Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2008-009 February 17, 2008

Unsupervised Distributed Feature Selection for Multi-view Object Recognition

C. Mario Christoudias, Raquel Urtasun, and Trevor Darrell

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

CSAIL

Unsupervised Distributed Feature Selection for Multi-view Object Recognition

C. Mario Christoudias, Raquel Urtasun, and Trevor Darrell

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge MA, 02139, USA
{cmch, rurtasun, trevor}@csail.mit.edu

Abstract

Object recognition accuracy can be improved when information from multiple views is integrated, but information in each view can often be highly redundant. We consider the problem of distributed object recognition or indexing from multiple cameras, where the computational power available at each camera sensor is limited and communication between sensors is prohibitively expensive. In this scenario, it is desirable to avoid sending redundant visual features from multiple views, but traditional supervised feature selection approaches are inapplicable as the class label is unknown at the camera. In this paper we propose an unsupervised multi-view feature selection algorithm based on a distributed compression approach. With our method, a Gaussian Process model of the joint view statistics is used at the receiver to obtain a joint encoding of the views *without* directly sharing information across encoders. We demonstrate our approach on recognition and indexing tasks with multi-view image databases and show that our method compares favorably to an independent encoding of the features from each camera.

1 Introduction

Object recognition or indexing from multiple views usually offers increased performance when compared to a single view. However, when multiple camera sensors exist in a bandwidth limited environment it may be impossible to transmit all the visual features in each image, and when the task or target class is not known *a priori* there may be no obvious way to decide which features to send from each view. If redundant features are chosen at the expense of informative features, performance can be worse with multiple views than with a single view, for a fixed bandwidth.

We consider the problem of how to select which features to send in each view to achieve optimal results at a centralized recognition or indexing module (see Figure 1). An efficient encoding of the streams might be possible in theory if a class label could be inferred at each sensor, enabling the use of supervised feature selection techniques to encode and send only those features that are relevant of that class. Partial occlusions, unknown camera viewpoint, and limited computational power, however, limit the ability to reliably estimate the image class label at a single sensor. Instead we propose an unsupervised feature selection algorithm to obtain an efficient encoding of the feature streams.

If each camera sensor had access to the information from all views this could trivially be accomplished by a joint compression algorithm that could, e.g., encode the features of the v-th image based on the information in the previous $v-1$ images. We are interested, however, in the case where there is *no* communication between sensors themselves, and messages are only sent from the cameras to the recognition module with a limited backchannel back to the cameras. In practice, many visual category recognition and indexing applications are bandwidth constrained (e.g., wireless surveillance camera networks, mobile robot swarms, mobile phone cameras), and it is infeasible to broadcast images across all sensors or to send the raw signal from each sensor to the recognition module.

Surprisingly, it is possible to achieve very efficient encoding without any information exchange between the sensors, by adopting a distributed encoding scheme that takes advantage of known statistics of the environment [14, 20, 18, 3]. We develop a new method for distributed encoding based on a Gaussian Process (GP) formulation, and demonstrate its applicability to encoding visual-word feature histograms; such representations are used in many contemporary object indexing and category recognition methods [19, 12, 5].

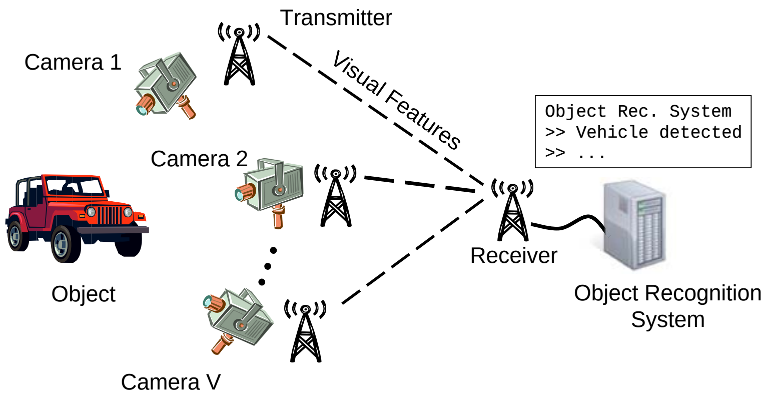


Figure 1: Distributed object recognition. Messages are only sent between the cameras and the recognition module, we presume *no* direct communication between cameras.

Our algorithm exploits redundancy between views and learns a statistical model of the dependency between feature streams during an off-line training phase at the receiver. This model is then used along with previously decoded streams to aid feature selection at each sensor. If the streams are redundant, then only a few features need to be sent. As shown in our experiments, our algorithm is able to achieve an efficient joint encoding of the feature streams without explicitly sharing features across views. This results in an efficient unsupervised feature selection algorithm that improves recognition performance in the presence of limited network bandwidth.

The use of local image features for object recognition has become a standard practice in the vision community. Both individual-feature and feature-histogram distributed encoding regimes are theoretically possible, but our experiments have shown that distributed coding of individual features does not improve over independent encoding at each sensor, while distributed coding of feature histograms does offer a significant advantage. This is expected since a local image feature is a fairly weak predictor of other features in the image.

