CHAPTER 6

Structured Dictionary Learning and Feature Encoding

Chapter abstract: *Sparse, redundant representations offer a powerful emerging model for signals and images (Mairal et al., 2008; Yang et al., 2009). This model approximates a data vector as a linear combination of a subset of basis elements from a learnt over-complete basis set. The combination is a sparse selection of basis elements. Sparsity is induced by adding a regularization constraint to the coefficients in the loss function. The degree of sparsity is typically determined empirically. A ℓ_0 or ℓ_1-norm is used to regularize the coefficients. This effective approach however considers each basis element individually. It does not utilize prior information of structure within the basis set and disregards possible groups of basis elements. In this chapter, sparse coding is augmented with structure learnt from co-clustering in terms of groups of semantically related basis elements. In the first part, a structured multiple manifold dictionary is learnt using co-clustered sub-spaces with Structured Sparse Principal Component Analysis (SSPCA) (Jenatton et al., 2009). It builds upon Sparse-PCA (Zou et al., 2004) where the sparse selection of sub-spaces to represent a feature vector is independent of semantic significance of the selected sub-spaces. The second part deals with learning structured feature encoding. Typically, the Lasso which uses ℓ_1-norm regularization is used for sparse selection of dictionary elements. To incorporate a prior structure, the group Lasso which uses $\ell_{2,1}$ mixed norm regularization (Zibulevsky and Elad, 2010) is used for sparse selection of groups of semantically related dictionary elements. Both structured sparse visual models are shown to improve performance over their corresponding regular sparse visual models.*

6.1 Introduction

In this chapter, semantically relevant structure that was estimated in previous chapters is incorporated in learning a visual model. The method selected for this

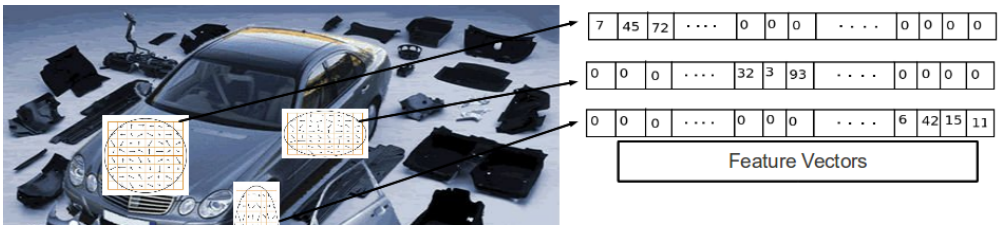

Figure 6.1: Conceptual illustration of relation between object part and descriptor sub-space. Different parts of a 'car' due to the local image structure are embedded in different sub-manifolds in feature space. A single sub-manifold can not satisfactorily model all three 'car' parts.

is sparse coding. Sparsity inducing algorithms for learning a visual model have enjoyed much success in recent years (Yang et al., 2009). They have the ability of simultaneous feature selection and model learning. One explanation for their notable performance is their ability to deal with intra-category appearance variation. Consider typical training feature data. It contains more elements than is required to represent any instance of any visual category. For example, the estimated intrinsic dimensionality of 128-dimensional feature vectors is in the neighbourhood of 14, based on the empirical results in section 3.2. The methods discussed in previous chapters first compute a single subset of all the sub-spaces. All the elements of this subset are subsequently used to represent every feature descriptor. However, a visual category consists of distinct parts, and each part spans a different subset of the basis set. A single subset can not satisfactorily accommodate the variation in appearance found in a visual category. In other words, regardless of the efficacy of the sub-space embedding technique, a single sub-manifold is inappropriate for all feature descriptors. A conceptual illustration of this is shown in fig. 6.1. Different parts of the visual object 'car' are embedded in different sub-manifolds due to their distinct local patch structure. A single sub-manifold can not account for all 'car' parts. Sparsity inducing methods on the other hand, do not select a subset of basis elements a priori. A feature descriptor is assigned its own subset of basis elements. Although the cardinality of the subset in both, sparse coding and previously discussed methods may be the same, it is the ability for individual subset selection that makes sparse coding better suited to model intra-category appearance variation.

Stated formally, the aim of sparse coding is to find a set of basis vectors $\{\mathbf{d}_1, \ldots, \mathbf{d}_r\}$ such that an input image $\mathbf{Z} \in \mathbb{R}^{p \times n}$ can be represented as a linear combination of these basis vectors $\mathbf{Z} = \sum_{i=1}^{r} \alpha_i \mathbf{d}_i$. While techniques such as PCA learn a complete set of basis vectors efficiently, the aim is to learn an over-complete set of basis vectors to represent input images. The elements of an over-complete basis are better able to capture structures and patterns inherent in the input data. However, with an over-complete basis, the coefficients α_i are no longer uniquely determined by the input vector $\mathbf{Z}$. Therefore, in sparse coding, the additional criterion of sparsity is utilized to resolve the degeneracy introduced by over-completeness. In the experiments in previous chapters, a 'universal' dictionary had been learnt from a corpus of all visual categories in a dataset[1]. A universal dictionary is an over-complete dictionary, as it contains elements from multiple visual categories.

The data matrix $\mathbf{Z} \in \mathbb{R}^{p \times n}$ is factorized to dictionary matrix $\mathbf{D} \in \mathbb{R}^{p \times r}$, and coefficient matrix $\mathbf{A} \in \mathbb{R}^{p \times n}$, with a sparse regularization constraint Ω on the columns of the coefficient matrix. Sparse coding methods solve the following regularized problem

$$\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{z}_i, \mathbf{d}\alpha_i) + \lambda \Omega(\alpha) \tag{6.1}$$

where λ is a regularization parameter. The first term $\frac{1}{n} \sum_{i=1}^{n} L(\mathbf{z}_i, \mathbf{d}\alpha_i)$ is called empirical risk or loss function. It is convex and continuously differentiable (Jenatton et al., 2010). Typical choices for Ω use the ℓ_0-norm or ℓ_1-norm. The regularized problem using the ℓ_0-norm is

$$\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} \parallel \mathbf{z}_i - \mathbf{D}\alpha_i \parallel^2 + \lambda \parallel \alpha_i \parallel_0 \tag{6.2}$$

The use of ℓ_0-norm forces selection of a subset of fixed cardinality. While this is conceptually appealing, efficient regularization methods that use ℓ_0-norm are difficult to implement. The ℓ_1-norm regularization on the other hand has

[1] This choice of dictionary is in contrast to learning a 'category specific' dictionary

become a popular tool. It profits from efficient algorithms (Efron et al., 2004; Lee et al., 2007; Yuen and Torralba, 2009) and a well developed theory for generalization properties and variable selection consistency (Shen et al., 2013; Zhang et al., 2009). The regularization problem using the ℓ_1-norm is

$$\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} \| \mathbf{z}_i - \mathbf{D}\alpha_i \|^2 + \lambda \| \alpha_i \|_1 \tag{6.3}$$

Notwithstanding their notable performance, sparse coding methods have an issue. When regularizing by the ℓ_1-norm, each variable is considered individually. The position of a variable in the input feature vector is disregarded. This means existing relationships and structures between the variables are ignored. However, using sparsity induction in learning a visual model could benefit from this type of prior knowledge. There are two reasons for including structural a priori information. One is improved predictive performance of the learnt model. The other is improved interpretability of the model. While the ℓ_1-norm regularization succeeds in inducing sparsity, it does so without knowledge of the semantic relevance of the variables selected. Incorporating a priori structure can encourage the selection of variables with regard to their semantic relation to other variables.

