U.P.B. Sci. Bull., Series C, Vol. 75, Iss. 2, 2013 ISSN 2286-3540

CLUSTERING LARGE DATASETS - BOUNDS AND APPLICATIONS WITH K-SVD

by Cristian Rusu

This article presents a clustering method called T-mindot that is used to reduce the dimension of datasets in order to diminish the running time of the training algorithms. The T-mindot method is applied before the K-SVD algorithm in the context of sparse representations for the design of overcomplete dictionaries. Simulations that run on image data show the efficiency of the proposed method that leads to the substantial reduction of the execution time of K-SVD, while keeping the representation performance of the dictionaries designed using the original dataset.

Keywords: sparse representations, clustering, KSVD.
MSC2000: 94A 12.

1. Introduction

The problem of designing overcomplete dictionaries for sparse representations [1] has received a lot of attention during the recent years. Fueled by the goal to create dictionaries for very efficient representations, this framework was applied with success in many application areas [2] [3].

The problem is formulated in the following manner: given a dataset $\boldsymbol{Y} \in \mathbb{R}^{n \times N}$ and a target sparsity level denoted k_0 solve the non-convex optimization problem

$$\begin{aligned} \underset{\boldsymbol{D},\boldsymbol{X}}{\text{minimize}} \quad & ||\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}||_F^2 \\ \text{subject to} \quad & ||\boldsymbol{x}_i||_0 \leq k_0, \ 1 \leq i \leq N \end{aligned} \tag{1}$$

where the matrix $\boldsymbol{D} \in \mathbb{R}^{n \times m}$ is called dictionary and its columns are called atoms, the sparse matrix $\boldsymbol{X} \in \mathbb{R}^{n \times N}$ is called the representation matrix with target sparsity k_0 on each column and $||\boldsymbol{E}||_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{N} E_{ij}^2$ is the Frobenius norm of the error matrix $\boldsymbol{E} = \boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}$.

A popular approach to find a good solution for problem (1) is the K-SVD [4] algorithm heuristic. This iterative algorithm employs a two step optimization procedure in the following manner:

(1) Keep the dictionary $\boldsymbol{D}$ fixed and compute the sparse representations using the orthogonal matching pursuit (OMP) algorithm [5].

Department of Automatic Control and Computers, University “Politehnica”of Bucharest, Spl. Independenţei 313, Bucharest 060042, Romania (e-mail: cristian.rusu@schur.pub.ro).

(2) Keep the representation matrix $\boldsymbol{X}$ fixed and update each column of the dictionary using the singular value decomposition (SVD) applied on the data samples that use the respective atom.

The algorithm stops after a fixed number of iterations or after a certain error limit is reached.

One of the most important issues with this approach is the fact that it runs quite slowly when the training set is large. This drawback makes it difficult to use in settings where a large number of samples are used in the training phase. Attempts have been made to speed up the procedure by implementing a bulk OMP solver and replacing the SVD with steps of the power method [6].

This article describes a reduction procedure called T-mindot [7] that takes place before the K-SVD algorithm is applied. The role of this reduction procedure is to diminish the dimension of the training set such that: the K-SVD steps run much faster and the result obtained on the new reduced set is relevant also on the original dataset. Additionaly, new theoretical results are presented that introduce bounds on the effect that the grouping procedure has on the input dataset and the way the K-SVD algorithm works on this new, clustered, dataset. The bounds describe the limits of performance and the trade-off between the achieved speed-up and the accuracy of the results.

New simulations on image data are presented and theoretical bounds are supplied to quantify the difference between the original dataset and the clustered one. All results validate the proposed method and show its efficiency.

Outline. The remainder of this article is structured as follows. Section 2 describes the T-mindot grouping procedure and Section 3 presents theoretical bounds that characterize the changes to the dataset made by the grouping procedure. In Section 4 the results obtained on image data show the impact that T-mindot has on the application of K-SVD. Conclusions end the article in Section 5.

Notation. The set of real numbers is described by $\mathbb{R}$. Bold characters denote multivariate entities (vectors, matrices).

2. The clustering procedure: T-mindot

This section describes the details of the proposed grouping procedure called T-mindot [7].

The main idea of the T-mindot is to split the dataset into groups and then replace each group with a single data item (called the centroid). In some sense, the grouping method reassembles the k-means clustering algorithm but the main difference is that the number of centroids is not apriori set. T-mindot groups the dataset with the target of discovering a variable number of centroids such that every data item from the dataset is within a maximum distance from at least one of the centroids.

