SHRINKING LARGE VISUAL VOCABULARIES USING MULTI-LABEL AGGLOMERATIVE INFORMATION BOTTLENECK

Wojciech Wojcikiewicz[†‡]

Alexander Binder[†], *Motoaki Kawanabe*[‡†]

[†]Technical University Berlin
Department of Computer Science
Franklinstr. 28 / 29, 10587 Berlin, Germany

[‡]Fraunhofer Institute FIRST
Intelligent Data Analysis Group
Kekuléstr. 7, 12489 Berlin, Germany

ABSTRACT

The quality of visual vocabularies is crucial for the performance of bag-of-words image classification methods. Several approaches have been developed for codebook construction, the most popular method is to cluster a set of image features (e.g. SIFT) by k-means. In this paper, we propose a two-step procedure which incorporates label information into the clustering process by efficiently generating a large and informative vocabulary using class-wise k-means and reducing its size by agglomerative information bottleneck (AIB). We introduce an extension of the AIB procedure for multi-label problems and show that this two-step approach improves the classification results while reducing computation time compared to the vanilla k-means. We analyse the reasons for the performance gain on the PASCAL VOC 2007 data set.

1. INTRODUCTION

Bag-of-visual-words models [1, 2] have been successfully applied to image classification problems in recent years. In the first step, image features e.g. SIFT descriptors are computed on a dense grid or from keypoints, then they are clustered into visual words, so that an image can be finally represented by a fixed-size histogram over the visual words. A popular method for codebook generation is to cluster (a subset of) the SIFTs from all training images using the k-means algorithm (`KM`). Although this approach has been successfully applied to various computer vision tasks, a couple of its drawbacks have been recognized.

First of all it is not feasible to construct codebooks larger than a few thousands visual words with this method. This can be overcome by using hierarchical k-means (`HKM`) [3] i.e. applying k-means in a hierarchical manner. Although this seems to work quite well in practice, it is not clear how to select the number of levels and the number of clusters per level a priori. Another drawback of `KM` (and `HKM`) is that no label information is used when clustering SIFT descriptors from all classes together. Recent studies [4, 5] showed that incorporating class-specific data into the classification can provide better results. Finally it has been showed [6] that k-means chooses most cluster centers to be near high density regions, thus under-representing equally discriminant low-to-medium density ones.

In order to overcome these issues of `KM`, as the first step, we deploy the class-wise k-means (`CWKM`) procedure proposed by Farquhar *et al.* [1], i.e. we construct small vocabularies for each class and then aggregate them into one large vocabulary. This approach allows to construct large vocabularies very fast, since only small class-specific codebooks need to be generated and the process can be performed in parallel for different classes. Furthermore label information is incorporated in a natural way.

In order to reduce the processing time for new query images, particularly in word assignment and kernel computation steps, we construct mid-size vocabularies based on reclustering the visual words by agglomerative information bottleneck (AIB) [7, 5]. We introduce a modification of AIB suitable for multi-label problems we deal with. Furthermore, we propose a novel criterion 'purity' to quantify informativeness of the visual words. We show that our approach with `CWKM` and purity-enhanced AIB outperforms the baselines with the `KM` vocabularies.

This paper is organized as follows. In Section 2, we explain our two-step procedure and the purity measure. After describing our experimental setup in Section 3, we compare the performance between `KM` and `CWKM` with and without word reclustring on PASCAL VOC 2007 data. Section 5 concludes with future research issues.

2. CODEBOOK GENERATION PROCEDURE

2.1. Class-wise Clustered Vocabularies

At first, for all classes $c = 1, \dots, C$, we construct a small vocabulary $V_c = \{v_{c1}, \dots, v_{cm}\}$ by applying k-means clustering to the SIFT descriptors $\{s_{c1}, \dots, s_{cd}\}$ selected from the training images containing category c. Then, the resulting vocabularies are aggregated into one large overall vocabulary $V = \{V_1, \dots, V_C\}$ (size $C \times m$). In this way, we can produce large vocabularies which achieve better classification performance in a fraction of time (hours vs. days) compared to `KM`.