For example, in the field of face recognition, robustness to occlusions can be increased by incorporating structure. The features are considered as sets of pixels that form small convex regions on the face images (Jenatton et al., 2010). A simple ℓ_1-norm regularization can not encode this specific spatial locality constraint (Jenatton et al., 2010). Another example in the field of computer vision is object and scene recognition, where a goal is the computation of bounding boxes in images (Harzallah et al., 2009). They are detected by modelling the spatial arrangement of the pixels over the images. An unstructured sparsity-inducing regularization that disregards this spatial information is therefore not necessarily able to segment the image with bounding boxes as it may be modelling a mixture of object feature with background contextual features.

These examples motivate the need for sparsity-inducing regularization schemes,

capable of encoding more sophisticated prior knowledge about the expected sparsity patterns. As mentioned above, the ℓ_1-norm focuses only on cardinality and cannot easily specify information about the patterns of non-zero coefficients induced in the solution, since all non-zero patterns are theoretically possible. Group ℓ_1-norms (Huang et al., 2009; Roth and Fischer, 2008; Yuan and Lin, 2006) consider a partition of all variables into a certain number of subsets and penalize the sum of the Euclidean norms of each one, leading to the selection of groups rather than individual variables. The consistency of group sparse method has been studied in (Bach, 2008).

Combining the ideas of sparse models and group structure from co-clustering, is work on structured sparsity induced matrix factorization (Kim et al., 2012). At its heart is a $\ell_{q,1}$-mixed norm where $q \in \{2, \ldots, \infty\}$, where typically $q = 2$ (Liu and Zhang, 2008). It achieves sparsity at the group level, where data elements within a group are treated equally using the ℓ_2-norm, while sparsity is induced upon entire groups using the ℓ_1-norm. The group Lasso (Bach, 2008) is the $\ell_{2,1}$-norm regularization equivalent of Lasso (Tibshirani, 1994) for ℓ_1-norm regularization. During matrix factorization the matrices within each group are orthogonal.

In the previous chapter, co-clustering was utilized to estimate groups of sub-spaces and groups of dictionary elements. In this chapter, the estimated groups of sub-spaces are used in conjunction with structured sparse principal component analysis (SSPCA) to learn a group structured multiple-manifold dictionary, in section 6.2. This dictionary is compared to a dictionary computed using regular sparse principal component analysis (SPCA). The empirical evaluation and results are described in section 6.2.3. The groups of dictionary elements are used with group Lasso to compute a group structured sparse encoding, in section 6.3. The structured sparse encoding is empirically compared to regular sparse encoding in section 6.3.3.

6.1.1 Contributions

The principal contributions in this chapter are:

- Groups of sub-spaces estimated to be semantically related by co-clustering are used with SSPCA to learn a dictionary on semantically relevant multiple sub-manifolds. Unlike SPCA, the choice of subspaces is relevant since sparsity is induced on the selection of sub-manifolds rather than individual sub-spaces. This novel approach is empirically shown to provide performance benefits.
- Groups of dictionary elements estimated by co-clustering are used with group Lasso to learn structured encoding of images. The $\ell_{2,1}$ mixed norm regularization is used to induce sparsity in selection of groups of dictionary elements, rather than individual elements. The experiments show a small aggregate performance improvement over regular sparse coding.

6.2 Structured Sub-manifold Dictionary

This section discusses learning a structured sub-manifold dictionary. First it describes learning a sparse sub-manifold dictionary using SPCA. Next groups of sub-spaces are used with SSPCA to compute a structured sparse sub-manifold dictionary.

6.2.1 Sparse Subspace Dictionary

Classical PCA, which has been discussed in section 3.3.1, has two interpretations. Typically, it is viewed as a method for computing orthogonal directions maximizing the variance of the Eigenvectors. The Eigenvectors constitute the learnt basis set. This is viewing PCA as a tool for analysis. The other interpretation is to view it as a tool for synthesis, where PCA finds a basis, or orthogonal dictionary, such that the feature vectors admit decompositions with

low reconstruction error. The point to note is that these two interpretations recover the same basis of principal components for PCA. Upon computing PCA, the next step is to reduce the basis set. The typical approach is to assign all basis elements below an arbitrary pre-specified threshold to zero. Motivated by the applicability of Lasso (Tibshirani, 1994) for sparse selection, Zou et al. (2004) formulated PCA as a regression type optimization problem, where Lasso can be utilized as a regression criterion, so that the modified PCA encourages sparse selection. This modified PCA method is called Sparse-PCA (SPCA). Different formulations of SPCA have been proposed in (Aspremont and Bach, 2008; Jolliffe, 2002; Zou and Hastie, 2003). The analysis and synthesis interpretations of SPCA have different formulations. Of interest here is the synthesis interpretation. It leads to non-convex global formulations which simultaneously estimate all principal components (Mairal et al., 2010).

The key element of a sparse subspace dictionary learning technique using matrix factorization is the regularization term on the dictionary elements. For a training set of feature vectors $\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ in a matrix $\mathbf{Z} \in \mathbb{R}^{p \times n}$, the matrix factorization routine for dictionary size r computes a coefficient matrix $\mathbf{A} \in \mathbb{R}^{n \times r}$ and dictionary $\mathbf{D} \in \mathbb{R}^{p \times r}$. The dictionary matrix has r columns which are the dictionary elements $\{\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(r)}\}$. The factorization aims to represent the columns of $\mathbf{Z}$ as a combination of the columns of $\mathbf{D}$, with minimum error and with sparse dictionary elements. Note that the regularization is not induced on the coefficient matrix $\mathbf{A}$, which would correspond to a sparse selection of dictionary elements to represent a feature vector $\mathbf{z}$. Regularization applied to $\mathbf{D}$ means that all dictionary elements are used to represent each feature vector, but that each dictionary element is individually encouraged to be sparse. The factorization is formulated as a convex optimization problem with regularization $\Omega_{\mathbf{D}}$ for the dictionary elements as

$$\min_{\mathbf{D}} \frac{1}{2} \parallel \mathbf{Z} - \mathbf{D}\mathbf{A} \parallel^2 + \lambda \sum_{j=1}^{r} \Omega_{\mathbf{D}}(\mathbf{D}^{(j)}) \tag{6.4}$$

The choice of $\Omega_{\mathbf{D}}$, based on (Lee et al., 2007), is the ℓ_1-norm.