Given the dataset $\boldsymbol{A} \in \mathbb{R}^{n\times N}$ with columns $\boldsymbol{a_i}, i = 1, \dots, N$ and the threshold parameter T, the grouping procedure is described by the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{c_j}}{\text{minimize}} \quad & M \\ \text{subject to} \quad & |\boldsymbol{a_i}^T \boldsymbol{c_j}| \geq T, \forall i, \text{ for at least one } j \\ & ||\boldsymbol{c_j}||_2 = 1. \end{aligned} \tag{2}$$

and the output is the matrix $\boldsymbol{C} \in \mathbb{R}^{n\times M}$ containing the centroids $\boldsymbol{c_j}, j = 1, \dots, M$ columnwise concatenated. Additionally, the extra output matrix $\boldsymbol{B} \in \mathbb{R}^{n\times N}$ represents the dataset where each data item is replaced with its associated centroid. This non-convex problem is solved by applying the T-mindot heuristic.

The general structure of T-mindot is:

(1) Parameters
   (a) $S = \max\{\lfloor 0.01 \times N \rfloor, 500\}$ - current working dimensions.
   (b) $S_{\text{max}} = \max\{\lfloor 0.02 \times N \rfloor, 2000\}$ - maximum working dimensions.
   (c) $\text{D}_{\text{fast}} = 1.1$ - control of fast dynamic.
   (d) $\text{D}_{\text{slow}} = 0.9$ - control of slow dynamic.

(2) Initialization: set the centroid set to be the first data item from the dataset $\boldsymbol{A}$.

(3) Iterative procedure
   (a) Extract in the set $\mathbb{W}$ a block of dimension S (maximum S_{max}) from the dataset $\boldsymbol{A}$.
   (b) Compute the dot products of the centroids with the data items from the set $\mathbb{W}$ and group the data items that are at the minimum distance (absolute value of the dot product greater than T)
   (c) Only with the data items that were not grouped in the previous step, begin a procedure that finds the best centroids among this remaining set of items and allocates the data items around these centroids.
   (d) If more than 5% of vectors are not clustered in the previous step then $S = \lceil S \times \text{D}_{\text{slow}} \rceil$, else $S = \lceil S \times \text{D}_{\text{fast}} \rceil$
   (e) Repeat until all data items are clustered.

(4) Finalization: Replace each centroid with the normalized average of its clustered data items.

Since the procedure is applied on highly correlated data we expect that $M \ll N$. This reduction is will lead to the speed up of the subsequent training procedure that is applied. The values of the parameters were chosen after conducting several numerical experiments.

The presented grouping procedure runs on the input training dataset before the application of the K-SVD algorithm.

3. Bounds

This section establishes bounds on the error terms that quantify the difference between the problem before and after T-mindot.

Like in the previous section, consider the normalized training vectors columnwise concatenated in the matrix $\boldsymbol{A} \in \mathbb{R}^{n\times N}$, the result of the clustering procedure in the matrix $\boldsymbol{B} \in \mathbb{R}^{n\times N}$ and in the matrix $\boldsymbol{C} \in \mathbb{R}^{n\times M}$, the concatenation of the weighted centroids found

$$\begin{array}{rl} \boldsymbol{A} = & [\boldsymbol{a_1} \quad \boldsymbol{a_2} \quad \dots \quad \boldsymbol{a_k} \quad \dots \quad \dots \quad \boldsymbol{a_N} \quad] \\ \boldsymbol{B} = & [\boldsymbol{a_1} \quad \boldsymbol{a_2} \quad \dots \quad \boldsymbol{c_1} \quad \dots \quad \dots \quad \boldsymbol{c_M} \quad] \\ = & [\boldsymbol{a_1} \quad \boldsymbol{a_2} \quad \dots \quad \boldsymbol{a_k}+\boldsymbol{v_k} \quad \dots \quad \dots \quad \boldsymbol{a_N}+\boldsymbol{v_N} \quad] \\ \boldsymbol{C} = & [\boldsymbol{a_1} \quad \boldsymbol{a_2} \quad \dots \quad \sqrt{s_1}\boldsymbol{c_1} \quad \dots \quad \sqrt{s_M}\boldsymbol{c_M}], \end{array} \tag{3}$$

where $||\boldsymbol{a_i}||_2 = 1$ and $||\boldsymbol{c_j}||_2 = 1$ chosen such that $|\boldsymbol{c_j}^T\boldsymbol{a_i}| \geq T$ for every i with at least one j, s_j represents the number of items clustered around centroid $\boldsymbol{c_j}, \forall j = 1, \dots, M$, $c = \sum_{j=1}^M s_j$ and $\boldsymbol{v_i}$ with $i = 1, \dots, N$ represents the offset from each training vector to the centroid such that

$$\boldsymbol{A}\boldsymbol{A}^T \approx \boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{C}\boldsymbol{C}^T. \tag{4}$$

We are interested in these products since the goal is to compute the singular values and vectors of the training matrix.