We evaluate our approach using the COIL-100 multi-view image database [11] on the task of instance-level recognition from multiple views; we compare unsupervised distributed feature selection to independent stream encoding. For a two-view problem, our algorithm achieves a compression factor of over $100 : 1$ in the second view while preserving multi-view recognition accuracy. In contrast, independent encoding at the same rate does not improve over single-view performance.

2 Related Work

Contemporary methods for object recognition use local feature representations and perform recognition over sets of local features corresponding to each image [8, 12, 5, 22]. Several techniques have been proposed that generalize these methods to include object view-point in addition to appearance [16, 21, 22, 17]. Rothganger et. al. present an approach that builds an explicit 3D model from local affine-invariant image features and uses that model to perform view-point invariant object recognition [16]. Thomas et. al. [21] extend the Implicit Shape Model (ISM) of Leibe and Schiele [8] for single-view object recognition to multiple views by combining the ISM model with the recognition approach of Ferrari et. al. [2]. Similarly, Savarese and Li [17] present a part-based approach for multi-view recognition that jointly models object view-point and appearance.

Traditionally, approaches to multi-view object recognition use only a single input image at test time [16, 21, 22, 17]. Recently, there has been a growing interest in application areas where multiple input views of the object or scene are available. The presence of multiple views can lead to increased recognition; however, the transmission of content rich image data from multiple cameras places a burden on the network, and this is especially difficult in the case of power and bandwidth contraints. In this paper, we propose an unsupervised feature selection algorithm that enables effective object recognition from multiple cameras in the presence of limited network bandwidth.

Feature selection algorithms exploit data dependency or redundancy to derive compact representations for classification [9]. For our problem, traditional supervised feature selection approaches are inapplicable as the class label is unknown at each camera. Many approaches have been proposed for unsupervised feature selection [1, 9, 13]. Peng et. al. [13] define a minimum-redundancy or maximum-relevance criterion for unsupervised feature selection based on mutual information. Dy and Brodley [1] compute relevant feature subsets using a clustering approach based on a maximum likelihood criterion with the expectation maximization algorithm. For multiple views, it is possible to apply the above unsupervised feature selection techniques independently at each camera to efficiently encode and transmit features over the network. A better encoding of the features, however, can be achieved if features are jointly selected across views [20]. Under limited network bandwidth a joint encoding is not possible as communication between sensors is often prohibitively expensive.

Distributed coding algorithms [14, 20, 18, 3] seek a joint encoding of the streams *without* sharing features across views. These methods exploit data redundancy at a shared, common receiver to perform a distributed feature selection that in many cases approaches the joint encoding rate [20]. Contemporary techniques to distributed coding include the DISCUS algorithm of Pradhan and Ramchandran [14] based on data cosets, and the approach of Schonberg [18] that builds upon low-density parity check codes. In this paper, we present a new distributed coding algorithm for bag-of-words image representations in multi-view object recognition with Gaussian Processes.

Gaussian Processes (GPs) [15] have become popular because they are simple to implement, flexible (i.e., they can capture complex behaviors through a simple parametrization), and are fully probabilistic. The latter enables them to be easily incorporated in more complex systems, and provides an easy way of expressing and evaluating prediction uncertainty. GPs have been suggested as a replacement for supervised neural networks in non-linear regression [15] and they generalize a range of previous techniques (e.g. kriging, splines, RBFs). As shown in our experiments, GPs are well suited for distributed feature selection as the uncertainty measure provided by GPs is correlated with data redundancy.

The remainder of this paper is organized as follows. A brief review of GPs is given in the next section. We then provide a formal description of our problem and present our GP-based distributed feature selection algorithm in Section 4. Experiments and results are discussed in Section 5. Finally, in Section 6 we provide concluding remarks and a discussion of future work.

3 Gaussian Process Review

A Gaussian Process is a collection of random variables, any finite number of which have consistent joint Gaussian distributions [15]. Given a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \cdots, N\}$, composed of inputs $\mathbf{x}_i$ and noisy outputs $\mathbf{y}_i$, we assume that the noise is additive, independent and Gaussian, such that the relationship between the (latent) function, $f(\mathbf{x})$, and the observed noisy targets, $\mathbf{y}$, is given by

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon_i \ , \tag{1}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2_{noise})$ and σ^2_{noise} is the noise variance.

GP regression is a Bayesian approach that assumes a GP prior over the space of functions,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(0, \mathbf{K}) \ , \tag{2}$$

where $\mathbf{f} = [f_1, \cdots, f_n]$ is the vector of latent function values, $f_i = f(\mathbf{x}_i)$, $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$, and $\mathbf{K}$ is a covariance matrix whose entries are given by a covariance function, $K_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$. GPs are non-parametric models and are entirely defined by their covariance function (and training data); the set of possible covariance functions is defined by the set of Mercer kernels. During training, the model hyper-parameters, $\bar{\beta}$, are learned by minimizing

$$-\ln p(\mathbf{X}, \bar{\beta} \,|\, \mathbf{Y}) = \frac{D}{2} \ln |\mathbf{K}| \ + \ \frac{1}{2} \mathrm{tr}\left(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T\right) + C \ . \tag{3}$$

where $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_N]$, C is a constant, and D is the dimension of the output.