6.2.2 Structured Sparse Subspace Dictionary

This section discusses learning a structured sparse sub-manifold dictionary. It uses structured sparse principal component analysis (SSPCA), which is based on the use of structured regularization during selection of subspaces. Whereas classical regularization priors are concerned with cardinality of the selected subspaces, structured regularization incorporates higher level information. A conceptual illustration of this dictionary learning is shown in fig. 6.2. A feature vector $\mathbf{z} \in \mathbb{R}^8$ is projected to a sub-manifold $\mathcal{S}_{PCA} \in \mathbb{R}^6$ using PCA. The vector $\hat{\mathbf{z}}_{PCA}$ is not the true $\mathbf{z}$, which is embedded in some other sub-manifold of a different dimensionality than $\mathcal{S}_{PCA}$. SPCA is used to represent $\mathbf{z}$ using a $\mathbb{R}^3$-dimensional sub-manifold in $\mathbf{R}^6$. While SPCA, is able to estimate the true dimensionality of $\mathbf{z}$, the sub-spaces that SPCA selects are not semantically relevant. SSPCA computes $\hat{\mathbf{z}}_{SSPCA}$ by representing $\mathbf{z}$ using a semantically relevant sub-manifold $\mathcal{S}_{SSPCA}$. The subspaces that constitute $\mathcal{S}_{SSPCA}$ are sparse and semantically relevant simultaneously. This allows $\hat{\mathbf{z}}_{SSPCA}$ to be the best estimate of the true $\mathbf{z}$.

Similar to the formulation of SPCA, n feature vectors $\mathbf{z} \in \mathbb{R}^p$ are utilized to compute the dictionary $\mathbf{D}$ of r elements. The feature vectors are in matrix $\mathbf{Z} \in \mathbb{R}^{p \times n}$. The empirical risk of a dictionary element $\mathbf{d} \in \mathbb{R}^p$ for the n feature vectors is $\frac{1}{n}\sum_{i=1}^{n} L(\mathbf{z}_i, \mathbf{d}\alpha_i)$. Co-clustering is utilized to compute groups of subspaces $\mathcal{G}$, which was described in section 5.3. The groups account for all the subspaces of the descriptor, so $\cup_{G \in \mathcal{G}} G = \{1, \ldots, p\}$. The regularization term Ω using this group structure, based on (Jenatton et al., 2010), is

$$\begin{aligned} \Omega(\mathbf{d}) &= \textstyle\sum_{G \in \mathcal{G}} (\sum_{j \in G} (d_j^G)^2 |\mathbf{d}_j|^2)^{\frac{1}{2}} \\ &= \textstyle\sum_{G \in \mathcal{G}} \parallel d^G \circ \mathbf{d} \parallel_2 \end{aligned} \tag{6.5}$$

where the term $(d^G)_{G \in \mathcal{G}}$ serves as a indicator vector; a $|\mathcal{G}|$-tuple p-dimensional vector, such that $d_j^G > 0$ if $j \in G$ and $d_j^G = 0$ otherwise. The regularization problem is

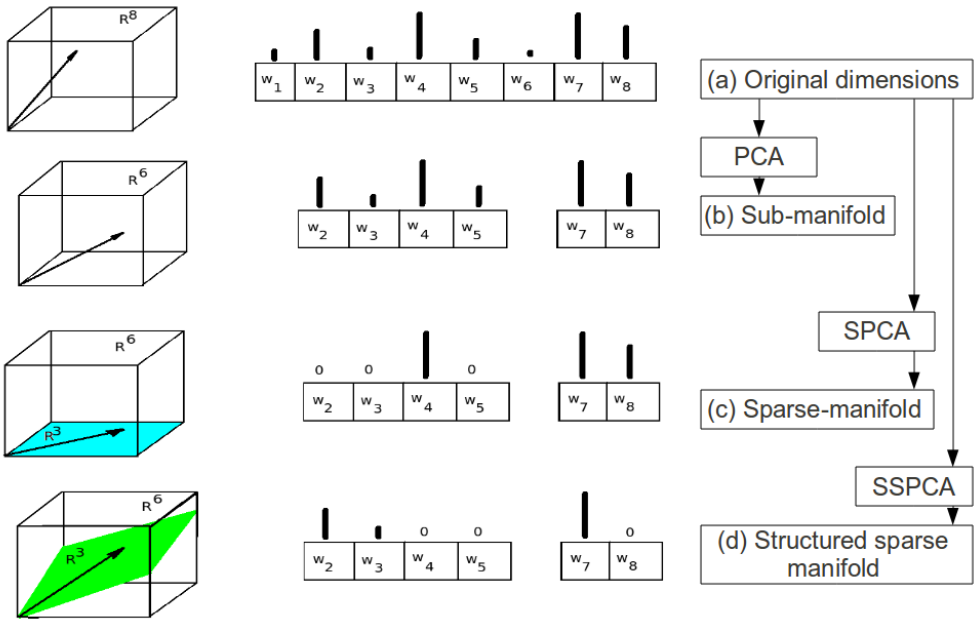

Figure 6.2: Conceptual illustration of structure sparse sub-manifold dictionary. A feature vector $\mathbf{z} \in \mathbb{R}^8$ is projected to a sub-manifold $\mathcal{S}_{PCA} \in \mathbb{R}^6$ using PCA. The vector $\hat{\mathbf{z}}_{PCA}$ is not the true $\mathbf{z}$, which is embedded in some other sub-manifold of a different dimensionality than $\mathcal{S}_{PCA}$. SPCA is used to represent $\mathbf{z}$ using a $\mathbb{R}^3$-dimensional sub-manifold in $\mathbf{R}^6$. While SPCA, is able to estimate the true dimensionality of $\mathbf{z}$, the sub-spaces that SPCA selects are not semantically relevant. Consequently, $\hat{\mathbf{z}}_{SPCA}$ is not the true $\mathbf{z}$. SSPCA computes $\hat{\mathbf{z}}_{SSPCA}$ by representing $\mathbf{z}$ using a semantically relevant sub-manifold $\mathcal{S}_{SSPCA}$. The subspaces that constitute $\mathcal{S}_{SSPCA}$ are sparse and semantically relevant simultaneously. This allows $\hat{mathbfz}_{SSPCA}$ to be the best estimate of the true $\mathbf{z}$.

$$\min \frac{1}{2} \parallel \mathbf{Z} - \mathbf{DA} \parallel^2 + \lambda \sum_{G \in \mathcal{G}} \parallel d^G \circ \mathbf{d} \parallel_2 \tag{6.6}$$

The regularization term $\Omega(\mathbf{d}) = \sum_{G \in \mathcal{G}} \parallel d^G \circ \mathbf{d} \parallel_2$ is a mixed $\ell_{2,1}$-norm (Zhao et al., 2009). At the group level, it functions like an ℓ_1-norm and consequently $\Omega(\cdot)$ induces group sparsity. In other words, each $d^G \circ \mathbf{d}$ is encouraged to be zero. In contrast, within the groups $G \in \mathcal{G}$, the ℓ_2-norm does not encourage sparsity. Intuitively, for a certain subset of groups $\mathcal{G}' \subseteq \mathcal{G}$, the vectors $\mathbf{d}_G$ associated with the groups $G \in \mathcal{G}'$ will be exactly equal to zero, leading to a set of zeros which is the union of these groups $\bigcup_{G \in \mathcal{G}'} G$. The set of allowed zero patterns should be the union-closure of $\mathcal{G}$: $\mathcal{Z} = \{\cup_{G \in \mathcal{G}'} G \; ; \; \mathcal{G}' \subseteq \mathcal{G}\}$. The complementary non-zero patterns are : $\mathcal{P} = \{\cap_{G \in \mathcal{G}'} G^c \; ; \; mathcal G' \subseteq \mathcal{G}\}$.

