Subspaces Clustering Approach to Lossy Image Compression

Przemysław Spurek[1,*], Marek Śmieja[1,**], and Krzysztof Misztal[2,***]

[1] Jagiellonian University
Faculty of Mathematics and Computer Science
Łojasiewicza 6, 30-348 Kraków, Poland
{przemyslaw.spurek,marek.smieja}@ii.uj.edu.pl
[2] AGH University of Science and Technology
Faculty of Physics and Applied Computer Science
al. A. Mickiewicza 30, 30-059 Kraków, Poland
Krzysztof.Misztal@fis.agh.edu.pl

Abstract. In this contribution lossy image compression based on subspaces clustering is considered. Given a PCA factorization of each cluster into subspaces and a maximal compression error, we show that the selection of those subspaces that provide the optimal lossy image compression is equivalent to the 0-1 Knapsack Problem. We present a theoretical and an experimental comparison between accurate and approximate algorithms for solving the 0-1 Knapsack problem in the case of lossy image compression.

Keywords: lossy compression, image compression, subspaces clustering.

1 Introduction

The vector quantization is the basic approach to lossy image compression [1–4]. The procedure relies on encoding a possibly large set of points from a multidimensional vector space into a finite set of values from a discrete subspace of lower dimension. Clustering algorithms are widely used in vector quantization [5–7]. In such cases the effect of the compression depends strictly on the selection of the clustering algorithm. In this paper we consider a special case of subspaces clustering [8–12] based on Principal Component Analysis (PCA) [13–15]. We focus on finding the division of data and clusters representation which have the highest possible level of compression and minimal error (loss of image quality).

[*] The work of this author was supported by the National Centre of Science (Poland) [grant no. 2013/09/N/ST6/01178].

[**] The work of this author was supported by the Polish Ministry of Science and Higher Education from the budget for science in the years 2013–2015 [grant no. IP2012 055972].

[***] The work of this author was supported by the National Centre of Science (Poland) [grant no. 2012/07/N/ST6/02192].

K. Saeed and V. Snášel (Eds.): CISIM 2014, LNCS 8838, pp. 571–579, 2014.

We assume that a group of points $S \subset \mathbb{R}^N$ is compressed by its orthogonal projection onto a subspace generated by n principals components [16–18], i.e. the subspace spanned on n eigenvectors $\{v_1, \dots, v_n\}$ associated with the n highest eigenvalues of the covariance matrix $\Sigma = \mathrm{cov}(S)$ shifted by the mean $\mathrm{m} = \mathrm{mean}(S)$. The compression error is given by the sum of squared distances between the points and their orthogonal projections [8, 19]:

$$\mathrm{E}(S; n) \; = \sum_{x \in S} \left(\sum_{i=1}^{n} \mathrm{dist}(x; \mathrm{m} + \mathrm{span}(v_1, \dots, v_i))^2 \right)^{1/2},$$

where $\mathrm{dist}(x; \mathrm{m} + \mathrm{span}(v_1, \dots, v_i))$ denotes the distance between the point x and the subspace $\mathrm{m} + \mathrm{span}(v_1, \dots, v_i)$.

Consequently, given k-clusters $S_1, \dots, S_k$ and the dimensions $n_1, \dots, n_k$ of subspaces that are used for projection in appropriate clusters, the compression error equals

$$\mathrm{E}(S; n_1, \dots, n_k) \; = \sum_{j=1}^{k} \mathrm{E}(S_j; n_j).$$

In the case of image compression, the objective is to cluster a dataset for which the total compression error does not exceed ε, i.e:

$$\mathrm{E}(S; n_1, \dots, n_k) \leq \varepsilon,$$

and the number of parameters used to store the compressed data

$$\sum_{j=1}^{k} n_j \cdot \#S_j$$

is minimal, where $\#S_i$ denotes the cardinality of cluster S_i. In [8], where (k, ω)–means is presented, the authors proposed a possible solution for the selection of subspaces. The method is based on choosing the eigenvectors associated with the largest eigenvalues regardless of the cluster membership.