3.1. Bound for the error matrix. We discuss, without loss of generality, the case in which N training vectors group around the same centroid $\boldsymbol{c_1}$. It is obvious that we are interested to quantify the difference that is added in the clustering procedure, denoted here by $\boldsymbol{A}\boldsymbol{A}^T \approx \boldsymbol{B}\boldsymbol{B}^T$. Consider the error matrix $\boldsymbol{\Delta} \in \mathbb{R}^{n\times n}$

$$\begin{array}{rl} \boldsymbol{\Delta} = & \boldsymbol{B}\boldsymbol{B}^T - \boldsymbol{A}\boldsymbol{A}^T \\ = & \sum_{i=1}^N (\boldsymbol{a_i}+\boldsymbol{v_i})(\boldsymbol{a_i}+\boldsymbol{v_i})^T - \sum_{i=1}^N \boldsymbol{a_i}\boldsymbol{a_i}^T \\ = & \sum_{i=1}^N (\boldsymbol{a_i}\boldsymbol{v_i}^T + \boldsymbol{v_i}\boldsymbol{a_i}^T + \boldsymbol{v_i}\boldsymbol{v_i}^T) \\ = & \sum_{i=1}^N (\boldsymbol{c_1}\boldsymbol{v_i}^T + \boldsymbol{v_i}\boldsymbol{a_i}^T) = \sum_{i=1}^N \boldsymbol{\Delta_i}. \end{array} \tag{5}$$

Our first attempt is to bound the Frobenius norm of the error matrix, $||\boldsymbol{\Delta}||_F^2 = \sum_{i=1}^n \sum_{j=1}^n \delta_{ij}^2$. Following (5) we get

$$\begin{array}{rl} ||\boldsymbol{\Delta}||_F^2 = & ||\sum_{i=1}^N \boldsymbol{\Delta_i}||_F^2 \leq \sum_{i=1}^N ||\boldsymbol{\Delta_i}||_F^2 \\ = & \sum_{i=1}^N \operatorname{tr}((\boldsymbol{a_i}\boldsymbol{v_i}^T + \boldsymbol{v_i}\boldsymbol{c_1}^T)(\boldsymbol{v_i}\boldsymbol{a_i}^T + \boldsymbol{c_1}\boldsymbol{v_i}^T)) \\ = & \sum_{i=1}^N 2\boldsymbol{v_i}^T\boldsymbol{v_i} + 2\boldsymbol{v_i}^T\boldsymbol{c_1}\boldsymbol{v_i}^T\boldsymbol{a_i} \\ = & \sum_{i=1}^N 2(1-(\boldsymbol{c_1}^T\boldsymbol{a_i})^2) \\ \leq & 2N(1-T^2), \end{array} \tag{6}$$

where we used the fact that $\boldsymbol{v_i}^T\boldsymbol{v_i} = 2(1-\boldsymbol{c_1}^T\boldsymbol{a_i})$ and $\boldsymbol{v_i}^T\boldsymbol{c_1}\boldsymbol{v_i}^T\boldsymbol{a_i} = -(1-\boldsymbol{c_1}^T\boldsymbol{a_i})^2$. Notice that the last inequality in (6) is the worst case bound.

Secondly, notice that because

$$\begin{array}{rl} \mathrm{tr}(\boldsymbol{\Delta_i}) = & \mathrm{tr}(\boldsymbol{a_i v_i^T} + \boldsymbol{v_i a_i^T} + \boldsymbol{v_i v_i^T}) \\ = & \boldsymbol{v_i^T a_i} + \boldsymbol{v_i^T a_i} + \boldsymbol{v_i^T v_i} \\ = & (\boldsymbol{a_i} - \boldsymbol{c_1})^T(\boldsymbol{a_i} + \boldsymbol{c_1}) \\ = & \boldsymbol{a_i^T a_i} - \boldsymbol{c_1^T c_1} = 0, \end{array} \tag{7}$$

and all $\boldsymbol{\Delta_i}$, $\forall i = 1, \ldots, N$, are either $\mathbf{0}$ or rank 2 symmetric matrices (sum of two zero matrices if $\boldsymbol{a_i} = \boldsymbol{c_1}$ or the sum of two rank-1 matrices otherwise) we conclude that the two eigenvalues of $\boldsymbol{\Delta_i}$ are equal in magnitude and therefore

$$||\boldsymbol{\Delta_i}||_2^2 = \frac{||\boldsymbol{\Delta_i}||_F^2}{2}. \tag{8}$$

