ABSTRACT

Sparse representations using predefined and learned dictionaries have widespread applications in signal and image processing. Sparse approximation techniques can be used to recover data from its low dimensional corrupted observations, based on the knowledge that the data is sparsely representable using a known dictionary. In this paper, we propose a method to improve data recovery by ensuring that the data recovered using sparse approximation is close to its manifold. This is achieved by performing regularization using examples from the data manifold. This technique is particularly useful when the observations are highly reduced in dimensions when compared to the data and corrupted with high noise. Using an example application of image inpainting, we demonstrate that the proposed algorithm achieves a reduction in reconstruction error in comparison to using only sparse coding with predefined and learned dictionaries, when the percentage of missing pixels is high.

Index Terms— sparse representation, manifold projection, image inpainting, dictionary learning.

1. INTRODUCTION

Low dimensional manifold models find applications in representation and processing of natural signals and images. Typically, we do not have a complete description of the manifold, rather we only have samples obtained from it. The samples from the manifold are also referred to as examples. When modeling the low dimensional manifolds, a useful approach is to consider the manifold as a union of convex sets, where each convex set is formed using a small subset of examples [1, 2]. This amounts to a piecewise linear approximation of the manifold using low dimensional simplices, similar to piecewise linear approximation of one dimensional curves using lines. Furthermore, this interpretation can generalize well to cases where we have data clustered in disjoint unions of convex sets rather than a continuous submanifold. A useful application of this generalization is in the case of high contrast natural image patches, which have been shown to lie in clusters and non-linear low dimensional submanifolds [3].

Recent approaches in representing natural image patches involve learning dictionaries that can represent the patches using a sparse linear combination of the dictionary atoms. Dictionary learning algorithms such as the K-SVD [4] and multilevel dictionary learning [5] have been successfully used for sparse coding in applications such as denoising [6], classification [7] and compressive recovery [5]. A dictionary learned from a sufficient number of training examples from the data manifold captures the key features present in the data. In scenarios with missing or incomplete data, the dimensionality of the observations is lesser than that of the original data. Methods that use sparse approximations for the recovery of the original data reduce the dimension of the dictionary and hence operate in a lower dimensional space. Examples of such scenarios include image inpainting [4] and compressive sensing [5]. The efficiency of such recovery schemes can be improved if additional regularization is performed using the higher dimensional training examples from the data manifold. This is particularly useful when the observations have highly reduced dimensions and are corrupted by a high degree of noise.

In this paper, we propose an algorithm that uses samples from the data manifold directly when computing sparse representations with a predefined dictionary, without the need to learn dictionaries. We compute a sparse code using the observation such that the reconstructed data lies close to the data manifold. Since natural image patches lie close to low dimensional submanifolds, utilizing the additional information about the data manifold leads to an improved performance in applications involving recovery of image patches. Reconstruction of the test data is performed using only the sparse code and the predefined dictionary, and does not require knowledge of the manifold. Inpainting of images with high percentage of missing pixels and high noise corruption is chosen as the application of this approach. Results demonstrate that the proposed approach using manifold projection and a DCT (Discrete Cosine Transform) dictionary performs better the standard sparse coding based approach with predefined and learned dictionaries.

2. BACKGROUND

2.1. Sparse Coding

The test data sample to be represented is denoted by the vector \( y \in \mathbb{R}^M \). The \( T \) training samples obtained from the un-
derlying manifold $\mathcal{M}$, of the test sample, are indicated by $X = \{x_i\}_{i=1}^T$, where $x_i \in \mathbb{R}^M$ and $X \in \mathbb{R}^{M \times T}$. The dictionary used for sparse representation of the test data is denoted by $D = \{d_j\}_{j=1}^K$, where $D \in \mathbb{R}^{M \times K}$. The dictionary can be a collection of predefined basis functions, such as the DCT, or can be learned from the training set itself using algorithms such as the K-SVD [4] or multilevel dictionary learning [5]. In order to compute an $S$-sparse representation of the test vector $y$ using a dictionary $D$, the following optimization problem has to be solved,

$$\hat{\gamma} = \arg\min_{\gamma} \|y - D\gamma\|_2^2 \text{ subject to } \|\gamma\|_0 \leq S$$