In this paper we show that the aforementioned optimization problem can be transformed into the 0-1 Knapsack Problem and that the solution proposed in [8] realizes its greedy approximation. Moreover, we consider an exact solution constructed with the use of dynamic programming algorithm which for the case of lossy image compression gives a slightly better results than the greedy method. An experimental study conducted on standard images of sizes 512 × 512 [20] showed that both approaches work in a comparable computation time. Therefore, it is more preferable to apply the dynamic algorithm. However, for high resolution images, the exact method can be numerically inefficient.

2 Image Compression

In this section we define a problem of lossy image compression based on PCA. We then show how to transform it into the 0-1 Knapsack Problem and present two approaches to solving it.

Suppose that a data-set S is divided into k clusters $S_1,\dots,S_k$. Every cluster S_i is represented by a subspace $\mathrm{V}^k = \mathrm{m}^k + \mathrm{span}(v_1^k,\dots,v_N^k)$ where $\mathrm{m}^k = \mathrm{mean}(S_k)$ and $v_1^k,\dots,v_N^k$ are eigenvectors of $\mathrm{cov}(S_k)$ ordered increasingly respectively to corresponding eigenvalues $\lambda_1^k,\dots,\lambda_N^k$. Such a representation can be obtained by applying PCA for every cluster S_i.

In order to compress the image, $n_i \leq N$ principal components are chosen for each cluster S_i, and vectors are projected onto constructed n_i dimensional spaces. The error associated with one cluster after projecting its elements onto n_i principal components can be calculated with the use of the following proposition.

Proposition 1. *Let S be a subset of $\mathbb{R}^N$ and let $n < N$. By $\{\lambda_1,\dots,\lambda_N\}$ we denote the increasingly ordered eigenvalues corresponding to eigenvectors $\{v_1,\dots,v_N\}$ of covariance matrix* $\mathrm{cov}(S)$. *Then*

$$E(S;n) = \sum_{i=n+1}^{N} \#S \cdot \lambda_i.$$

Proof Compare with [21, Propetries A1-A5].

Given k clusters $S_1,\dots,S_k$ the total compression error after projecting data onto appropriate $n_1,\dots,n_k$ dimensional subspaces is given by

$$\mathrm{E}(S;n_1,\dots,n_k) = \sum_{j=1}^{k} \left(\#S_j \sum_{i=n_i+1}^{N} \lambda_i^j \right).$$

Let $\varepsilon > 0$ denote the maximal compression error allowed. We seek the minimal number of parameters to describe the image for which the overall compression error does not exceed ε:

Problem 1. *Let $\varepsilon > 0$ be given. Find the dimensions $n_1,\dots,n_k$ of clusters $S_1,\dots,S_k$, such that the total compression error does not exceed ε, i.e.*

$$\sum_{j=1}^{k} \sum_{i=n_i+1}^{N} \#S_j \cdot \lambda_i^j < \varepsilon$$

and which minimize the number of parameters, i.e.

$$\min_{n_1,\dots,n_k} \{\sum_{j=1}^{k} n_j \cdot \#S_j\} = \min_{n_1,\dots,n_k} \{\sum_{j=1}^{k} \sum_{i=1}^{n_j} \#S_j\}.$$

In [8] the authors proposed a method to select the subspaces dimensions. In general, their idea is based on choosing the eigenvectors related to the largest eigenvalues. This is not the optimal solution. Since the compression error is bounded by $E(S;0,\dots,0)$ we transform the above minimization problem into an equivalent maximization one:

Problem 2. *Let $\varepsilon > 0$ be given. Find the dimensions $n_1, \dots, n_k$ of clusters $S_1, \dots, S_k$, such that*

$$\sum_{j=1}^{k} \sum_{i=n_j+1}^{N} \#S_j \cdot \lambda_i^j < \varepsilon$$

and which maximize

$$\sum_{j=1}^{k} \sum_{i=n_j+1}^{N} \#S_j.$$

This is the 0-1 Knapsack Problem. For a proper illustration of this matter, let us define the items parameters for the Knapsack Problem. The weights and values of $N = k \cdot n$ items are defined as follows:

$$w_{(i-1)n+j} = w_{i,j} = \#S_j \cdot \lambda_i^j,$$

$$v_{(i-1)n+j} = v_{i,j} = \#S_j.$$

The goal of the 0-1 Knapsack Problem is to select those items which maximize the overall profit and do not exceed the knapsack capacity, i.e. to define numbers $k_l \in \{0, 1\}$ which maximize:

$$\sum_{l=1}^{N} k_l v_l, \text{ subject to } \sum_{l=1}^{N} k_l w_l \le \varepsilon.$$