Thus we obtain a bound on the 2-norm of the error matrix

$$\begin{array}{rl} ||\boldsymbol{\Delta}||_2^2 \leq & N\sum_{i=1}^N (1 - (\boldsymbol{c_1^T a_i})^2) \\ \leq & N(1 - T^2). \end{array} \tag{9}$$

The bounds on the two considered norms are in practice too pessimistic since most of the clustered items are very close to the centroid and only a few actually reach the maximum allowed distance. Because of this, we construct an average case error bound replacing the dot product $\boldsymbol{c_1^T a_i}$ with the real-valued random variable Z and conclude that

$$\begin{array}{rl} \mathbb{E}\left[||\boldsymbol{\Delta}||_2^2\right] \leq & N\mathbb{E}\left[1 - Z^2\right] \\ = & N(1 - \mathbb{E}\left[Z\right]^2 - \mathrm{Var}\left[Z\right]) \\ \approx & N(1 - \mathbb{E}\left[Z\right]^2). \end{array} \tag{10}$$

The resulting bounds are much stronger than the ones in (6) and (9) because usually $\mathbb{E}\left[Z\right]^2 \gg T^2$ and $\mathrm{Var}\left[Z\right] \approx 0$ because in general the items are grouped closely around the centroid and the maximum allowed error is close to 1, $0 \ll T < 1$.

Concerning the full description of the random variable introduced Z, taking into account that all data items grouped with threshold T are contained in the n-sphere of volume $V_n(R) = \frac{\pi^{n/2}}{\Gamma(n/2+1)}R^n$ and of radius $R_{\max} = \max\{\sqrt{\boldsymbol{v_i^T v_i}}\} = \sqrt{2(1-T)}$, for $R = 0, \ldots, R_{\max}$ its cumulative distribution function is

$$F(R) = \frac{R^n}{\sqrt{[2(1-T)]^n}}. \tag{11}$$

In the case of a single centroid grouping, a potentially useful observation is that since $\mathrm{tr}(\boldsymbol{\Delta_i}) = 0$, $\forall i = 1, \ldots, N$ and the trace operator is linear, this means that $\mathrm{tr}(\boldsymbol{\Delta}) = 0$ and furthermore because $\mathrm{rank}(\boldsymbol{\Delta}) \approx 2$ (the matrix has two dominant singular values because it is the sum of outer products between two sets of highly correlated vectors) it follows that

$$||\boldsymbol{\Delta}||_2^2 \approx \frac{||\boldsymbol{\Delta}||_F^2}{2}. \tag{12}$$

The results obtained in this subsection are strong because they bound the expectation operator $\mathbb{E}$ and they target the two most widely used matrix norms (Frobenius and the 2-norm). The results are intuitive because the absolute errors increase with the number of items clustered by the each centroid, since each new item clustered adds extra variation to the group (if it is different from the centroid), and the threshold T appears squared since we are interested in the product $\boldsymbol{AA^T}$.

3.2. **Bound on the singular vectors.** Of course, we are mostly interested in the difference of direction that might appear in the singular vectors associated with the largest singular value of the training matrix, before and after T-mindot is applied.

In order to analyze this effect, considering the two systems

$$\begin{aligned} \boldsymbol{BB}^T\boldsymbol{x_1} = &\ \ \lambda_1\boldsymbol{x_1} \\ \boldsymbol{AA}^T\boldsymbol{y_1} = &\ \ \sigma_1\boldsymbol{y_1}. \end{aligned} \tag{13}$$

where λ_1 is the largest singular value of $\boldsymbol{B}$ and $\boldsymbol{x_1}$ its corresponding eigenvector and σ_1 is the largest singular value of $\boldsymbol{A}$ and $\boldsymbol{y_1}$ its corresponding eigenvector. We are interested in the dot product $|\boldsymbol{x_1^T y_1}|$ to be as large as possible (ideally 1). For the sake of simplicity, consider that all vectors in the training matrix $\boldsymbol{A}$ where clustered around the same centroid $\boldsymbol{c_1}$ leading to a rank-1 matrix $\boldsymbol{B}$.

