in all the three tasks, MLT achieves better performance than UVD by replacing the Gaussian distribution with a multinomial distribution. As can be seen in Figure 1, the MME criterion also demonstrates a significantly lower variance than the others. (Quantitative measurement of the variance can be found in the supplemental material). The superior performance of MME and the improvement of MLT over UVD preliminarily demonstrate the importance of properly setting the two key factors and the generalizability of our probabilistic framework.

4.2. Scene classification on the 15-Scenes data set

The five criteria are further compared on discriminating 15 different classes of scenes in the benchmark 15-Scenes data set. We exhaustively test all the 105 pairwise classification cases, and report the average pairwise classification

[6] http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries

performance. Again, all these are conducted on ten pairs of randomly split training/test sets and the average result is used. The size of the initial codebook is 1000.

Figure 2(a) plots the average pairwise classification error rate. As seen, by the proposed MME, the created codebooks consistently give the highest performance. Moreover, most of these compact codebooks produce better performance than the initial codebook of size 1000, even if the codebook size reduces to a number as low as 10. These clearly indicates the excellent performance of MME. In terms of performance stability, MME is also the most stable one with respect to different codebook sizes, as demonstrated by the variance listed in supplemental material. As examples, sub-figure(b) and (c) show two pairwise classification cases where MME achieves the most significant improvement and the advantage of MME becomes more pronounced. Focusing on MLT and UVD and going through the above comparison again, we can see that MLT gives better classification performance than UVD as expected.

4.3. Action classification on the KTH data set

In recent years, the Bag-of-Word model has also been applied to action recognition by extracting local spatial-temporal features of videos [8, 21]. This experiment is carried out on the KTH data set, a benchmark data set in action recognition [16]. It consists of 25 subjects performing 6 actions: boxing, hand-clapping, jogging, running, walking and hand-waving. We randomly choose 16 subjects for training and the remaining 9 subjects for test, forming 10 pairs of training/testing sets. Laptev's spatial-temporal feature proposed in [8] is used. By using k-means clustering, an initial large codebook with size of 4000 is obtained. In this experiment, we compare the five criteria in the setting of one-vs-rest binary classification tasks, each of which discriminate one target action from the others. We report the classification performance averaged over the 6 one-vs-rest classification tasks in Figure 3(a). Along with it, we particularly show the comparison on the two most challenging tasks (identified based on both our experiment and those reported in [16]), that is, running vs. the rest, and jogging vs the rest. For the other easier tasks, all criteria give nearly perfect classification results.

As shown in all the sub-figures of Figure 3, MME still achieves top performance and only CSM is comparable to it. However, reviewing the results of CSM in all the previous experiments shows that the overall performance of CSM is far behind that of MME. As for the criteria of UVD, AIB and MLT, their performance is inferior and can deteriorate quickly with the decreasing codebook size. Comparatively, UVD shows the worst performance, but MLT still manages to keep clear improvement over UVD, especially when the codebook size is smaller.

4.4. Preliminary result on multi-class classification

We conduct preliminary study of our multi-class MME extension . Following previous experimental settings, the five criteria are compared in terms of the classification error rates averaged on 10 training/test pairs. Due to the scalability problem of current MME, this experiment is carried on Graz02 and a "reduced" version of KTH and 15-scenes data sets. For KTH, only the three most confused classes, namely, hand waving, jogging and running are selected; For 15-Scenes, we choose bedroom, kitchen and industry which have been demonstrated in previous binary classification experiment. The initial codebook size in all three data sets are reduced to 100. The results are shown in Figure 4. It is clear that MME still produces the best performance with the decreasing codebook size. Also, we observe that MLT again achieves better performance than UVD.

5. Conclusion

We have presented a generalized probabilistic framework, with which we unify existing merging criteria and design new criteria for compact codebook construction. With the better performance achieved by the new criteria, we have demonstrated the importance of properly setting the two key factors of this framework. In future work, we will address the scalability issue in the current multi-class MME method. Also, more effective merging criteria are to be explored within this framework.

References

[1] B.Fulkerson, A. Vedaldi, and S. Soatto. Localizing objects with smart dictionaries. In *Proc. Eur. Conf. Comp. Vis.*, 2008.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[3] C.-C. Chang and C.-J. Lin. *LIBSVM: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`*, 2001.

[4] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. Eur. Conf. Comp. Vis.*, 2004.

[5] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, Oct. 2010.

[6] R. Herbrich and T. Graepel. Bayes point machines. *J. Mach. Learn. Res.*, 1:245–279, 2001.

[7] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proc. IEEE Int. Conf. Comp. Vis.*, Washington, DC, USA, 2005. IEEE Computer Society.