Plenty of strategies have been proposed for the 0-1 Knapsack Problem which is NP-hard with respect to the number of items [22]. The greedy approach finds an approximated solution and relies on choosing elements ordered with respect to the highest density v_l/w_l. In the case of compression, it depends on sorting with respect to decreasing eigenvalues:

$$\frac{v_{i,j}}{w_{i,j}} = \frac{\#S_j}{\#S_j \cdot \lambda_i^j} = \frac{1}{\lambda_i^j}.$$

It is easily seen that this is exactly the method proposed in [8].

If m is the maximum value of items that fit into the knapsack (in the optimal solution), the greedy algorithm is guaranteed to achieve at least an overall value of items equal $m/2$ [23]. However, this is not a common situation in the image compression – more often the solution returned by both algorithms is similar. This can be seen in the following example.

Example 2. Let the cardinalities of all clusters be the same, i.e.

$$c := \#S_i = \#S_j, \text{ for all } i, j = 1 \dots, k.$$

Then both algorithms return identical items. Indeed, we maximize

$$\sum_{j=1}^{k} \sum_{i=n_j+1}^{n} 1, \text{ subject to } \sum_{j=1}^{k} \sum_{i=n_j+1}^{n} \lambda_i^j < \frac{\varepsilon}{c}$$

Since all items are equally valuable, the optimal solution includes the lightest items. This strategy is also preferred by the greedy algorithm.

Also, many algorithms which construct an exact solution exist, e.g. the dynamic programming method. More precisely, let us denote by $F(l,v)$ the minimal overall weight of elements chosen from 1 to l such that their overall value is maximal and at least $v \geq 0$, i.e.

$$F(l,v) = \min_{k_1,\ldots,k_l} \left\{ \sum_{i=1}^{l} k_i w_i : \sum_{i=1}^{l} k_i v_i \geq v \right\}$$

for $l = 0, \ldots, N$, $v = 0, \ldots, V$, where $V = \sum_{l=1}^{N} v_l$ (we assume that $l = 0$ means that no items are included into knapsack – then $F(0,0) = 0$ and $F(0,v) = \infty$, for $v > 0$). The maximal value of items included in the knapsack is denoted by:

$$C^* := \max\{v : F(N,v) \leq \varepsilon\}.$$

This value can be calculated in a recursive procedure:

$$F(l,v) = \begin{cases} 0 & \text{, for } l = 0 \text{ and } v = 0 \\ \infty & \text{, for } l = 0 \text{ and } v > 0 \\ \min\{F(l-1,v), F(l-1,v-v_l)+w_l\}, \\ \qquad\qquad \text{for } l = 1,\ldots,N, \end{cases}$$

and is realized by a bottom up algorithm. The complexity equals $\Theta(N \cdot V) = \Theta(k \cdot n \cdot \#S)$, which for large datasets is quite high.

3 Experiments

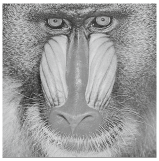
In this section we present the results of numerical experiments illustrating the performance of the lossy image compression based on subspaces clustering with the use of the greedy and dynamic approaches.

We apply classical methods often used in such situations: PCA, k–means ($k = 5$) with PCA representation for each cluster and (k,ω)–means ($k = 5$ and ω with non zero elements on 11-15 coordinates). Table 1 contains the results of these compression methods with the use of greedy and dynamic algorithms. Dynamic approach delivers slightly better results for all methods except for the PCA method because only one cluster was considered. This is a consequence of Example 2.3. Both of these algorithms worked in a similar time. Moreover, (k,ω)–means algorithm gave the best results. Therefore, the further experiments will be performed with the use of this method.

Figure 1 presents sample compression results for the classical Lena image. Given the maximal compression error, the qualities of images are comparable for both Knapsack algorithms, while the number of parameters varies greatly. The advantage of using the dynamic programming algorithm is evident.