Consider that

$$\begin{aligned} \boldsymbol{y_1} = &\ \ \boldsymbol{x_1} + \boldsymbol{\delta x_1} \\ \sigma_1 = &\ \ \lambda_1 + \delta\lambda_1 \\ \boldsymbol{AA}^T = &\ \ \boldsymbol{BB}^T - \boldsymbol{\Delta}. \end{aligned} \tag{14}$$

plug into the second equation of (13), expand the terms and take into account that $\boldsymbol{BB}^T = \lambda_1\boldsymbol{x_1}$ to get

$$\begin{aligned} (\boldsymbol{BB}^T - \boldsymbol{\Delta})(\boldsymbol{x_1} + \boldsymbol{\delta x_1}) = &\ \ (\lambda_1 + \delta\lambda_1)(\boldsymbol{x_1} + \boldsymbol{\delta x_1}) \\ \boldsymbol{BB}^T\boldsymbol{\delta x_1} - \boldsymbol{\Delta x_1} - \boldsymbol{\Delta\delta x_1} = &\ \ \lambda_1\boldsymbol{\delta x_1} + \delta\lambda_1\boldsymbol{x_1} + \delta\lambda_1\boldsymbol{\delta x_1}. \end{aligned} \tag{15}$$

Let $\boldsymbol{\delta x_1}$ to be a linear combination of the orthonormal basis formed by the eigenvectors of $\boldsymbol{BB}^T$

$$\boldsymbol{\delta x_1} = \sum_{j=1}^{n} \epsilon_{1j}\boldsymbol{x_j}, \tag{16}$$

with coefficients $|\epsilon_{1j}| \ll 1$ and plug into (15) to get

$$\epsilon_{11}\lambda_1\boldsymbol{x_1} - \boldsymbol{\Delta x_1} - \boldsymbol{\Delta}\sum_{j=1}^{n} \epsilon_{1j}\boldsymbol{x_j} = \lambda_1\sum_{j=1}^{n} \epsilon_{1j}\boldsymbol{x_j} + \delta\lambda_1\boldsymbol{x_1} + \delta\lambda_1\sum_{j=1}^{n} \epsilon_{1j}\boldsymbol{x_j}. \tag{17}$$

Multiplications on the right of (17) with $\boldsymbol{x_1^T}$ and $\boldsymbol{x_k^T}$ respectivly yield

$$\begin{aligned} \delta\lambda_1 = &\ \ -\frac{\boldsymbol{x_1^T\Delta x_1} - \boldsymbol{x_1^T\Delta}\sum_{j=1}^{n} \epsilon_{1j}\boldsymbol{x_j}}{1+\epsilon_{11}} \\ \epsilon_{1k} = &\ \ -\frac{\boldsymbol{x_k\Delta x_1} + \boldsymbol{x_k^T\Delta}\sum_{j=1}^{n} \epsilon_{1j}\boldsymbol{x_j}}{\lambda_1}, \forall k \geq 2. \end{aligned} \tag{18}$$

The expression for the desired dot product is derived from

$$\begin{aligned} \boldsymbol{y_1} =& \ \ \boldsymbol{x_1} + \delta \boldsymbol{x_1} \\ \boldsymbol{y_1} =& \ \ \boldsymbol{x_1} + \textstyle\sum_{k=1}^{n} \epsilon_{1k} \boldsymbol{x_k} \\ (\boldsymbol{x_1}^T \boldsymbol{y_1})^2 =& \ \ (1 + \epsilon_{11})^2. \end{aligned} \tag{19}$$

and it is clear that the main focus is to compute ϵ_{11}. To reach an expression for ϵ_{11}, start the development from

$$\begin{aligned} &\boldsymbol{y_1}^T \boldsymbol{y_1} &&= 1 \\ &(\boldsymbol{x_1} + \delta\boldsymbol{x_1})^T (\boldsymbol{x_1} + \delta\boldsymbol{x_1}) &&= 1 \\ &1 + 2\boldsymbol{x_1}^T \delta\boldsymbol{x_1} + \delta\boldsymbol{x_1}^T \delta\boldsymbol{x_1} &&= 1 \\ &2\boldsymbol{x_1}^T \delta\boldsymbol{x_1} + \delta\boldsymbol{x_1}^T \delta\boldsymbol{x_1} &&= 0 \\ &2\epsilon_{11} + \textstyle\sum_{k=1}^{n} \epsilon_{1k}^2 &&= 0 \\ &\epsilon_{11}^2 + 2\epsilon_{11} + \textstyle\sum_{k=2}^{n} \epsilon_{1k}^2 &&= 0. \end{aligned} \tag{20}$$