(1)

where $\|\cdot\|_2$ indicates the $\ell_2$ norm and $\|\cdot\|_0$ indicates the $\ell_0$ norm. Because of the combinatorial complexity of $\ell_0$ minimization, the problem of computing the sparse representation is usually solved as an $\ell_1$ minimization problem,

$$\hat{\gamma} = \arg\min_{\gamma} \|y - D\gamma\|_2^2 + \lambda \|\gamma\|_1$$

(2)

where $\lambda$ is the parameter that controls the trade-off between sparsity and reduction in error. The coefficients obtained by solving (2) have been shown to be equivalent to the sparse code obtained with $\ell_0$ minimization under certain conditions on the dictionary $D$ [8]. Global dictionaries learned using patches from a set of natural images have been found to generalize well in obtaining sparse representations of a wide range of images, not included in the training set [5]. The process of dictionary learning can effectively identify representative patterns from samples lying on a manifold. However, the proposed method explicitly forces the reconstructed data to lie close to the manifold and improves the approximation, particularly in cases when we only have observations that are highly corrupted, low dimensional versions of the test data.

2.2. Manifold Projection using Weighted Sparse Coding

Let the set of vectors in $X$ be the samples obtained from the manifold $\mathcal{M}$. Projecting a test sample $y$ onto the manifold $\mathcal{M}$ can be performed by selecting a small set of samples $X_\Omega = \{x_i\}_{i \in \Omega}$ in the neighborhood of $y$ and projecting $y$ onto the simplex spanned by $X_\Omega$. $\Omega$ is the index set of manifold samples in the neighborhood of $y$. Usually the neighborhood is chosen using an $\ell_2$ distance measure, and this approach has been used in manifold learning algorithms such as the Locally Linear Embedding (LLE) [9]. The selection of the neighborhood and projection of the test sample onto the low dimensional simplex can be posed as a weighted sparse coding problem,

$$\hat{\beta} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda \sum_{i=1}^T \|y - x_i\|_2^2/\beta_i$$

subject to $\sum_{i=1}^T \beta_i = 1, \beta_i \geq 0 \forall i$. 

(3)

where $\beta \in \mathbb{R}^T$ are coefficients for the projection of $y$ on $X$. The constraints on $\beta$ ensure that the test sample is projected onto the convex hull (simplex) spanned by the chosen manifold samples. The optimization problem given in (3) has been used in [7] to learn manifolds from high dimensional data and shown to be highly beneficial for image classification.

3. COMBINED SPARSE CODING AND MANIFOLD PROJECTION

In this section, we present the proposed approach to combine the paradigms of manifold projection and sparse coding in order to achieve a better representation performance. We begin by assuming that the test sample, $y$, lies near the manifold $\mathcal{M}$, which is true in many cases such as when the data are patches from natural images.

The knowledge of the underlying manifold can be exploited by learning a dictionary directly from the manifold samples and computing a sparse code for the test data using the learned dictionary. However, in cases with missing/incomplete data, we have the observations $\hat{y} = \Phi y + n$, where $\Phi \in \mathbb{R}^{N \times M}$ with $N < M$ and $n \sim \mathcal{N}(0, \sigma^2 I_N)$. When $N << M$ and $\sigma^2$ is high, apart from sparsity constraints, additional regularization in the form of samples from the manifold will aid in the recovery of $y$. We propose a framework in which we use a pre-defined dictionary, but use training examples (manifold samples) for regularization. It is important to note that once the sparse code is generated, the proposed approach performs the reconstruction using only the predefined dictionary and we do not require the manifold samples. To summarize, the goal of the proposed approach is to find a sparse code using the observation, such that the recovered test sample lies close to the manifold $\mathcal{M}$.

The proposed algorithm involves two components: (a) computing the sparse code using the observation, $\hat{y}$, and $D$ and (b) projection of the recovered test sample $D\hat{\gamma}$ close to $\mathcal{M}$. This in essence involves combining the sparse representation and manifold projection problems given in (2) and (3) into the joint optimization problem,

$$\{\hat{\gamma}, \hat{\beta}\} = \arg\min_{\gamma, \beta} \|D\gamma - X\beta\|_2^2 + \lambda \|\gamma\|_1$$

subject to $\|\hat{y} - \Phi D\gamma\|_2^2 \leq \epsilon, \sum_{i=1}^T \beta_i = 1, \beta_i \geq 0 \forall i$. 