Inference in the GP model is straightforward, assuming a joint GP prior over training, $\mathbf{f}$, and testing, $\mathbf{f}_*$, latent variables,

$$p(\mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(0, \begin{pmatrix} \mathbf{K}_{f,f} & \mathbf{K}_{*,f} \\ \mathbf{K}_{f,*} & \mathbf{K}_{*,*} \end{pmatrix}\right) \ , \tag{4}$$

where $*$ is used as shorthand for f_* and the dependency on $\mathbf{X}$ is omitted for clarity of presentation, $\mathbf{K}_{f,f}$ is the covariance of the training data, $\mathbf{K}_{*,*}$, the covariance of the test data, and $\mathbf{K}_{f,*} = \mathbf{K}^T_{*,f}$ is the cross-covariance

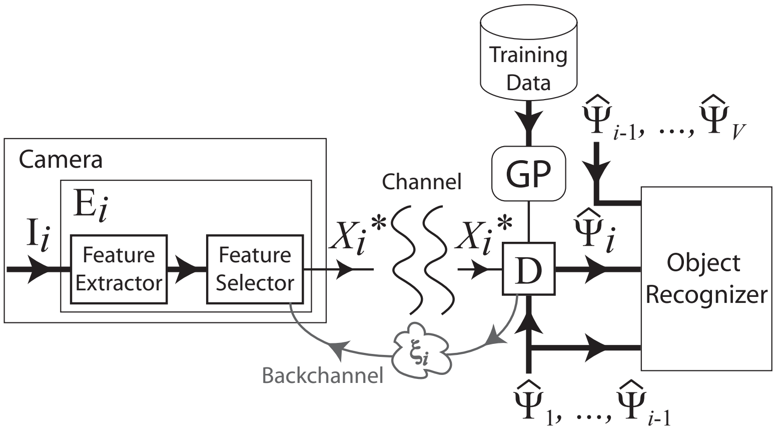


Figure 2: System diagram. Image I_i is coded by encoder E_i and decoder D. X_i^* are the encoded image features, $\hat{\Psi}_i$ the reconstructed histograms, and ξ_i the non-redundant bin indices for view i, $i = 1, ..., V$ (see Section 4 for details).

of training and test data. The joint posterior $p(\mathbf{f}, \mathbf{f}_*|\mathbf{Y})$ is Gaussian:

$$p(\mathbf{f}, \mathbf{f}_*|\mathbf{Y}) = \frac{p(\mathbf{f}, \mathbf{f}_*)p(\mathbf{y}|\mathbf{f})}{p(\mathbf{y})} . \tag{5}$$

Marginalizing the training latent variables, $\mathbf{f}$, can be done in closed form and yields a Gaussian predictive distribution [15], $p(\mathbf{f}_*|\mathbf{y}) = \mathcal{N}(\mathbf{M}, \mathbf{C})$, with

$$\mathbf{M} = \mathbf{K}_{*,f}(\mathbf{K}_{f,f} + \sigma^2_{noise}\mathbf{I})^{-1}\mathbf{Y} \tag{6}$$

$$\mathbf{C} = \mathbf{K}_{*,*} - \mathbf{K}_{*,f}(\mathbf{K}_{f,f} + \sigma^2_{noise}\mathbf{I})^{-1}\mathbf{K}_{f,*} . \tag{7}$$

The variance of the Gaussian Process is an indicator of the prediction uncertainty. In the following section we will show how the variance can be used to define a feature selection criteria.

4 Distributed Object Recognition

We consider the distributed recognition problem of V sensors transmitting information to a central, common receiver with no direct communication between sensors (see Figure 1). In our problem, each camera is equipped with a simple encoder used to compress each signal before transmission. A common decoder receives the encoded signals and performs a joint decoding of the signal streams using a model of joint statistics. Note that this coding scheme off-loads the computational burden onto the decoder and allows for computationally in-expensive encoders. In what follows, we assume a noiseless channel, but our approach is also applicable to the more general case.

Figure 2 illustrates our proposed distributed coding algorithm at a single sensor. With our algorithm, the decoder iteratively queries each of the V sensors and specifies the desired encoding rate the sensor should use. At the i-th sensor, the decoder uses its model of joint statistics along with *side information*, i.e., the previously decoded streams, to decode the signal. The use of side information allows the encoder to work at a lower encoding rate than if the stream were encoded independently. As discussed below, the decoder selects the sensor encoding rate based on the joint stream statistics and transmits this information back to the encoder. If the i-th signal is highly redundant with respect to the side information, then little-to-no information needs to be encoded and sent to the decoder.