6.2.3 Evaluating Structured Sparse Subspace Dictionary

In the experiments in this section, the datasets used are VOC-2006, VOC-2007. The image feature descriptor used is dense-SIFT. In the dense sampling, the descriptor patch size was 8×8 pixels with a step size of 4 pixels. All the images were converted to grey scale. The implementation of SIFT in (Vedaldi and Fulkerson, 2008) was used. Similar to the experiments in previous chapters, the training and test set were the train and validation set provided for VOC-2006, and VOC-2007 in (Everingham et al., 2006), and (Everingham et al., 2007) respectively.

A random sample of size 100000 feature vectors is collated from the images in the training set. It is used to compute co-clusters using information-theoretic co-clustering and sum-squared residue co-clustering. The number of groups in this experiment is 50. The estimated groups are utilized as a priori structure for the group regularization term in SSPCA. The dictionary size is 1000, which is in keeping with previous experiments. The sparse subspace dictionary uses the ℓ_1-norm regularization while the structured sparse subspace dictionary uses the mixed $\ell_{2,1}$-norm regularization. The implementation of SPCA and SSPCA is based on (Jenatton et al., 2009). The optimization routine is run for a maximum of 1000 iterations. The regularization parameter is $\lambda = 10^{-8}$. The stopping criterion for change in error is empirically determined to be 10^{-12}. The optimization follows an alternative scheme between the dictionary and the coefficient matrices, optimizing each in turn while the other remains fixed. The number of iterations for each matrix is 5. Similar to previous experiments, the SVM classifier with RBF kernel is used in this experiment.

The graphs show the performance, in terms of mAP of classification for each category in each of the datasets. The results show that the structured sparse sub-manifold dictionary is better than the regular sparse sub-manifold dictionary for a majority, but not all, of the categories in each of the datasets.The results for VOC-2006 using ITCC is reported in fig. 6.3, where the structured sparse dictionary performs better for 6 of the 10 categories. Results using SS-

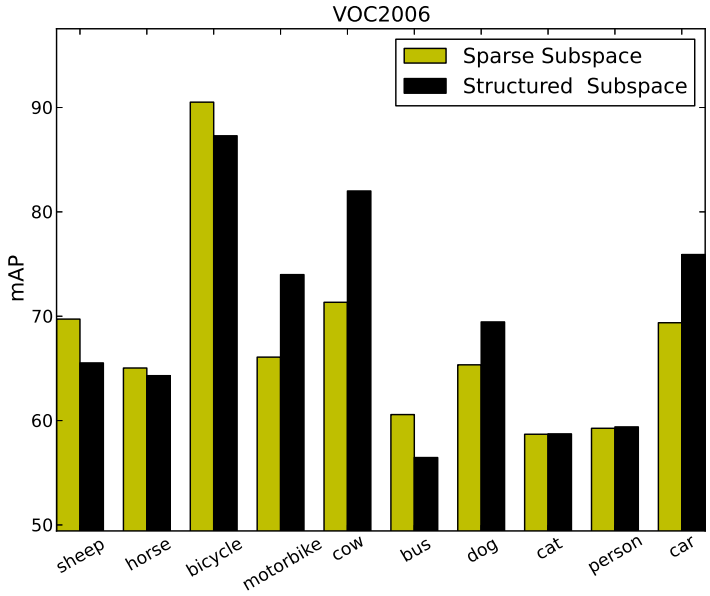

Figure 6.3: Comparative performance of Sparse Subspace and Structured Sparse Subspace dictionaries, for visual categories of VOC 2006 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using ITCC. The number of groups is 50.

RCC is reported in fig. 6.4, where the structured sparse dictionary performs better for 5 categories. Similarly, the results for VOC-2007 using ITCC is shown in fig. 6.5, where structured sparse dictionary is better for 14 of 20 categories. In the results using SSRCC is shown in fig. 6.6, the structured sparse dictionary is better for 10 categories. The variation in performance between categories is indicative of the inherent difference in difficulty of the categories. In addition, the number of groups of sub-spaces is the same for all categories, which contributes to the performance variation. In view of the inherent difference between categories, the number of groups should be tuned to be specific to each category.

Aggregate performance across datasets

To analyse the comparative performance of the sparse and structured sparse sub-manifold dictionaries, the aggregate classification performance across datasets is computed. The aggregate performance of each dataset is the mean of the classification score for all the categories in that dataset. This experimental

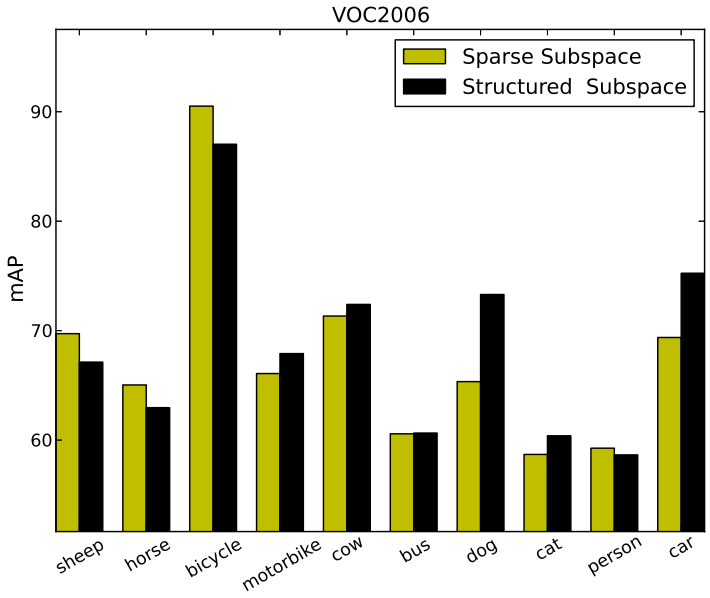

Figure 6.4: Comparative performance of Sparse Subspace and Structured Sparse Submanifold dictionaries, for visual categories of VOC 2006 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using SSRCC. The number of groups is 50.