2.2. Multi-label Agglomerative Information Bottleneck

The information bottleneck technique creates a compact representation of the inputs, while keeping supervised label information. Since image annotation requires multiple labels in general, we need to consider the label indicator Y which takes $2^{|C|}$ different states.

Let W and $\tilde{W}$ be the original and shrunk vocabularies, respectively. AIB clusters words in a hierarchical manner based on the decrease in the mutual information $I(Y, \tilde{W})$. That means AIB starts with a trivial partition $\tilde{W}_0$ where each cluster is represented by a word from W and at each step i it creates a new partition $\tilde{W}_i$ by merging two clusters from vocabulary $\tilde{W}_{i-1}$ into a single new component such that the merging loss $I(Y, \tilde{W}_{i-1}) - I(Y, \tilde{W}_i)$ is minimized.

In order to avoid probability estimation and implementation problems due to the large number of states of Y, we propose a modification of the agglomerative information bottleneck. Our idea is replacing $I(Y, \tilde{W})$ by its upper bond $\sum_{c=1}^{|C|} I(Y_c, \tilde{W})$, where $Y_c \in \{1, -1\}$ is the binary indicator for each class c. When two clusters $(\tilde{w}_i, \tilde{w}_j)$ are merged into $\tilde{w}_*$, the probabilities in information bottleneck are changed as

$$P(\tilde{w}_*|w) = \begin{cases} 1 & w \in \tilde{w}_i \text{ or } w \in \tilde{w}_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$P(y|\tilde{w}_*) = \frac{P(\tilde{w}_i)}{P(\tilde{w}_*)} P(y|\tilde{w}_i) + \frac{P(\tilde{w}_j)}{P(\tilde{w}_*)} P(y|\tilde{w}_j), \tag{2}$$

$$P(\tilde{w}_*) = P(\tilde{w}_i) + P(\tilde{w}_j). \tag{3}$$

At each step, AIB merges the pair of clusters which minimizes the merge loss

$$\begin{aligned} \delta I(Y, \tilde{W}) &:= \sum_{c=1}^{|C|} I(Y_c, \tilde{W}_{\text{before}}) - \sum_{c=1}^{|C|} I(Y_c, \tilde{W}_{\text{after}}) \\ &= \{P(\tilde{w}_i) + P(\tilde{w}_j)\} \sum_{c=1}^{|C|} D_{\text{JS}}(P(y_c|\tilde{w}_i), P(y_c|\tilde{w}_j)), \end{aligned} \tag{4}$$

where D_{JS} is the Jensen-Shannon divergence defined with Kullback-Leibler divergence D as

$$\begin{aligned} D_{\text{JS}}(P(y_c|\tilde{w}_i), P(y_c|\tilde{w}_j)) &:= P(\tilde{w}_i) D(P(y_c|\tilde{w}_i)||P(y_c|\tilde{w}_*)) \\ &+ P(\tilde{w}_j) D(P(y_c|\tilde{w}_j)||P(y_c|\tilde{w}_*)). \end{aligned} \tag{5}$$

2.3. Purity Measure

When selecting visual words, we need a ranking criterion. We introduce a novel ranking criterion here, called *purity measure* as it measures the average importance of a word with respect to the object classes.