(4)

Here, $\gamma$ denotes the sparse code for the observation $\hat{y}$, $\beta$ corresponds to the coefficient vector obtained from manifold projection, and $\lambda$ controls the trade-off between sparsity of $\gamma$ and the manifold projection residual. The first term in the objective function along with the second and third constraints perform the manifold projection on the recovered data $D\gamma$. When compared to (3), the manifold projection component in the cost function of (4) does not contain the weighted sparsity term, $\sum_{i=1}^T \|y - x_i\|_2^2/\beta_i$. This omission does not affect the
In order to demonstrate the effectiveness of the proposed approach, we performed inpainting on a set of standard images (lena, boat, peppers, house, couple and man) with 75% missing pixels and corrupted with additive Gaussian noise of \( \sigma = \{15, 20, 25\} \). The dictionary \( \mathbf{D} \) used in the proposed approach was overcomplete DCT of size \( 64 \times 256 \). The manifold samples consisted of 12, 500 randomly chosen patches of size \( 8 \times 8 \) from 250 training images of the Berkeley Segmentation Dataset (BSDS) [11]. The K-SVD dictionary was learned directly from the 12, 500 training samples. The training samples were preprocessed in three stages and a fraction of the samples were stacked in the matrix \( \mathbf{X} \) for use with the proposed framework. In the first stage of preprocessing, for each sample (64-dimensional), the mean value of its coordinates was subtracted from each of its coordinate. The second step involved computing the energy, as the sum of squares of coordinates, for each sample. The mean energy of all samples was computed and only those samples with energy greater than 95% of the mean energy were retained. The third stage was density filtration [3] in which we specified the number of nearest neighbors (15 here) and picked a percentage of the samples (30% here) that had their neighbors closest to them in space. This procedure identified the samples from regions in space where there was a high concentration of samples. \( \mathbf{X} \) contained 2678 samples after the preprocessing steps. The parameters for preprocessing were chosen in order to ensure that sufficient number of representative training samples were available, but the number was not so high that it would slow down the optimization. The regularization parameter \( \lambda \) in (4) was chosen as 1 using experiments as this gave a good trade-off between the sparsity term and the manifold projection error term.

The standard images corrupted with noise, were divided into non-overlapping patches of size \( 8 \times 8 \). 75% percent of the pixels were randomly chosen from the image and their indices were marked for masking. After dividing the image into patches, for each patch only the unmasked pixels were stored in the observation, \( \hat{\mathbf{y}} \). The number of unmasked pixels was \( N \) and \( \Phi \) would be an identity matrix with \( N \) of its randomly chosen rows retained. Inpainting based on sparse coding consisted of solving (2) using \( \hat{\mathbf{y}} \) and \( \Phi \mathbf{D} \), as the observation and the dictionary respectively. Inpainting using the proposed approach was performed by solving (4) for each observation \( \hat{\mathbf{y}} \) using \( \mathbf{X} \) and \( \Phi \mathbf{D} \) as manifold examples and the dictionary respectively. The residual error goal was computed as \( (1.1 \sigma)^2 N \) in both cases. In both sparse coding as well the proposed approach, the image patches were recovered as \( \Phi \mathbf{D} \hat{\mathbf{g}} \) using their respective sparse codes. The root mean-squared error (RMSE) of the inpainting procedure for all the \( P \) test patches is,

\[
RMSE = \left[ \frac{1}{MP} \sum_{i=1}^{P} \| y_i - \Phi \hat{\gamma}_i \|_2^2 \right]^{1/2}.
\]  

The PSNR is computed as 
\[
20 \log_{10}(255/RMSE).
\]  