The comparison of the errors achieved by the greedy and the dynamic algorithms for the Lena image is showed in Figure 2. Since the dynamic algorithm constructs the optimal solution, the compression error is greater than in the

Table 1. Parameters needed to obtain the desired error level for compression of a few sample images using (k, ω)–means, PCA and k–means with PCA methods. We compared the greedy and dynamic strategies for choosing the optimal compression configuration. Among all, (k, ω)–means provides better compression level than other methods.

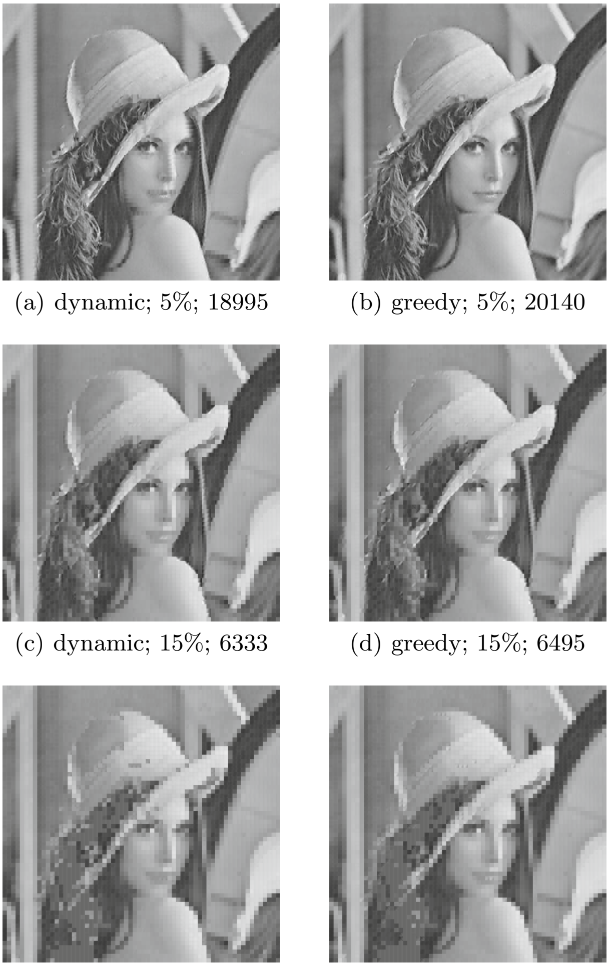
		Parameters					
	Error	(k, ω)–means		PCA		k-means	
		Dynamic	Greedy	Dynamic	Greedy	Dynamic	Greedy
	1%	96 543	96 903	105 625	105 625	179 737	180 304
	5%	18 995	20 140	21 125	21 125	48 313	48 948
	10%	9 866	10 028	12 675	12 675	24 547	24 676
	15%	6 333	6 495	8 450	8 450	15 723	16 582
	25%	3 859	4 021	4 225	4 225	8 678	9 194
	50%	2 801	2 963	4 225	4 225	3 541	3 850
	1%	316 086	316 325	401 375	401 375	481 563	481 617
	5%	103 642	103 721	143 650	143 650	240 836	241 061
	10%	46 555	46 873	67 600	67 600	147 632	147 882
	15%	24 099	24 178	33 800	33 800	104 404	104 441
	25%	9 064	9 143	12 675	12 675	59 618	60 850
	50%	2 064	2 536	8 450	8 450	16 312	17 123
	1%	125 330	125 940	143 650	143 650	307681	308705
	5%	21 283	21 283	25 350	25 350	83386	83788
	10%	11 816	12 264	12 675	12 675	39869	40560
	15%	8 158	8 938	8 450	8 450	24997	25444
	25%	4 489	4 489	8 450	8 450	15048	15658
	50%	977	977	4 225	4 225	5224	5224

case of the greedy approach (the results are closer to the black line). Moreover, the graph is smoother. In the case of the greedy algorithm, the ordering of eigenvalues is performed once for all error levels. Consequently, the graph is constant in the subsequent intervals. Clearly, the dynamic algorithm provides better compression level than the greedy solution (Figure 3).