In this work, we consider bag-of-words models for object recognition [12, 5]. With these models, an image is represented using a set of local image descriptors extracted from the image either at a set of interest point locations (e.g., those computed using a Harris point detector [10]) or on a regular grid. In our experiments, we employ the latter feature detection strategy in favor of simplicity at the encoder. To perform feature coding, the local image features are quantized using a global vocabulary that is shared by the encoder and decoder and computed from training images.

Let I_i, $i = 1, ..., V$ be a collection of V views of the object or scene of interest, imaged by each sensor and $X_i = \{x_i\}$ be the set of quantized local image features corresponding to image I_i computed by the

i-th encoder, E_i. In this context, the encoders of Figure 2 transmit quantized features to the central receiver and the encoding rate is the number of features sent. In the case of our distributed coding algorithm, this number is determined by the decoder, which uses a model of joint statistics in order to decide which features to request from each encoder, as described below.

In theory, distributed coding with individual image features (e.g., visual words) might be possible, but preliminary experiments have shown that distributed coding of local features does not improve over independent encoding at each sensor. Using a correlation table joint model over quantized features on COIL-100 with a 991 word vocabulary gave an entropy of 9.4 bits, which indicates that the joint feature distribution is close to uniform (for a 991 word feature vocabulary, the uniform distribution has an entropy of 10 bits). This is expected since a local image feature is a fairly weak predictor of other features in the image.

We have found, however, that distributed coding of histograms of local features is effective. As seen in our experiments, the distribution over features in one view is predictive of the distribution of features in other views and, therefore, feature histograms are a useful image representation for distributed coding. The use of a global image representation with feature histograms also avoids the many-to-one correspondence issue inherent in the local feature model.

4.1 Joint Feature Histogram Model

Let $\hat{X}_i = \{\hat{x_i}\}$ be the set of decoded features of each view, $i = 1, ..., n$. To facilitate a joint decoding of the feature streams, the decoder first computes a feature histogram, $\hat{\Psi}^i = \Psi(\hat{X}_i)$, using the global feature vocabulary. Note, in our approach, the form of $\Psi(\cdot)$ can either be a flat [19] or hierarchical histogram [12, 5] as we present a general distributed coding approach applicable to any bag-of-words technique. At the decoder, the joint stream statistics are expressed over feature histograms,

$$p(\Psi^1, \Psi^2, ..., \Psi^V) = p(\Psi^1) \prod_{i=2}^{V} p(\Psi^i | \Psi^{i-1}, ..., \Psi^1), \tag{8}$$

where the conditional probabilities are learned from training data as described in Section 3.

Assuming independence between the histogram bins and pair-wise dependence between histograms we write

$$p(\Psi^i | \Psi^{i-1}, ..., \Psi^1) = \prod_{k=1}^{i-1} \prod_{j=1}^{B} p(\psi_i^j | \Psi^k) \tag{9}$$

where ψ_i^j is the j-th bin of histogram Ψ^i, and $\Psi^i = [\psi_i^1, \cdots, \psi_i^B]^T$ with B denoting the number of bins. As demonstrated in our experiments, with the above simplified model one is able to effectively learn the dependency between feature histograms and achieve efficient distributed feature encoding.

The joint model of Equation 8 is used to determine which features at a given sensor are redundant with the side information. In particular, redundant features are those that are correlated with the redundant bins of the histogram of the current view. Since we are ultimately interested in the feature histograms for performing recognition, the encoders can send either histogram bin counts or the quantized visual features themselves.

We obtain a reconstruction of the feature histogram of each view from the view's decoded features and its side information. Before describing our GP distributed feature selection algorithm, we present our Gaussian Process model for bag-of-words representations in multi-view object recognition.

Let Ψ^v be the histogram of interest and Ψ^i, $i = 1, ..., v - 1$, its side information, where v is the current view considered by the decoder. From Equation 9 the probability of a histogram Ψ^v given its side information is found as,

$$p(\Psi^v | \Psi^{v-1}, ..., \Psi^1) = \prod_{k=1}^{v-1} p(\Psi^v | \Psi^k) \tag{10}$$

With our model, we learn a Gaussian Process for each bin of the feature histogram Ψ^v, assuming independence between bins,

$$p(\Psi^v | \Psi^{v-1}, ..., \Psi^1) = \prod_{j=1}^{B} \mathcal{N}(0, K^{b,v}) \tag{11}$$

where $B = |\Psi^v|$ is the number of histogram bins and a Gaussian Process is defined over each bin with kernel matrix $K^{b,v}$. We compute $K^{b,v}$ with a covariance function defined using an exponential kernel over the side