Substitute and expand, taking into account that $\lambda_1 = N$, $\lambda_k = 0, \forall k \geq 2$ (since $\mathrm{rank}(\boldsymbol{BB}^T) = 1$), $\boldsymbol{x_1} = \boldsymbol{c_1}$, $\boldsymbol{x_k}^T \boldsymbol{c_1} = 0, \forall k \geq 2$ and denote $\boldsymbol{a_i}^T \boldsymbol{c_1} = Z$ (where Z is a real-valued random variable) following the same idea as in the previous section to get

$$\begin{aligned} \epsilon_{1k} =& \ \ -\frac{\boldsymbol{x_k}^T (\sum_{j=2}^{N} \boldsymbol{c_1} \boldsymbol{v_j}^T + \boldsymbol{v_j} \boldsymbol{a_j}^T) \boldsymbol{x_1}}{\lambda_1} \\ =& \ \ -\frac{\boldsymbol{x_k}^T \sum_{j=2}^{N} \boldsymbol{v_j} \boldsymbol{a_j}^T \boldsymbol{x_1}}{\lambda_1} \end{aligned} \tag{21}$$

and then use it to compute the free term in the last equation of (20)

$$\begin{aligned} \mathbb{E}\,[S] =& \ \ \mathbb{E}\,[\textstyle\sum_{k=2}^{n} \epsilon_{1k}^2] \\ =& \ \ \textstyle\sum_{k=2}^{n} \mathbb{E}\,[\frac{Z^2}{N^2} (\boldsymbol{x_k}^T \sum_{j=1}^{N} \boldsymbol{v_j})^2] \\ =& \ \ \textstyle\sum_{k=2}^{n} \mathbb{E}\,[\frac{Z^2}{N^2} (\boldsymbol{x_k}^T \sum_{j=1}^{N} \boldsymbol{v_j})^2] \\ =& \ \ \mathbb{E}\,[\tfrac{Z^2}{N^2} (|| \textstyle\sum_{j=1}^{N} \boldsymbol{v_j} ||_2^2 - (\boldsymbol{x_1}^T \sum_{j=1}^{N} \boldsymbol{v_j})^2)] \\ \leq& \ \ \mathbb{E}\,[\tfrac{Z^2}{N^2} (N^2 2(1 - Z) - N^2 (1 - Z)^2)] \\ =& \ \ \mathbb{E}\,[Z^2 (1 - Z)(1 + Z)] \\ =& \ \ \mathbb{E}\,[Z^2 (1 - Z^2)], \end{aligned} \tag{22}$$

where we used the facts:

$$\begin{aligned} \mathbb{E}\,[(\boldsymbol{x_1}^T \textstyle\sum_{j=1}^{N} \boldsymbol{v_j})^2] =& \ \ \mathbb{E}\,[(\boldsymbol{c_1}^T \textstyle\sum_{j=1}^{N} (\boldsymbol{c_1} - \boldsymbol{a_j}))^2] \\ =& \ \ \mathbb{E}\,[(\textstyle\sum_{j=1}^{N} (1 - \boldsymbol{c_1}^T \boldsymbol{a_j}))^2] \\ =& \ \ \mathbb{E}\,[N^2 (1 - Z)^2], \end{aligned} \tag{23}$$

$$\begin{aligned} \mathbb{E}\,[|| \textstyle\sum_{j=1}^{N} \boldsymbol{v_j} ||_2] \leq& \ \ \mathbb{E}\,[\textstyle\sum_{j=1}^{N} || \boldsymbol{v_j} ||_2] \\ =& \ \ \mathbb{E}\,[\textstyle\sum_{j=1}^{N} \sqrt{2(1 - \boldsymbol{c_1}^T \boldsymbol{a_i})}] \\ =& \ \ \mathbb{E}\,[\textstyle\sum_{j=1}^{N} \sqrt{2(1 - Z)}] \\ =& \ \ \mathbb{E}\,[N \sqrt{2(1 - Z)}]. \end{aligned} \tag{24}$$