4. SIMULATION RESULTS

In order to demonstrate the effectiveness of the proposed framework, we performed inpainting on a set of standard images (lena, boat, peppers, house, couple and man) with 75% missing pixels and corrupted with additive Gaussian noise of \( \sigma = \{15, 20, 25\} \). The dictionary \( \mathbf{D} \) used in the proposed approach was overcomplete DCT of size \( 64 \times 256 \). The manifold samples consisted of 12, 500 randomly chosen patches of size \( 8 \times 8 \) from 250 training images of the Berkeley Segmentation Dataset (BSDS) [11]. The K-SVD dictionary was learned directly from the 12, 500 training samples. The training samples were preprocessed in three stages and a fraction of the samples were stacked in the matrix \( \mathbf{X} \) for use with the proposed framework. In the first stage of preprocessing, for each sample (64-dimensional), the mean value of its coordinates was subtracted from each of its coordinate. The second step involved computing the energy, as the sum of squares of coordinates, for each sample. The mean energy of all samples was computed and only those samples with energy greater than 95% of the mean energy were retained. The third stage was density filtration [3] in which we specified the number of nearest neighbors (15 here) and picked a percentage of the samples (30% here) that had their neighbors closest to them in space. This procedure identified the samples from regions in space where there was a high concentration of samples. \( \mathbf{X} \) contained 2678 samples after the preprocessing steps. The parameters for preprocessing were chosen in order to ensure that sufficient number of representative training samples were available, but the number was not so high that it would slow down the optimization. The regularization parameter \( \lambda \) in (4) was chosen as 1 using experiments as this gave a good trade-off between the sparsity term and the manifold projection error term.

The standard images corrupted with noise, were divided into non-overlapping patches of size \( 8 \times 8 \). 75% percent of the pixels were randomly chosen from the image and their indices were marked for masking. After dividing the image into patches, for each patch only the unmasked pixels were stored in the observation, \( \hat{\mathbf{y}} \). The number of unmasked pixels was \( N \) and \( \Phi \) would be an identity matrix with \( N \) of its randomly chosen rows retained. Inpainting based on sparse coding consisted of solving (2) using \( \hat{\mathbf{y}} \) and \( \Phi \mathbf{D} \), as the observation and the dictionary respectively. Inpainting using the proposed approach was performed by solving (4) for each observation \( \hat{\mathbf{y}} \) using \( \mathbf{X} \) and \( \Phi \mathbf{D} \) as manifold examples and the dictionary respectively. The residual error goal was computed as \( (1.1 \sigma)^2 N \) in both cases. In both sparse coding as well the proposed approach, the image patches were recovered as \( \Phi \mathbf{D} \hat{\mathbf{g}} \) using their respective sparse codes. The root mean-squared error (RMSE) of the inpainting procedure for all the \( P \) test patches is,

\[
RMSE = \left[ \frac{1}{MP} \sum_{i=1}^{P} \| y_i - \Phi \hat{\gamma}_i \|_2^2 \right]^{1/2}.
\]  

The PSNR is computed as 
\[
20 \log_{10}(255/RMSE).
\]  

4. SIMULATION RESULTS

In order to demonstrate the effectiveness of the proposed framework, we performed inpainting on a set of standard images (lena, boat, peppers, house, couple and man) with 75% missing pixels and corrupted with additive Gaussian noise of \( \sigma = \{15, 20, 25\} \). The dictionary \( \mathbf{D} \) used in the proposed approach was overcomplete DCT of size \( 64 \times 256 \). The manifold samples consisted of 12, 500 randomly chosen patches of size \( 8 \times 8 \) from 250 training images of the Berkeley Segmentation Dataset (BSDS) [11]. The K-SVD dictionary was learned directly from the 12, 500 training samples. The training samples were preprocessed in three stages and a fraction of the samples were stacked in the matrix \( \mathbf{X} \) for use with the proposed framework. In the first stage of preprocessing, for each sample (64-dimensional), the mean value of its coordinates was subtracted from each of its coordinate. The second step involved computing the energy, as the sum of squares of coordinates, for each sample. The mean energy of all samples was computed and only those samples with energy greater than 95% of the mean energy were retained. The third stage was density filtration [3] in which we specified the number of nearest neighbors (15 here) and picked a percentage of the samples (30% here) that had their neighbors closest to them in space. This procedure identified the samples from regions in space where there was a high concentration of samples. \( \mathbf{X} \) contained 2678 samples after the preprocessing steps. The parameters for preprocessing were chosen in order to ensure that sufficient number of representative training samples were available, but the number was not so high that it would slow down the optimization. The regularization parameter \( \lambda \) in (4) was chosen as 1 using experiments as this gave a good trade-off between the sparsity term and the manifold projection error term.