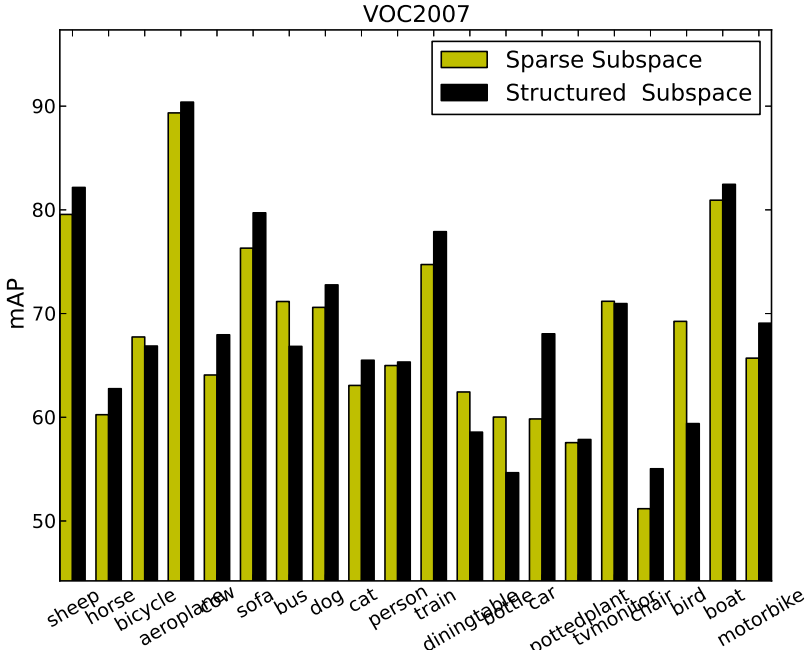

Figure 6.5: Comparative performance of Sparse Subspace and Structured Sparse Submanifold dictionaries, for visual categories of VOC 2007 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using ITCC. The number of groups is 50.

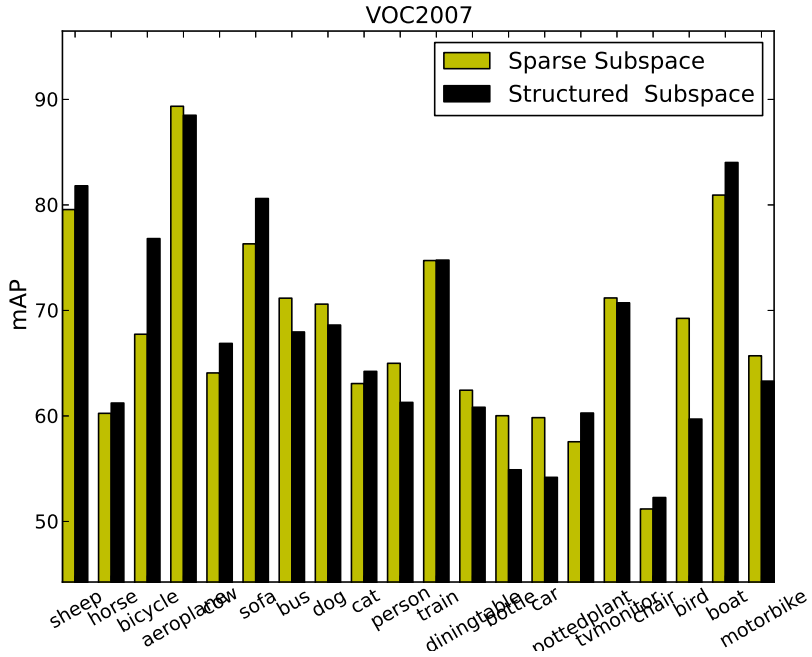

Figure 6.6: Comparative performance of Sparse Subspace and Structured Sparse Subspace dictionaries, for visual categories of VOC 2007 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using SSRCC. The number of groups is 50.

setup is the same as the previous experiment. The number of groups of subspaces is 50. The results for the VOC-2006 and VOC-2007 datasets is shown in table 6.1. The table shows the mean classification performance over all categories in each of VOC2006 and VOC2007 datasets. The results of structured sparse subspace dictionary using both ITCC and SSRCC is reported. The best performance for each dataset is shown in bold text. The structure sparse subspace dictionary provides a better result for both the datasets. However, for VOC2006 ITCC is comparatively better than SSRCC, whereas for VOC2007 the converse is true. The results are interesting first in that they show structure sparse dictionary has succeeded in incorporating semantic structure into the sub-manifolds and performed better than regular sparse subspace dictionary, and second in that the co-clustering method has a different performance for different datasets. This result emphasizes the need to use a semantic structure estimation technique that is tailored to the data distribution characteristic of a dataset. This thesis considered the K-L divergence (ITCC) and Euclidean

divergence (SSRCC), but there are several popular divergences in the family of Bregman divergences that could be better alternatives. Further exploration of the family of Bregman divergences and associated co-clustering technique is left for future work.

	Sparse Subspace	Structured Sparse Subspace	
Data Set		*ITCC*	*SSRCC*
VOC2006	67.5941	**70.8295**	68.5808
VOC2007	67.9971	68.0783	**68.3718**

Table 6.1: Aggregate classification performance of Sparse Subspace and Structured Sparse Subspace dictionaries for VOC2006 and VOC2007 datasets. The table shows the mean classification performance over all categories in each of VOC2006 and VOC2007 datasets. The results of structured sparse subspace dictionary using both ITCC and SSRCC is reported. The best performance for each dataset is shown in bold text.

6.3 Structured Sparse Encoding

There are two key insights that motivate the development of structured sparse encoding. The experiments in section 5.4 indicated the existence of structure in the dictionary, in terms of groups of semantically related dictionary elements. The second insight is that feature vector distribution is idiomatic to a visual category. The experiments in section 5.3.3 and section 5.4.2 indicate that feature vectors associated with a visual category are predominantly located in a group of disjoint regions of feature space. To model such a distribution requires a parsimonious encoding scheme that can learn to represent an image using the appropriate disjoint regions only. Sparse coding methods can potentially achieve this model, provided the number of related disjoint partitions are known a priori and these appropriate partitions are selected. Sparsity is induced by use of a regularization constraint on the learnt coefficients. The choice of regularization scheme is important for the type of sparsity achieved, which is elaborated upon in section 6.3.1.

Notwithstanding the success of sparse regularization methods, there is an unresolved issue. Sparse models succeed in selecting the appropriate number of

dictionary elements, but minimization of the loss function alone is not sufficient to guarantee selection of the appropriate subset of dictionary elements. In addition, the result of the optimization is unstable, since a small perturbation in the optimization routine could yield a different subset of dictionary elements for the representation of the same image. The reason for this issue is that these methods treat each dictionary element individually, so existing relationships and structures between these dictionary elements is disregarded. In order to reduce the instability in the selection, a potential solution is grouping dictionary elements. An appropriate grouping would consist of semantically related dictionary elements. This encourages the representation of an image using only those dictionary elements that are semantically relevant for that image. The method to achieve this structured selection of dictionary elements is discussed in section 6.3.2, following an overview of regular sparse coding.