Return to (20) to reach the final expression

$$\epsilon_{11} = \ \ -1 \pm \sqrt{1 - S}. \tag{25}$$

Therefore

$$\begin{aligned} (\boldsymbol{x_1}^T\boldsymbol{y_1})^2 = & \ 1 - S \\ \geq & \ 1 - Z^2(1 - Z^2) \\ \geq & \ 1 - T^2(1 - T^2). \end{aligned} \tag{26}$$

where the last inequality is the worst case bound. The expected value of the dot product has the final form

$$\begin{aligned} \mathbb{E}\left[(\boldsymbol{x_1}^T\boldsymbol{y_1})^2\right] = & \ \mathbb{E}\left[1 - S\right] \\ \geq & \ \mathbb{E}\left[1 - Z^2(1 - Z^2)\right] \\ = & \ 1 - \mathbb{E}\left[Z^2\right] + \mathbb{E}\left[Z^4\right] \\ = & \ 1 - \mathbb{E}\left[Z\right]^2 + \mathbb{E}\left[Z^2\right]^2 - \mathrm{Var}\left(Z\right) + \mathrm{Var}\left[Z^2\right] \\ \geq & \ \mathbb{E}\left[Z\right]^2, \end{aligned} \tag{27}$$

since $\mathbb{E}\left[1 - Z^2 + Z^4\right] = \mathbb{E}\left[Z^2 + (Z^2 - 1)^2\right] \geq \mathbb{E}\left[Z^2\right]$ following that the variance terms are approximately zero, because in general the items are grouped closely around the centroid and the maximum allowed error is close to 1, $0 \ll T < 1$.

The result described in this subsection bounds the actual difference in the singular vectors before and after the clustering procedure T-mindot is applied to the dataset. The result works only in the case when all the data items are grouped around the same centroid but since it is a good result it offers some intuition that in the general case things work similarly.

4. Experimental results

The demonstrate the potential of T-mindot we describe in this section numerical experiments and results obtained on a popular image dataset [8].

We extract from the images all the 8×8 non-overlapping patches and scale everything in the range $[-1, 1]$ with the DC component removed. Everything is columnwise concatenated in the matrix $\boldsymbol{A} \in \mathbb{R}^{n \times N}(n = 64, N = 10^5)$ and normed to 1. This acts as the test dataset that is used in all the simulations.

The tests run in the following manner: a fixed number $\tilde{N}$ of test vectors are extract from the whole dataset $\boldsymbol{A}$ and the T-mindot clustering procedure is applied to reduce the dataset to $\boldsymbol{B} \in \mathbb{R}^{n \times M}$ before the K-SVD algorithm is applied to train a sparse linear model of the data. We are interested in two important performance indicators: the reduction achieved by T-mindot (which leads to the actual speed-up) and the quality of the representation achieved by the dictionary computed on the reduced dataset when used on the original dataset.

All simulations are executed on the original dataset and on the reduced dataset. In order to measure the speed-up, the tables show separately the running times of T-mindot and the K-SVD training algorithm applied on the full original extracted dataset of size $\tilde{N}$ and the reduced dataset of size M denoted by $T_{\boldsymbol{A}}$ and $T_{\boldsymbol{B}}$ respectively. In the analogous way, we define the representation errors reached on the original dataset by both trained dictionaries $E_{\boldsymbol{A}}$ and $E_{\boldsymbol{B}}$.

Table 1

Simulation results for T=0.9.

$\tilde{N}$	M	M/N	$T_{\text{T-mindot}}$	$T_{\boldsymbol{A}}$	$T_{\boldsymbol{B}}$	$(T_{\text{T-mindot}} + T_{\boldsymbol{B}})/T_{\boldsymbol{A}}$	$E_{\boldsymbol{A}}$	$E_{\boldsymbol{B}}$
5000	1577	31.54%	2.0	256	190	75.00%	11.55	12.21
10000	3046	30.46%	3.5	339	214	64.15%	17.73	18.28
30000	7649	25.49%	5.3	687	119	18.09%	31.79	32.85
50000	11835	23.67%	9.6	1052	148	14.98%	41.70	43.42
70000	15693	22.41%	21.3	1358	353	27.56%	49.04	50.81
100000	21122	21.12%	29.7	1978	551	29.35%	59.11	60.32

Table 2

Simulation results for T=0.95.