Algorithm 1 GP Distributed Feature Selection

```
Let $E_i$, $i = 1, ..., V$ be a set of encoders and $\xi$ be defined over sets of feature histogram bin indices.

Initialize $\xi^1 = \{1, ..., B\}$
for $v = 1, ..., V$ do
  $X_v^* =$ request$(E_i, \xi^v)$
  $\hat{X}_v =$ decode$(X_v^*)$
  for $b = 1, ..., B$ do
    $\sigma^b(\hat{\Psi}^v) = k^b(\hat{\Psi}^v, \hat{\Psi}^v) - (k_*^{b,v})^{\mathrm{T}} K^{-1} k_*^{b,v}$ (Eq. 7)
  end for
  if $v < V$ then
    $\xi^{v+1} =$ step$(\sigma)$
  end if
end for
```

information,

$$k_{i,j}^{b,v}(\Psi_i^v, \Psi_j^v) = \prod_{r=1}^{v} \gamma_b^{-v} \exp\left(\frac{d(\Psi_i^r, \Psi_j^r)^2}{\alpha_b^2}\right) + \eta_b \delta_{ij} \tag{12}$$

where γ_b, α_b are the kernel hyper-parameters of bin b, which we assume to be the same across views, η_b is a per-bin additive noise term. The kernel hyper-parameters are learned from training data as described in Section 3. We define a different set of kernel hyper-parameters per bin since each bin can exhibit drastically different behavior with respect to the side information.

The variance of each Gaussian Process can be used to determine whether a bin is redundant: a small bin variance indicates that the GP model is confident in its prediction, and therefore the features corresponding to that bin are likely to be redundant with respect to the side information. In our experiments, we found that redundant bins generally exhibit variances that are small and similar in value and that these variances are much smaller than those of non-redundant bins.

4.2 GP Distributed Feature Selection

Distributed feature selection is performed by the decoder using an iterative process. The decoder begins by querying the first encoder to send all of its features, since in the absence of any side information no feature is redundant. At the v-th sensor, the decoder requests only those features corresponding to the non-redundant histogram bins of the v-th image, whose indices are found using the bin variances output by each GP. In particular, we perform non-redundant bin detection at each view, by simply sorting the bin variances and finding the maximum local difference. At each iteration, the GPs are evaluated using the reconstructed histograms of previous iterations as illustrated in Algorithm 1.

Given the decoded features $\hat{X}^v$, the decoder reconstructs histograms $\hat{\Psi}^v = [\hat{\psi}_i^1, \cdots, \hat{\psi}_i^B]^T$, $v = 1, ..., V$, such that bins that are non-redundant are those received and the redundant bins are estimated from the GP mean prediction

$$\hat{\psi}_v^b = \begin{cases} \tilde{\psi}_v^b, & b \in \xi^v \\ (k_*^{b,v})^{\mathrm{T}} K^{-1} \mathbf{y}_v^b, & \text{otherwise.} \end{cases} \tag{13}$$

where $\mathbf{y}_v^b$ are the bin values for view v and bin b in the training data, ξ^v are the bin indices of the non-redundant bins of the histogram of view v, and $\tilde{\Psi}^v = \Psi(\hat{X}^v) = [\tilde{\psi}_i^1, \cdots, \tilde{\psi}_i^B]^T$.

The GP distributed feature selection algorithm achieves a compression rate proportional to the number of bin indices requested for each view. For view v the compression rate of our algorithm in percent bins transmitted is

$$R = \frac{b}{B} = \frac{2|\xi^v|}{B}, \tag{14}$$

where B is the total number of histogram bins and b is the number of bins received, which is proportional to twice the number of non-redundant bins as a result of the decoder `request` operation. Note, however, that in the case of large amounts of redundancy there are few non-redundant bins encoded at each view and therefore a small encoding rate is achieved.

Our Gaussian Process distributed feature selection algorithm is summarized in Algorithm 1.

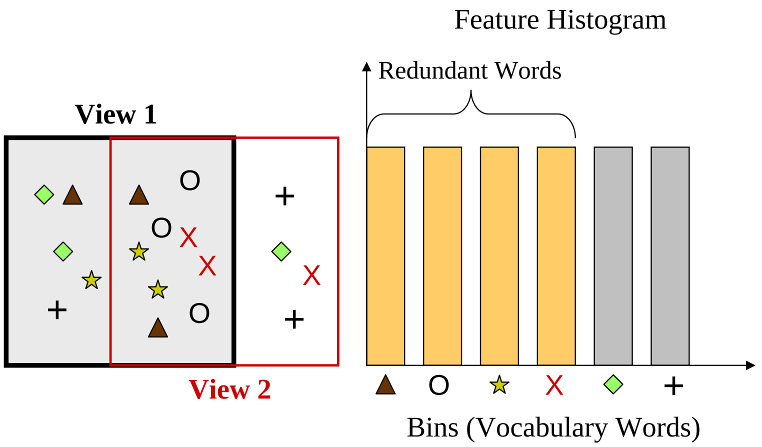


Figure 3: Synthetic example considered below. This scenario consists of two overlapping views of an object, which is presumed to fill the scene. Image features are represented using a 6 word vocabulary.