6.3.1 Sparse Coding

Based on (Aharon et al., 2006), for training feature vectors $\mathbf{Z} \in \mathbb{R}^{p \times n}$, sparse coding with the ℓ_1-norm solves the problem

$$\min_{\alpha \in \mathbb{R}^p} \| \mathbf{Z} - \mathbf{DA} \|^2 + \lambda \| \mathbf{A} \|_1 \tag{6.7}$$

where λ is a regularization parameter. During dictionary learning this is a joint optimization problem with respect to dictionary $\mathbf{D}$ and the sparse coefficients $\mathbf{A} = \{\alpha^1, \ldots, \alpha^r\} \in \mathbb{R}^{p \times r}$. To prevent $\mathbf{D}$ from attaining arbitrarily large values, the columns of $\mathbf{D}$ are constrained to have an ℓ_2-norm less than a specified threshold - typically $\| \mathbf{d} \|_2 \leq 1$. The canonical strategy for the joint optimization is to alternate between the two variables; keeping one fixed while minimizing the other.

The regularization problem in eq. (6.7) is solved to compute a dictionary matrix $\mathbf{D}$ or coefficient matrix $\mathbf{A}$. When learning a visual model, sparse coding is typically used to first compute a dictionary, where $\mathbf{Z}$ is random sample of train-

ing descriptors for the dataset (Donoho and Stodden, 2003; Rigamonti et al., 2011). The learnt dictionary is used next to compute encoded representations of an image by computing $\mathbf{A}$, where $\mathbf{Z}$ is set of descriptors from that image (Coates et al., 2011b; Guillermo and Sprechmann, 2010; Kreutz-Delgado et al., 2003). λ is a non-negative parameter controlling the trade-off between data fitting and regularization (Lee et al., 2009). A typical convex formulation is the ℓ_1-decomposition problem (Donoho and Stodden, 2003), also known as the Lasso (Tibshirani, 1994):

$$min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} \parallel \mathbf{z}_i - \mathbf{D}\alpha \parallel_2^2 + \Omega(\alpha) \tag{6.8}$$

When the value of λ is large, α is known to be sparse, and only a few dictionary elements are involved. Although the use of ℓ_1-norm is predominant, a natural choice would be to take the ℓ_0 pseudo-norm that counts the number of non-zero coefficients in α. Conceptually, an image should be represented by a subset of the dictionary elements. In other words, those dictionary elements that do not pertain to a visual category should not be involved in the representation of the coefficient matrix $\mathbf{A}$. However, solving the equation above in this setting is often intractable, such that one has either to look for an approximate solution using a greedy algorithm, or resort to a convex relaxation (Gregor and Lecun, 2010).

A method closely related to Lasso was proposed by Efron et al. (2004) called Least Angle Regression Selection (LARS). Both these methods operate in two stages. The first step is to build a solution path indexed by a certain tuning parameter. In the next step the model is selected on the solution path by either cross-validation or use of some criterion. The solution paths for both Lasso and LARS are shown in (Efron et al., 2004) to be piecewise linear, which renders them computationally efficient. A comparative analysis of this was provided by Rosset and Zhu (2007).

6.3.2 Group Sparse Coding

There are a couple of issues with the Lasso and LARS methods. Although they have excellent performance and computational efficiency, they are designed for selecting individual dictionary elements. They can not be utilized for general factor selection in eq. (6.7). In other words they are transparent to structure in the dictionary. So, they tend to make selections based on the strength of individual derived dictionary element rather than the strength of groups of dictionary elements (or topics), often resulting in selecting more factors than necessary. Another drawback of using the Lasso and LARS in eq. (6.7) is the dependency of the solution upon how the factors are orthogonalised. This is undesirable since the solution to a factor selection and estimation problem should not depend on how the factors are represented (Yuan and Lin, 2006). Evidently these methods require an extension that considers the dictionary elements with a pre-defined structure. The group Lasso and group LARS (Bach, 2008; Yuan and Lin, 2006) are popular choices for this extension.

In a nutshell, the dictionary elements are assumed to be clustered in groups, and instead of summing the absolute values of each individual coefficient, the sum of Euclidean norms of the coefficients in each group is used. Intuitively, this should drive all the coefficients in one group to zero together, and thus lead to group selection (Yuan and Lin, 2006). In other words, the group Lasso essentially replaces groups of size one by groups of size larger than one.

Following from section 6.3.1 and based on (Yuan and Lin, 2006), the loss function for group Lasso augments eq. (6.7) with the additional inclusion of group information of the dictionary elements. The feature vectors $\{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ are in matrix $\mathbf{Z} \in \mathbb{R}^{p \times n}$. A dictionary $\mathbf{D} \in \mathbb{R}^{p \times r}$ for size r has been learnt by one of the techniques discussed in section 2.3. The elements of $\mathbf{D}$ are grouped into k groups, using co-clustering as discussed in section 5.4.1. The number of elements in each group are $\{t_1, \ldots, t_k\}$, where $\sum_{i=1}^{k} t_i = r$. As in previous section, the coefficient matrix is $\mathbf{A} \in \mathbb{R}^{r \times n}$. Consider positive definite matrices $\{K_1, \ldots, K_k\}$ corresponding to the k groups in the dictionary. For feature vec-

tors $\mathbf{Z}$, dictionary $\mathbf{D}$, and coefficients $\mathbf{A}$, group Lasso solves the regularization problem

$$\min \frac{1}{2} \parallel \mathbf{Z} - \sum_{j=1}^{k} \mathbf{D}_j \mathbf{A}_j \parallel_2^2 + \lambda \sum_{j=1}^{k} \parallel \mathbf{A}_j \parallel_{K_j} \tag{6.9}$$

Adopting a similar formulation to (Lin and Zhang, 2006), $\mathbf{D}_j$ and K_j are chosen respectively to be the group of dictionary elements and the reproducing kernel of the functional space induced by the j^{th} group. Note that eq. (6.9) does have a degenerate expression as Lasso in eq. (6.7) when all the groups contain a single dictionary element, so the cardinality of each topic is one, i.e. $t_j = 1 \forall j \in [\![1; k]\!]$. The regularization constraint here lies between ℓ_1, corresponding to Lasso, and ℓ_2, corresponding to ridge regression. To visualize this consider fig. 6.7, which shows the geometric representation of the group regularization term $\Omega = \sum_{j=1}^{k} \parallel \mathbf{A}_j \parallel_{K_j}$. As an illustrative example, consider the case where there are two groups, $k = 2$. The corresponding coefficients are $\mathbf{A}_1$ and $\mathbf{A}_2 = (\mathbf{A}_{11}, \mathbf{A}_{12})^T$. In fig. 6.7 the double pyramid shaped structure in *(a)* corresponds to ℓ_1-norm, so $|\mathbf{A}_1|+|\mathbf{A}_{21}|+|\mathbf{A}_{22}| = 1$, which is the Lasso regression. The sphere shaped structure in *(c)* corresponds to ℓ_2-norm, so $\parallel (\mathbf{A}_1, \mathbf{A}_2) \parallel = 1$, which is the ridge regression. The bi-cone shaped structure in *(b)* corresponds to $\ell_{2,1}$-norm, so $|\mathbf{A}_1| + \parallel \mathbf{A}_2 \parallel = 1$, which is the group Lasso. The ℓ_1 considers the coefficients individually and induces sparsity in them separately. The ℓ_2-norm also considers the coefficients equally, but does not encourage sparsity. In contrast to both of these norms, the essence of $\ell_{2,1}$-norm is evident in the bi-conic geometric structure. The coefficients $\mathbf{A}_{21}$ and $\mathbf{A}_{22}$ are treated equally with ℓ_2-norm applied within the group. The ℓ_1-norm is used between the two groups. In other words, sparsity will be induced on $\mathbf{A}_1$ or the pair $(\mathbf{A}_{21}, \mathbf{A}_{22})$.