Let $c = 1, \ldots, C$ be an object category, $y_c \in \{1, -1\}$ be the indicator of the class c, and $w \in \{1, \ldots, |W|\}$ be a visual word in a vocabulary W. Suppose that we have the joint probability $P(y_c, w)$, where $\sum_{y_c} \sum_w P(y_c, w) = 1$. We propose a purity measure of the word w for class c by Kullback-Leibler divergence as:

$$\rho_c(w) := \sum_{y_c = \pm 1} P(y_c|w) \log \frac{P(y_c|w)}{P(y_c)} \tag{6}$$

This measure is zero if the occurence of word w is independent of the label c, it is greater than zero otherwise. The final ranking criterion used in this paper is the average purity over the object classes:

$$r(w) := \frac{1}{C} \sum_{c=1}^{C} \frac{\rho_c(w)}{\rho_c^{\max}}, \tag{7}$$

where $\rho_c^{\max} := \max\{-\log P(y_c = 1), -\log P(y_c = -1)\}$ is the maximum value of $\rho_c(w)$ introduced for compensating the unbalanced class probabilities $P(y_c)$.

2.4. Shrinking Large Codebooks

Different approaches can be used to reduce the size of a visual vocabulary. In this paper we are mainly interested in the multi-label AIB algorithm as it retains label information. We compare several methods:

- `purity`: Select purest words, do not apply AIB.
- `AIB_center`: New visual words are the centers of clusters created by AIB.
- `AIB_purity`: New visual words are the set of most pure words from each cluster created by AIB.
- `AIB_genuine`: Use the word assignments from `CWKM20000`, sum up occurence frequencies of words in each AIB cluster.[1]

2.5. Related Work

Farquhar *et al.* [1] applied dimensionality reduction (e.g. PCA and PLS) to the large vocabularies obtained by `CWKM` in order to get compact representations, but it does not alleviate the word assignment and projection onto a low-dimensional subspace is hard to interpret. Fulkerson *et al.* [5] also proposed a two-step approach similar to `AIB_genuine` by using AIB class-wise to shrink large `HKM` vocabularies. However, from our experiences, `HKM` generally performs significantly worse than `CWKM` (0.453 vs. 0.475 in mean AP score) and is sensitive with respect to the choice of cell per level. Furthermore, the results show that `AIB_genuine` does not improve `KM` in mid vocabulary sizes (see Table 1). Thus, we propose to use a reduced vocabulary and perform word assignment instead.

[1] Assigning image features to nearest visual words is part of the bag-of-words pipeline and needs to be done for every image we want to classify. `AIB_genuine` uses the assignments of `CWKM20000` and simply sums up the frequency counts for each word in an AIB cluster. Thus no distortions are introduced as no further word assignment is needed, but computing the assignments for `CWKM20000` is slow and requires the large codebook.

3. EXPERIMENTAL SETUP

In our experiments we used the VOC 2007 data set [8] containing 9963 images of 20 object classes. We used the official split into training, validation and test and employed a histograms of visual words (HoW) representations based on SIFT descriptors over the grey channel and whole image extracted on a dense grid of pitch six. Our choice of kernel function is the χ^2 kernel, the kernel width was set to be the mean of the χ^2 distances between all pairs of training samples.

The evaluation is based on precision-recall curves. The principal quantitative measure is average precision (AP) over all recall values. The regularization parameter C of the SVM was optimized on the validation data. We provide test results.

4. RESULTS

We compared the class-wise clustered vocabulary with 20000 words (CWKM20000) with the standard 4000 words codebook (KM4000). The mean AP (MAP) gain over all classes for CWKM is 0.0205 (or 4.5 %), however, the maximal increase in AP is 0.0838 (or 33.2 %) for the class 'diningtable' and no class performs worse. We remark that the time needed for clustering is much less for CWKM, i.e. few hours vs. one week.

Since large vocabularies have disadvantages in terms of computation time[2], we constructed mid-size vocabularies directly by CWKM, but also created codebooks in a two-step approach i.e. we first generated a large CWKM codebook (20000 words) and reduced its size. The results for different vocabulary sizes are summarized in Table 1. As can be seen the direct application of CWKM does not work too well (see second row), even worse than KM, however, the codebook construction is much faster. This can be explained by an insufficient number of words in the class-specific vocabularies e.g. in the 800 words case only 40 words per class are used, which is probably not enough to capture the relevant class information.