5 Experiments

We evaluate our distributed coding approach on the tasks of object recognition and indexing from multiple views. Given Ψ^v, $v = 1, .., n$, multi-view recognition is performed using a nearest-neighbor classifier over the fused distance measure, computed as the average distance across views

$$D_{i,j}(\Psi_i, \Psi_j) = \frac{1}{n} \sum_{v=1}^{n} d(\Psi_i^v, \Psi_j^v) \tag{15}$$

where for flat histograms we define $d(\cdot)$ using either the L_1 or L_2 norm, and with pyramid match similarity [5] for multi-resolution histograms[1].

We consider two performance metrics for nearest-neighbor classification. The first metric is a *majority* rule. Under this metric a query example is correctly classified if a majority ($\geq k/2$) of its k nearest-neighbors are of the same category or instance. The second metric is an *at-least-one* rule. With this metric, a query example is correctly classified if at least one of its k nearest-neighbors is of the same category or instance. The at-least-one metric has complementary behavior to the majority rule and defines a more lenient evaluation criteria; it can be useful for database retrieval tasks where results are provided for evaluation by an end-user. We compare distributed coding to independent encoding at each view with a random feature selector that randomly selects features or histogram bins according to a uniform distribution, and report feature selection performance in terms of percent bins encoded as $R = b/B$, where b is number of encoded histogram bin values and B the total number of histogram bins.

In what follows, we first present experiments on a synthetic example with our approach and then discuss our results on COIL-100.

5.1 Synthetic Example

To demonstrate our distributed feature selection approach we consider the scenario illustrated in Figure 3. In this scenario an object is imaged with two overlapping views, and the image features of each view are represented using a 6 word vocabulary. As shown by the figure, the images are redundant in 4 of the 6 words, as 2 of the words (i.e., diamond and plus) do not appear in the overlapping portion of each view. Although real-world problems are much more complex than described above, we use this simple scenario to give intuition and motivate our approach.

We first consider the case where there is no noise between the redundant features in each view and the redundant features appear only in the overlapping region. To simulate this scenario, we randomly generated $n = 100$, 6-D histograms, where each histogram was generated by sampling its bins from a uniform distribution between 0 and 1, and the histograms were normalized to sum to one. Each histogram was split into two views by replicating the first 4 bins in each view and randomly splitting the other two bins. The above data was used to form a training set of examples, where each pair of histograms corresponds to a single object

[1] Note our distributed coding algorithm is independent of the choice of classification method and others can be used with our method.

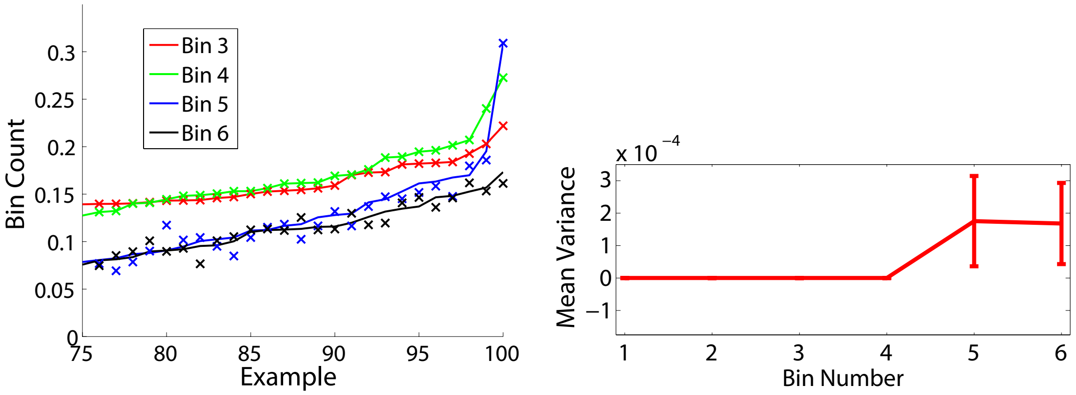


Figure 4: The prediction of each GP is plotted with crosses; the ground truth is shown as the solid line. Note that in this synthetic example the redundant dimensions are much more accurately predicted than the non-redundant ones (Bin 5, Bin 6). (Bins 1,2 have similar performance as 3,4, and are not shown for the sake of clarity).

instance. To form the test set zero mean Gaussian noise was added to the training set with $\sigma = 0.01$ and the test set histograms were split into two views using the same split ratios as the training set.