6.3.3 Evaluating Structured Sparse Encoding

In this experiment the Scene-15 dataset is used. Dense SIFT feature descriptors are generated following the procedure described in the previous experiment. A

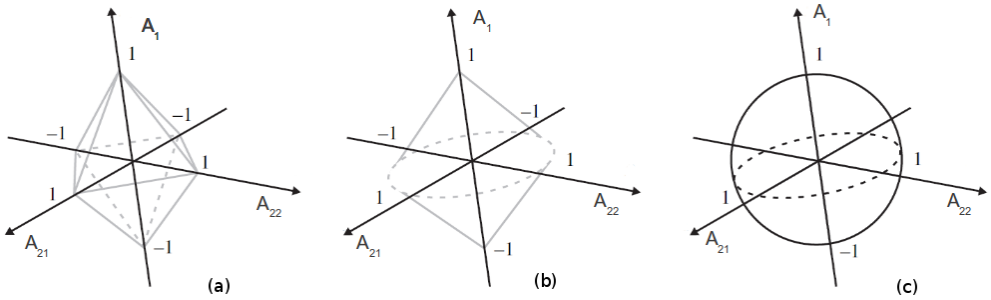

Figure 6.7: Pictorial representation of $\Omega = \sum_{j=1}^{k} \| \mathbf{A}_j \|_{K_j}$. *In this example there are* $k = 2$ *groups of coefficients* $\mathbf{A}_1$ *and* $\mathbf{A}_2$. *(a)* ℓ_1*-norm and Lasso regression, where* $|\mathbf{A}_1| + |\mathbf{A}_{21}| + |\mathbf{A}_{22}| = 1$. *(b)* $\ell_{2,1}$*-norm and group Lasso regression, where* $|\mathbf{A}_1| + \| \mathbf{A}_2 \| = 1$. *(c)* ℓ_2*-norm and ridge regression, where* $\| (\mathbf{A}_1, \mathbf{A}_2) \| = 1$.

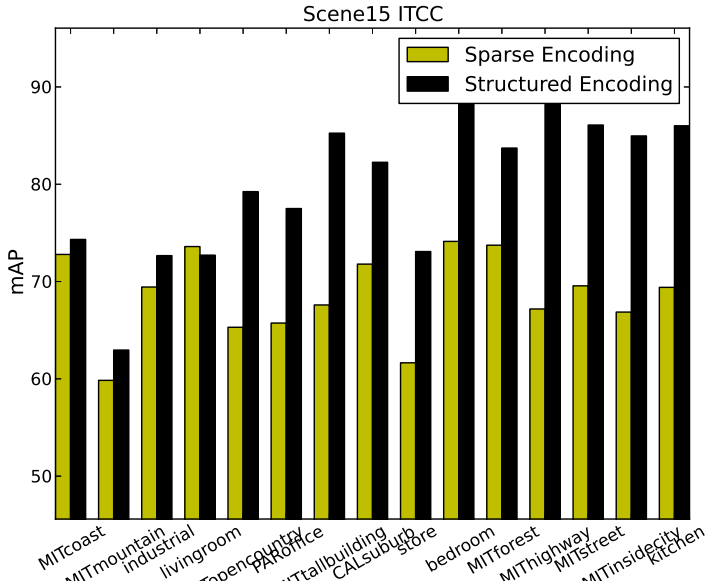

Figure 6.8: Comparative performance of Sparse Encoding and Structured Sparse Encoding, for visual categories of Scene-15 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using information theoretic co-clustering. The number of groups is 100 *from a* 1000 *element dictionary.*

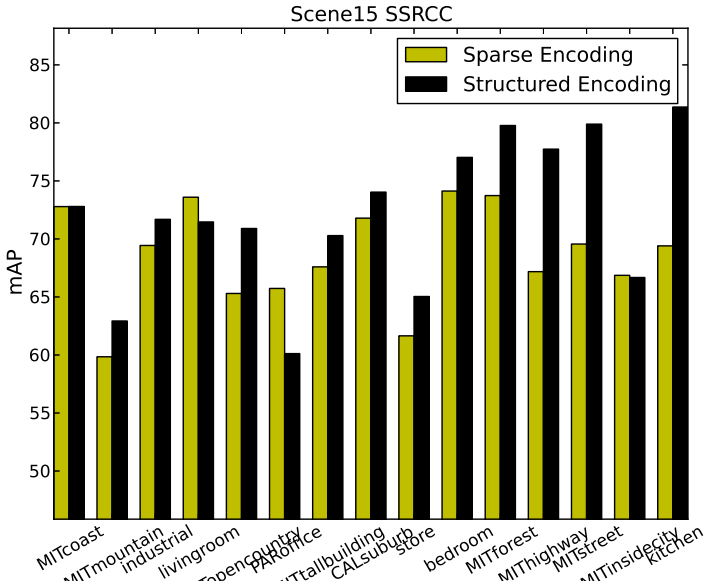

Figure 6.9: Comparative performance of Sparse Encoding and Structured Sparse Encoding, for visual categories of Scene-15 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using sum-squared residue co-clustering. The number of groups is 100 *from a* 1000 *element dictionary.*

random sample of 100000 feature vectors from all the images in the training set of every visual category for each dataset is used here. It is used to compute a dictionary of size 1000, using k-means which is run for a maximum of 100 iterations. In order to compare the sparse and structured sparse encoding schemes the choice of dictionary is based on the objective of not biasing the dictionary towards either of the encoding methods. The information-theoretic and sum-squared residue co-clustering methods are used to estimate topics of size 100. The procedure for using co-clustering has been described in the experiments in chapter 5. Sparse encoding uses the Lasso while the structured sparse uses the group Lasso. The choice of regularization parameter is $\lambda = 0.1$, based on preliminary analysis of an appropriate λ for the data used in this experiment. The optimization routine is run for a maximum of 200 iterations for encoding each image in the training and test sets. The choice of iterations here was based on the dual objective of allowing convergences a satisfactory encoded representation and curtailing the total time required to encode thousands of training and test images. The classifier used is a SVM with RBF kernel and the performance

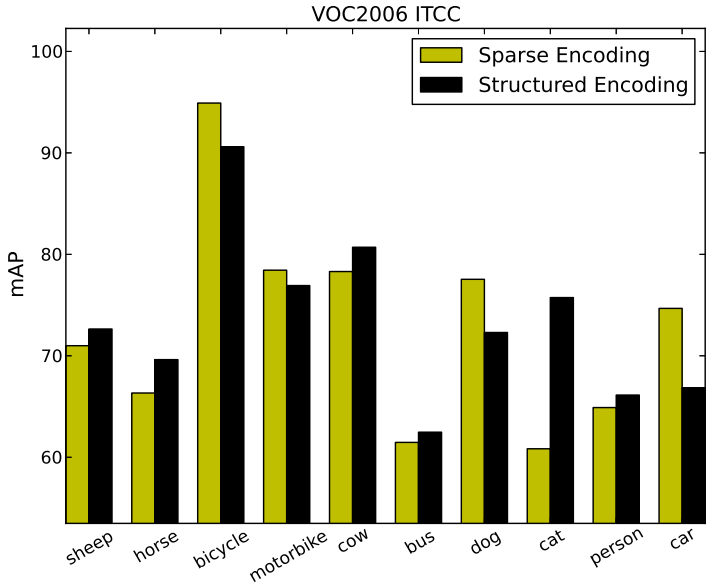

Figure 6.10: Comparative performance of Sparse Encoding and Structured Sparse Encoding, for visual categories of VOC-2006 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using information theoretic co-clustering. The number of groups is 100 *from a* 1000 *element dictionary.*

is reported as mAP.