Much better results can be achieved when using a two-step approach. For example when using AIB_purity we obtain an average performance gain of up to 4%. However, the performance change is not uniformly distributed over the classes. Some classes perform much better, even up to 22%, few classes are doing worse (-3%) and the performance of the other classes does not change much at all. A potential explanation for this nonuniformity gives the negative correlation between performance change and absolute performance e.g. for the 4000 words vocabulary the correlation coefficient is -0.61 which means that classes which perform bad on KM (e.g. 'diningtable') have the largest performance gain, whereas classes which are classified well (e.g. 'person') do not perform better or even lose a bit.

So the question is why do classes like 'diningtable' perform so badly on KM compared to the two-step procedure. We conjecture that these classes are not well represented by the KM vocabulary e.g. due to concentration on high-density regions in descriptor space. Since we are using class-wise clustering in the two-step procedure with a large number of words, each class is represented well in the CWKM20000 vocabulary. In fact, no class performs worse with the CWKM codebook compared to KM4000. So we have a better vocabulary and reduce it by a supervised method, AIB in this case. Therefore the new codebook represents those classes better than the direct KM approach and thus provides better results.

Method	4000 words	2000 words	800 words
KM	45.45	44.42	41.23
CWKM	45.15	44.01	40.23
purity	45.45	43.98	41.42
AIB_center	45.51	44.68	42.18
AIB_purity	**46.30**	**45.14**	42.98
AIB_genuine	45.58	45.03	**43.24**

Table 1. Results (MAP × 100) for VOC images. CWKM20000 result is 47.49.

Comparing the different selection methods shows that selecting words solely based on purity value performs worst. This may be because ranking individual words does not take into account interactions between words e.g. it is possible that words which are related to only one (or few) classes or are located in some small subregion in descriptor space are ranked very high and are therefore selected, however, the vocabulary is not representative and does not perform well in this case. Methods taking into account the whole vocabulary (like AIB) provide better results. This claim is supported by figure 1. It shows from which class-specific vocabulary[3] the selected words come from. We see that selecting words based on purity value alone overemphasises some classes e.g. class 1 'aeroplane' or class 20 'tvmonitor' while almost ignoring other e.g. class 10 'cow'. In contrast to that AIB selection is balanced, thus much more representative.

When comparing AIB_center and AIB_purity, we see that the latter produces better results. We conjecture that AIB_center performs worse as AIB does not take into account distances between words in a cluster i.e. clusters may consist of words located very far apart in descriptor space, thus taking the center of the cluster as new word does not make sense in such a case. Figure 2 shows the maximal distance between words from the same cluster for AIB and k-means reclustering for different sizes. We see that the AIB clusters are much larger than the clusters created by k-means, this may be the reason why AIB_center does not perform so well. However, we can motivate the use of AIB_purity when regarding AIB clusters as groups of words containing similar information about the label vector Y i.e. by chosing

[2] The time needed for histogram and kernel computation grows linearly with the number of words

[3] Note that CWKM is an aggregation of 20 class-specific vobabularies

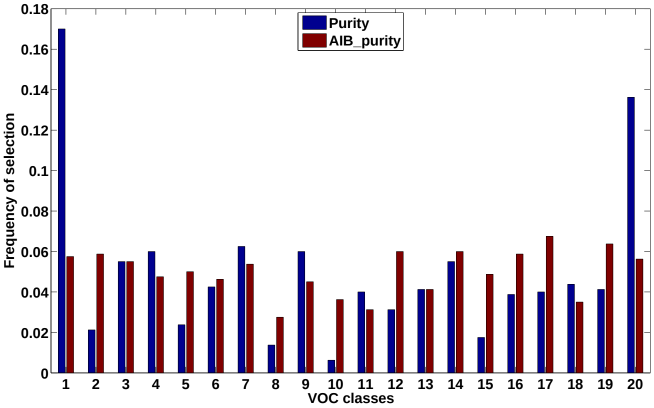

Fig. 1. Frequency of selection of words from the 20 VOC classes for AIB_purity and purity