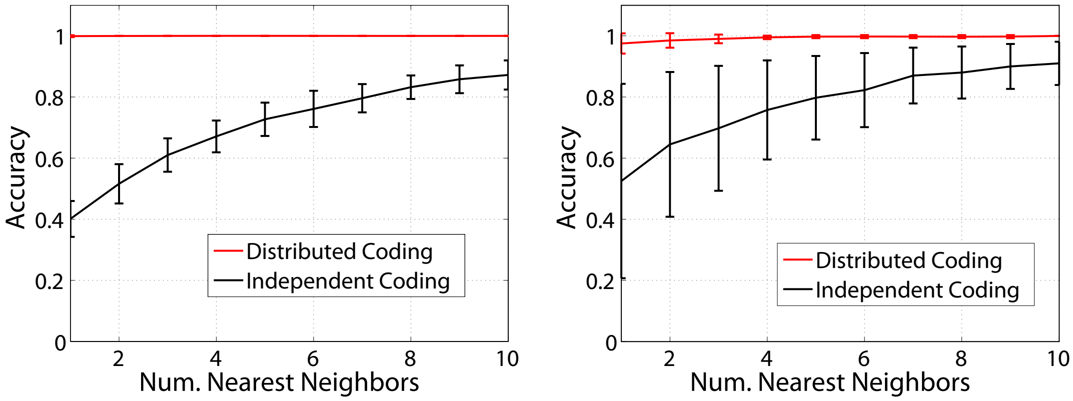
For distributed coding we trained 6 GPs, one per dimension, using each view. Figure 4 displays the predicted bin value of 4 of the 6 GPs evaluated on the second view of the test set. In the figure, examples are sorted per dimension by bin value and the output of each GP is plotted over ground truth. The GPs are able to learn the deterministic mapping that relates the redundant bins. For the 2 non-redundant bins, the variance of the GPs predictions is quite large compared to that of the redundant bins. Also shown in Figure 4, are the mean GP variances plotted for each histogram bin. The error bars in the plot indicate the standard deviation. As seen from the figure, the GP variance is much larger for the non-redundant bins than those of the redundant ones whose variances are small and centered about 0. This is expected since non-redundant bins are uncorrelated and therefore the GPs are less certain in their prediction of the value of these bins from side information.

Evaluating our distributed coding algorithm on the above problem gave a bin rate of $R = 0.66$ in the second view (see Equation 14), i.e., the second view was compressed by a factor of $1.5 : 1$ with our algorithm. Figure 5(a) displays the result of nearest-neighbor instance-level recognition over each of the 100 instances in the training set for varying neighborhood sizes. In this Figure, the average recognition performance, averaged over 10 independent trials, is shown for both distributed and independent coding of the second view, where for independent coding features were selected at the same rate as distributed coding. As seen from the figure, distributed coding far outperforms independent encoding in the above scenario.

We also considered the case of partial redundancy, where the redundant bins are only partially correlated as a result of noise. To simulate partial redundancy we added zero mean Gaussian noise to the split ratios of the first 4 bins with $\sigma = \{0, 0.01, 0.05, 0.1\}$ in the above experiment. Figure 5(b) displays the result of nearest-neighbor recognition with distributed and independent coding of the second view. In the plot, average recognition performance is reported, averaged across the different σ values, along with error bars indicating the standard deviation. For this experiment, an average bin rate of $R = 0.78 \pm 0.23$ was achieved with our distributed feature selection algorithm. As seen from the figure, our distributed coding algorithm can perform favorably to independent encoding even when the bins are only partially redundant.

5.2 COIL-100 Experiments

We evaluated our distributed feature selection algorithm using the COIL-100 multi-view object database [11] that consists of 72 views of 100 objects viewed from 0 to 360 degrees in 5 degree increments. A local feature representation is computed for each image using 10 dimensional PCA-SIFT features [6] extracted on a regular 4 x 4 grid. We evaluate our distributed coding algorithm and perform recognition with the COIL-100 dataset using multi-resolution vocabulary-guided histograms [4] computed with LIBPMK [7], a publicly available library for computing bag-of-words image representations. We split the COIL-100 dataset into a training and test set by taking alternating views of each object. We then paired images 50 degrees apart to form the two views of our problem.



(a) (b)

Figure 5: Nearest-neighbor instance-level recognition for the two-view synthetic dataset; average recognition accuracy is plotted over varying neighborhood sizes, error bars indicate ± 1 standard deviation. (a) Full redundancy. For a fixed rate, our algorithm far outperforms the independent encoding baseline. (b) Partial redundancy. Our distributed coding algorithm performs favorably to independent encoding even when the bins are only partially redundant. (see text for details)

Using the training image features we perform hierarchical k-means clustering to compute the global feature vocabulary used to form the multi-resolution pyramid representation. Using 4 levels and a tree branch factor of 10 gave a 991 word vocabulary at the finest level of the hierarchy. GP distributed feature selection is performed over the finest level of the histogram, such that the encoders and decoder only communicate bins at this level. The upper levels of the tree are then recomputed from the bottom level when performing recognition. To perform GP distributed coding we used a kernel defined using L2 distance over a coarse, flat histogram representation, although pyramid match similarity can also be used.

Figure 6 displays the result of nearest-neighbor recognition for one and two views using both the at-least-one and majority performance metrics. As seen from the figure, a significant performance increase is achieved by using the second view when there are no bandwidth constraints. Applying GP distributed feature selection on the above dataset resulted in a bin rate (percent bins encoded) of $R < 0.01$ in the second view; this is a compression rate of over 100:1. Figure 6 displays the result of recognition performance of the histograms found with our GP distributed feature selection algorithm. As seen from the figure, by exploiting data redundancy our algorithm performs significantly better than single view performance while achieving a very low encoding rate. The result of independent encoding is also shown in the Figure, where independent feature selection was performed at the same rate as our algorithm. As seen from the figure, in contrast to our approach, independent encoding is not able to improve over single-view performance and in fact does worse at such low encoding rates.