$\tilde{N}$	M	M/N	$T_{\text{T-mindot}}$	$T_{\boldsymbol{A}}$	$T_{\boldsymbol{B}}$	$(T_{\text{T-mindot}} + T_{\boldsymbol{B}})/T_{\boldsymbol{A}}$	$E_{\boldsymbol{A}}$	$E_{\boldsymbol{B}}$
5000	2140	42.80%	1.8	256	170	67.10%	11.55	11.88
10000	3982	39.82%	3.6	339	199	59.76%	17.73	17.67
30000	10766	35.88%	5.8	687	187	28.06%	31.79	32.24
50000	17246	34.49%	11.3	1052	342	33.58%	41.70	42.45
70000	23037	32.91%	24.0	1358	377	29.52%	49.04	49.42
100000	31441	31.44%	44.2	1978	501	27.56%	59.11	59.25

Table 3

Simulation results for T=0.99.

$\tilde{N}$	M	M/N	$T_{\text{T-mindot}}$	$T_{\boldsymbol{A}}$	$T_{\boldsymbol{B}}$	$(T_{\text{T-mindot}} + T_{\boldsymbol{B}})/T_{\boldsymbol{A}}$	$E_{\boldsymbol{A}}$	$E_{\boldsymbol{B}}$
5000	2683	53.66%	2.5	256	190	75.19%	11.55	11.76
10000	5369	53.69%	4.0	339	250	74.92%	17.73	17.77
30000	15588	51.96%	8.3	687	424	62.92%	31.79	31.91
50000	25153	50.30%	18.2	1052	615	60.19%	41.70	41.74
70000	34502	49.28%	35.5	1358	694	53.71%	49.04	49.08
100000	47828	47.82%	75.8	1978	1022	55.50%	59.11	58.75

In all situations, the K-SVD algorithm starts with a random initialization to design dictionaries of 256 atoms with target sparsity $k_0 = 6$. It runs for a maximum of 80 iterations and it stops earlier if the relative error between two consecutive iterations drops below 10^{-5}, since no significant progress can be achieved any longer.

The numerical simulations are depicted in the following tables for various runs of T-mindot with different reduction thresholds: 0.9, 0.95 and 0.99. The running time and the representation errors obtained on the original dataset are copied in all tables for an easy comparison.

From the three presented tables it is very clear that the threshold parameter of T-mindot has a crucial impact on the speed of the overall training procedure (the running times are 2-3 time smaller for most of the cases) and only a minor effect on the representation errors (there is only a small decrease in the errors as the threshold is increased). Also, the actual application of T-mindot with a high threshold does not seem to impact negatively the representation errors since the computed values are very close to the ones computed by training the dictionaries on the full dataset. Simulations show that smaller values of the threshold (eg. 0.7 or 0.8) lead to a rapid degradation of the representation errors.

5. Conclusions

This papers describes a grouping procedure that reduces the dimension of a training set before it is use in the context of overcomplete dictionary design. The proposed method is tested by applications on image data and the results show conclusively the speed up achieved. Additionally, the paper presents theoretical bounds that characterize the error introduced by the reduction method.

6. Acknowledgements

This work was supported by the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE-2011-3-0400 and by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/88 /1.5/S/61178.

REFERENCES

[1] *A. M. Bruckstein, D. L. Donoho, and M. Elad*, From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images, SIAM Review, Vol. 51, pp. 34–81, 2009.

[2] *R. Neff and A. Zakhor*, Very Low Bit-Rate Video Coding Based on Matching Pursuits, IEEE Trans. Circuits and Systems, Vol. 7, pp. 158–171, 1997.

[3] *Y. Zang, S. Mei, Q. Chen and Z. Chen*, A novel image/video coding method based on Compressed Sensing theory, IEEE International Conference on Acoustics Speech and Signal Processing, pp. 1361 – 1364, 2008.

[4] *M. Aharon, M. Elad and A. Bruckstein*, K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation, IEEE Trans. Signal Processing, Vol. 54, pp. 4311–4322, 2006.

[5] *S. G. Mallat and Z. Zhang*, Matching Pursuits With Time-Frequency Dictionaries, IEEE Trans. Signal Processing, Vol. 41, pp. 3397–3415, 1993.

[6] *R. Rubinstein, M. Zibulevsky and M. Elad*, Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit Technical Report - CS Technion, 2008.

[7] *C. Rusu*, Clustering before training large datasets - case study: K-SVD, European Signal Processing Conference (EUSIPCO), pp. 2188–2192, 2012.

[8] Yale Face Database. [Online]. Available: http://cvc.yale.edu/projects/ yalefaces/yalefaces.html.