	Sparse Encoding	Structured Sparse Encoding	
Data Set		*ITCC*	*SSRCC*
VOC-2006	72.8386	**73.3977**	72.7738
Scene-15	68.5737	**79.8794**	72.1155

Table 6.2: Aggregate classification performance of sparse encoding and structured sparse encoding for the VOC-2006 and Scene-15 datasets. The best performance for both datasets is shown in bold text. Structured sparse encoding using ITCC topic dictionary has the best comparative performance.

The results for Scene-15 dataset using information-theoretic co-clustering and sum-squared residue co-clustering are reported in fig. 6.8 and fig. 6.9 respectively. Structured sparse encoding performs better for 14 of 15 categories using ITCC and 11 of 15 categories using SSRCC. Similarly, the results for VOC-2006 dataset using ITCC and SSRCC are reported in fig. 6.10 and fig. 6.11 respectively, where ITCC performs better for 6 of 10 categories and SSRCC performs better for 6 of 10 categories. With regards to the VOC-2006 dataset, structured

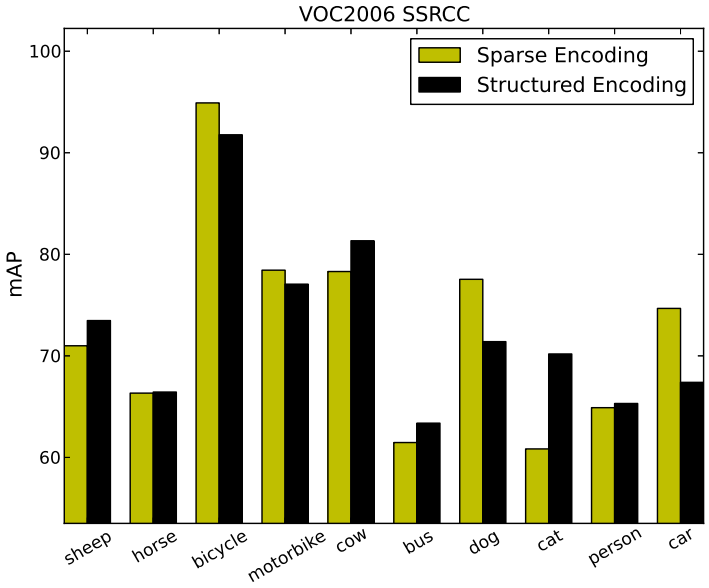

Figure 6.11: Comparative performance of Sparse Encoding and Structured Sparse Encoding, for visual categories of VOC-2006 dataset. The graph shows the classification score measured as mAP for visual categories. The structure here is estimated using sum-squared residue co-clustering. The number of groups is 100 *from a* 1000 *element dictionary.*

sparse encoding has a performance comparable to regular sparse encoding. It is better for six of the ten categories in the dataset for both ITCC and SSRCC. For the Scene-15 dataset, the structured sparse encoding method has a better performance in comparison to sparse encoding for almost all of the categories. This is an encouraging result for structured learning. The aggregate performance for all categories in the dataset is reported in table 6.2 to compare between the co-clustering methods. This is a positive result for structured sparse encoding, as the aggregate classification performance using ITCC is the best for both datasets.

These results can be interpreted as an encouraging outcome for structured learning algorithms. It should be noted that performance of structured sparse encoding depends upon several factors, including the $\ell_{1,2}$-norm regularization, the quality of the estimated groups of dictionary elements, the clustering algorithm which partitioned feature space. The $\ell_{1,2}$-norm is only one of the family of $\ell_{1,q}$-norm that could be used for enforcing group sparsity. It remains to be found if some value of q other than 2 may yield a better structured encoding. Another

aspect is the dictionary size, which was kept the same across datasets. Since different datasets have different visual content, it is reasonable to expect that the appropriate dictionary size for different datasets would be different. This is one contributing factor for difference between the VOC-2006 and Scene-15 datasets.

6.4 Summary

In this chapter, semantically related groups of basis elements estimated by co-clustering were used to augment sparsity inducing visual models with a priori semantic information.

The goal of the structured sparse subspace dictionary was to successfully incorporate the groups of subspaces estimated by the co-clustering methods to produce a dictionary that performs better than a sparse subspace dictionary without a priori structural information. The comparative performance for multiple VOC datasets in the experiments in section 6.2.3 indicate that the goal has been achieved. There are some visual categories where the performance of SPCA is better. However, this result is expected in view of the significant difference between the categories in terms of inherent complexity and availability of training data to learn these complexities equally for all categories.

The aim of structured sparse encoding was to leverage the groups of dictionary elements estimated by co-clustering to encourage sparsity in the encoding of images in terms of semantically relevant topics rather than individual words. It was compared to sparse coding using ℓ_1-norm, which is an excellent performing scheme. In the experiments in section 6.3.3 the performance of the structured sparse encoding is marginally better than regular sparse coding when considering aggregate performance. Amongst the visual categories the comparative performance is not entirely conclusive. It should be noted that the performance of structured sparse model depends upon: the size of the dictionary and its ability to appropriately partition feature space; the presence of semantically distinct groups; and the efficacy of the co-clustering approach in estimating

these groups. The presence of distinct groups is a weak hypothesis in view of the available weakly labelled data and relatively small training data. Therefore the results of the experiments on group structured dictionary should be considered as indicative, rather than conclusive evidence for this approach.

In the results, the performance of each category is different from other categories regardless of the method used. This suggests an inherent difference in complexity associated with each category. Of course this is not an unexpected result. One contributing factor is the tradition in the community to use the same number of training images for all categories. It is reasonable to assume that categories with a comparatively higher degree of complexity will require more training images. The number of training images has a strong bearing on the performance that can be expected from a method designed to estimate semantically relevant structure. There should be sufficient training images that span the variation in appearance in a category to allow a learning method that opportunity to learn the semantic structure. Therefore the margin of improvement of structured sparse visual model over regular sparse visual model could be attributed to lack of sufficient training data. This possibility does not itself qualify structured learning. Nevertheless the results of the experiments in chapter 5 and in chapter 6 demonstrate that the methods here have succeeded in estimating semantic structure.