6 Conclusion and Future Work

In this paper we presented an unsupervised distributed feature selection algorithm for multi-view object recognition. We developed a new method for distributed coding with Gaussian Processes and demonstrated its effectiveness for encoding visual word feature histograms on both synthetic and real-world datasets. For a two-view problem with COIL-100, our algorithm was able to achieve a compression rate of over 100:1 in the second view, while significantly increasing accuracy over single-view performance. At the same coding rate, independent encoding was unable to improve over recognition with a single-view. For future work, we plan to investigate techniques for modeling more complex dependencies as well as one-to-many mappings between views.

References

[1] J. Dy and C. Brodley. Feature subset selection and order identification for unsupervised learning. In *ICML*, 2000. 3

[2] V. Ferrari, T. Tuytelaars, and L. V. Gool. Integrating multiple model views for object recognition. *CVPR*, 2004. 2

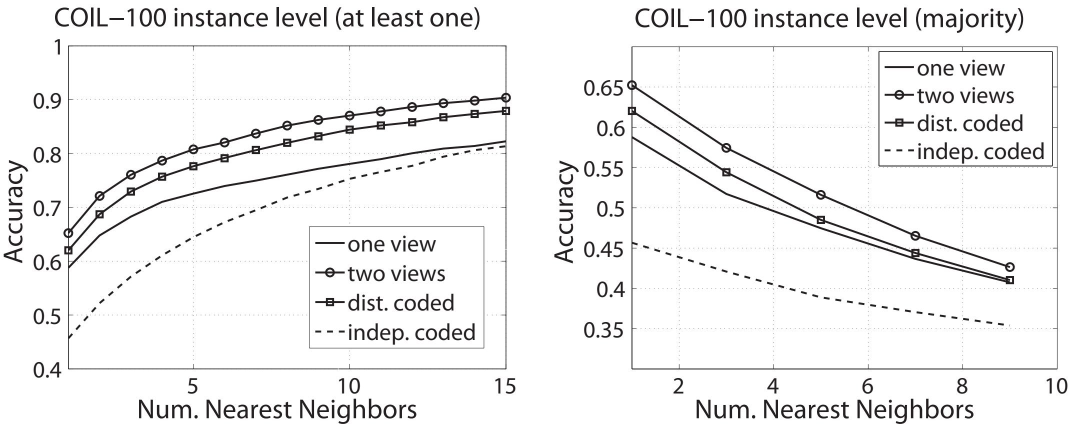


Figure 6: Nearest-neighbor recognition with two-views on COIL-100 with the (left) at-least-one and (right) majority performance metric. Our algorithm performs significantly better over single view performance under each metric while achieving a very low encoding rate. For the at-least-one metric, our approach performs near multi-view performance. The independent encoding baseline is also shown, where independent feature selection was performed at the same rate as our algorithm. Note that independent encoding with two views does worse than a single view when operating at such a low encoding rate.

[3] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, Jan. 2005. 1, 3

[4] K. Grauman and T. Darrell. Approximate correspondences in high dimensions. In *NIPS*, 2006. 8

[5] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 2007. 1, 2, 4, 5, 7

[6] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. *CVPR*, 2004. 8

[7] J. J. Lee. *LIBPMK: A Pyramid Match Toolkit*. MIT, http://people.csail.mit.edu/jjl/libpmk/. 8

[8] B. Leibe and B. Schiele. Scale-invariant object categorization using a scale-adaptive mean-shift search. In *DAGM*, 2004. 2

[9] H. Liu and L. Yu. Feature selection for data mining. Technical report, Arizona State University, 2002. 2, 3

[10] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 2005. 4

[11] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-100). Technical report, Colmubia University, 1996. 2, 8

[12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. 2006. 1, 2, 4, 5

[13] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *PAMI*, 2005. 3

[14] S. S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (discus): design and construction. *Transactions on Information Theory*, 2003. 1, 3

[15] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006. 3, 4

[16] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *IJCV*, 2006. 2

[17] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007. 2

[18] D. Schonberg. *Practical Distributed Source Coding and Its Application to the Compression of Encrypted Data*. PhD thesis, University of California, Berkeley, 2007. 1, 3

[19] J. Sivic and A. Zisserman. *Toward Category-Level Object Recognition*, chapter Video Google: Efficient Visual Search of Videos. Springer, 2006. 1, 5

[20] D. Slepian and J. Wolf. Noiseless coding of correlated information sources. *Transactions on Information Theory*, 1973. 1, 3

[21] A. Thomas, V. Ferrar, B. Leibe, T. Tuytelaars, B. Schiel, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006. 2

[22] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007. 2