The standard images corrupted with noise, were divided into non-overlapping patches of size \( 8 \times 8 \). 75% percent of the pixels were randomly chosen from the image and their indices were marked for masking. After dividing the image into patches, for each patch only the unmasked pixels were stored in the observation, \( \hat{\mathbf{y}} \). The number of unmasked pixels was \( N \) and \( \Phi \) would be an identity matrix with \( N \) of its randomly chosen rows retained. Inpainting based on sparse coding consisted of solving (2) using \( \hat{\mathbf{y}} \) and \( \Phi \mathbf{D} \), as the observation and the dictionary respectively. Inpainting using the proposed approach was performed by solving (4) for each observation \( \hat{\mathbf{y}} \) using \( \mathbf{X} \) and \( \Phi \mathbf{D} \) as manifold examples and the dictionary respectively. The residual error goal was computed as \( (1.1 \sigma)^2 N \) in both cases. In both sparse coding as well the proposed approach, the image patches were recovered as \( \Phi \mathbf{D} \hat{\mathbf{g}} \) using their respective sparse codes. The root mean-squared error (RMSE) of the inpainting procedure for all the \( P \) test patches is,

\[
RMSE = \left[ \frac{1}{MP} \sum_{i=1}^{P} \| y_i - \Phi \hat{\gamma}_i \|_2^2 \right]^{1/2}.
\]  

The PSNR is computed as 
\[
20 \log_{10}(255/RMSE).
\]
ages is summarized in Table 1. It can be seen that for 75% missing pixels at all high noise levels, the proposed approach performs substantially better than sparse coding using DCT and K-SVD dictionaries. We also repeated these tests with a set of 12 images chosen from the BSDS testing set [11]. The images contained a wide variety of scenes with anisotropic, isotropic textures and geometric patterns. On an average, with 75% of the pixels missing and \( \sigma = 20 \), the proposed algorithm resulted in a PSNR of 18.76 dB, whereas sparse coding with DCT and K-SVD dictionaries resulted in PSNRs of 18.09 dB and 18.18 dB respectively.

### Table 1. Comparison of PSNRs obtained with inpainting for standard images, with 75% missing pixels, using sparse coding (with DCT and K-SVD dictionaries) and the proposed algorithm (with DCT dictionary).

<table>
<thead>
<tr>
<th>Noise std. dev.</th>
<th>Method Used</th>
<th>Image</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Boat</td>
<td>Man</td>
</tr>
<tr>
<td>15</td>
<td>DCT</td>
<td>20.21</td>
<td>20.56</td>
</tr>
<tr>
<td>20</td>
<td>DCT</td>
<td>19.27</td>
<td>19.72</td>
</tr>
<tr>
<td></td>
<td>K-SVD</td>
<td>19.29</td>
<td>19.69</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>19.89</td>
<td>20.37</td>
</tr>
<tr>
<td>25</td>
<td>DCT</td>
<td>18.40</td>
<td>18.76</td>
</tr>
<tr>
<td></td>
<td>K-SVD</td>
<td>18.43</td>
<td>18.68</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>18.91</td>
<td>19.31</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS

In this paper, we presented an optimization procedure that incorporates manifold projection in the sparse coding framework, thereby improving the recovery performance for missing/incomplete data cases. We demonstrated that, by performing regularization using manifold samples, we were able to perform better recovery of the inpainted image patches. This setup is particularly useful in conditions of severe corruption of images. Future extensions include investigating the use of this manifold based recovery framework in other forms of extreme dimensionality reduction, such as compressive sensing and performing analysis that quantifies the performance variations of the proposed framework in comparison to a sparse coding based recovery framework.

6. ACKNOWLEDGEMENT

This work is supported in part by the SenSIP center and consortium award number S0322.

7. REFERENCES