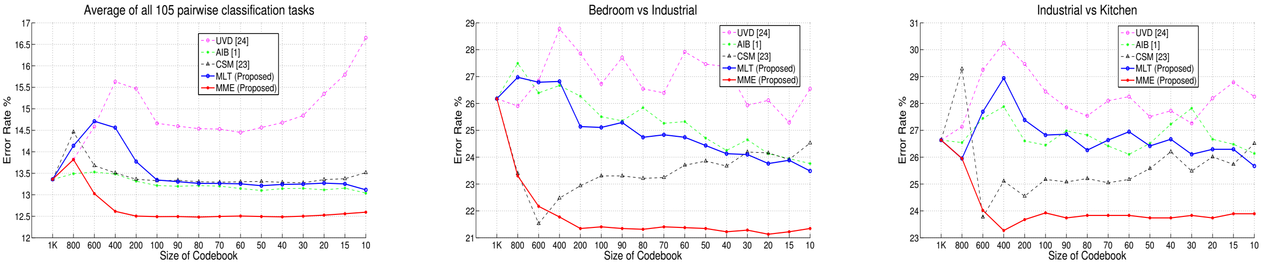
[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–8, June 2008.

[9] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31:1294–1309, 2009.

[10] J. Liu and M. Shah. Learning human actions via information maximization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2008.

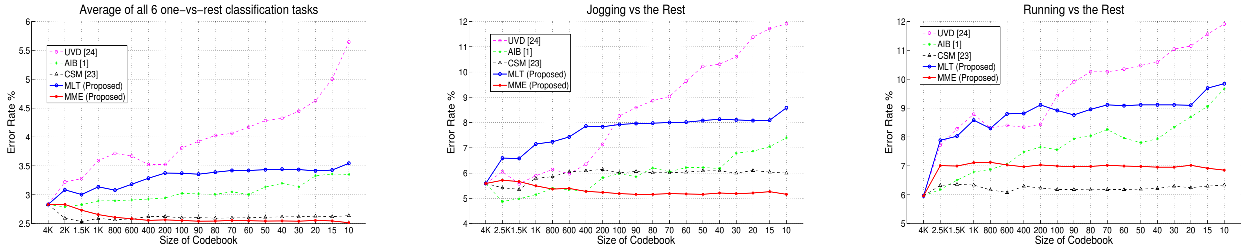
[11] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1150–1157, 1999.

[12] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comp. Vis.*, 60(1):63–86, 2004.



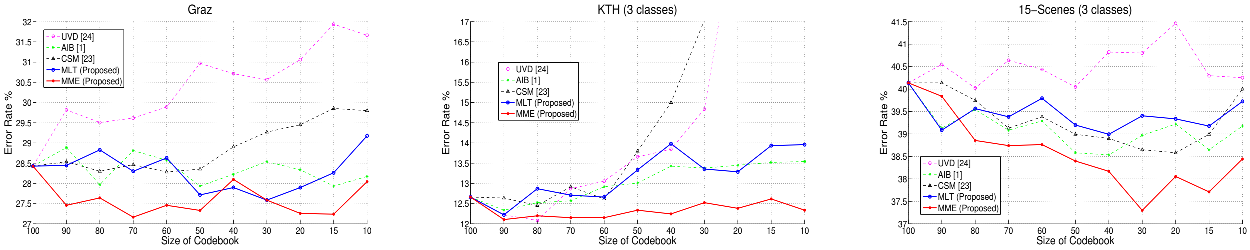
(a) (b) (c)

Figure 2: Comparison of five compact codebook creation algorithms on the 15-Scenes data set.



(a) (b) (c)

Figure 3: Comparison of five compact codebook creation algorithms on the KTH data set.



(a) (b) (c)

Figure 4: Comparison on three data sets in the multi-class setting.

[13] T. P. Minka. *Estimating a Dirichlet distribution*, 2003.

[14] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Proc. Adv. Neural Inf. Process. Syst.*, pages 985–992. MIT Press, Cambridge, MA, 2007.

[15] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28:2006, 2004.

[16] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. *Proc. IEEE Int. Conf. Patt. Recogn.*, 3:32–36, 2004.

[17] C. Shen, H. Li, and M. J. Brooks. Supervised dimensionality reduction via sequential semidefinite programming. *Pattern Recogn.*, 41(12):3644–3652, December 2008.

[18] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, volume 2, pages 1470–1477, Oct. 2003.

[19] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 617–623, 1999.

[20] A. Vedaldi and B. Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[21] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. British Mach. Vis. Conf.*, page 127, sep 2009.

[22] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2010.

[23] L. Wang, L. Zhou, and C. Shen. A fast algorithm for creating a compact and discriminative visual codebook. In *Proc. Eur. Conf. Comp. Vis.*, volume 4, pages 719–732, Marseille, France, October 2008.

[24] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. IEEE Int. Conf. Comp. Vis.*, volume 2, pages 1800–1807, 17-21 Oct 2005.