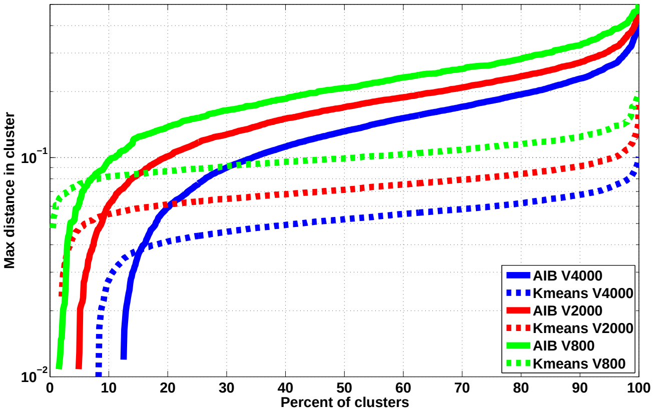

Fig. 2. Maximal distance between words inside a cluster (in log-scale) for AIB and k-means reclustering.

the purest word from each cluster, we make sure to conserve most relevant information in the new reduced vocabulary.

The method AIB_genuine was motivated by the fact that it represents the real AIB partitioning of the descriptor space without any distortion[4] from the subsequent word assignment when using a reduced vocabulary. It seems that this approach overfits in the validation phase as AIB was performed on trainval data i.e. the validation performance is almost 25% better than the corresponding AIB_purity one, but it does not perform so well in test phase. Furthermore we noticed a difference in the purity values between trainval and test data for some classes e.g. the correlation between the purity values computed on trainval and test data (from CWKM20000) is over 0.9 for class 13 'horse', but less than 0.4 for class 16 'pottedplant'. Thus overfitting and a change in the purity distribution may be the reasons for the modest results of AIB_genuine.

5. CONCLUSION

In this paper, we evaluated class-wise clustered vocabularies on a challenging data set. We have shown that class-wise clustering produces better results while allowing to generate large vocabularies in a fraction of time compared to KM (hours vs. days). We introduced an extension of the agglomerative information bottleneck algorithm to multi-label data and showed that it can be used to reduce the vocabulary while keeping label information. Furthermore we introduced a novel measure of purity which can be used as ranking criterion for visual words. Since CWKM does not scale with the number of classes, we are investigating the use of a group-wise clustering approach i.e. groups of similar object classes are created by the user and vocabularies are clustered for each group separately. This should bring a performance gain compared to the all-in-one approach.

Acknowledgements We thank Klaus-Robert Müller for valuable discussions. This work was supported by the Federal Ministry of Economics and Technology of Germany under the project THESEUS (01MQ07018), by the German Research Foundation (GRK 1589/1) and the FP7-ICT Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-216886.

6. REFERENCES

[1] J. Farquhar, S. Szedmak, H. Meng, and J. Shawe-Taylor, "Improving bag-of-keypoints image categorisation," Tech. Rep., University of Southampton, 2005.

[2] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *CIVR '07*, 2007.

[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR '06*, 2006.

[4] F. Perronnin, C. Dance, G. Csurka, and M. Bressan, "Adapted vocabularies for generic visual categorization," in *ECCV '06*, 2006, pp. 464–475.

[5] B. Fulkerson, A. Vedaldi, and S. Soatto, "Localizing objects with smart dictionaries," in *ECCV '08*, 2008.

[6] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *ICCV '05*, 2005.

[7] N. Slonim and N. Tishby, "Agglomerative information bottleneck," in *Advances in Neural Information Processing Systems*. 1999, vol. 12, pp. 617–623, MIT Press.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," .

[4] When words inside an AIB cluster are far apart, not all SIFT features assigned to them will be assigned to the new visual word created by the cluster.