The presented experiments confirmed that the dynamic approach delivers better results than the greedy one. The differences are especially evident for large dimension images that contain complicated patterns.

4 Conclusions

Subspaces clustering algorithms are very often used for image compression. In such a situation elements from each group are represented by the orthogonal projection onto low dimensional subspaces. The crucial problem lies in determining such subspaces that minimize the use of memory and do not exceed arbitrarily given loss of image quality. In this paper we showed that this optimization problem can be transformed into the 0-1 Knapsack Problem. Moreover, two possible solutions, the exact and the approximated one, were presented. Consequently,



(a) dynamic; 5%; 18995

(b) greedy; 5%; 20140

(c) dynamic; 15%; 6333

(d) greedy; 15%; 6495

(e) dynamic; 30%; 3859

(f) greedy; 30%; 4021

Fig. 1. Compressed Lena image. Description of each image consists of: the name of the compression algorithm, the level of compression error and the number of parameters needed for compression. The algorithm using the dynamic approach for compression needs less parameters.

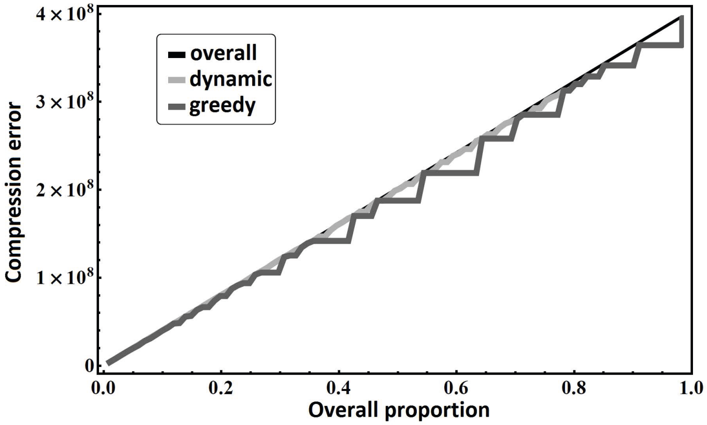


Fig. 2. Compression error level for the Lena image. The overall compression error level (black line) is compared with the compression error level realized by using the dynamic and the greedy algorithm. The error of compression with the use of the dynamic approach is closer to the overall one.

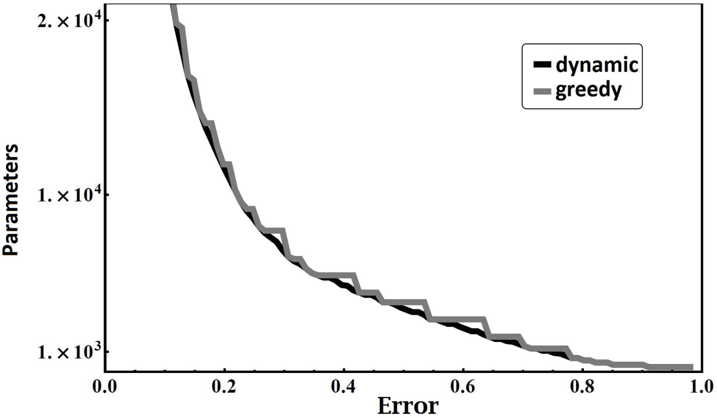


Fig. 3. Number of parameters needed to reach the desired error level. Number of parameters decreases with increasing compression error. Generally, the dynamic algorithm provides better compression level.

the method from [8] realizes a greedy approximation of the 0-1 Knapsack Problem. Experiments performed on standard images showed that both algorithms work in a similar computation time.

References

1. Gray, R.: Vector quantization. IEEE ASSP Magazine 1(2), 4–29 (1984)
2. Scheunders, P.: A genetic lloyd-max image quantization algorithm. Pattern Recognition Letters 17(5), 547–556 (1996)

3. Lin, Y.-C., Tai, S.-C.: A fast linde-buzo-gray algorithm in image vector quantization. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 45(3), 432–435 (1998)
4. Gray, R.M., Neuhoff, D.L.: Quantization. IEEE Transactions on Information Theory 44(6), 2325–2383 (1998)
5. Equitz, W.H.: A new vector quantization clustering algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing 10(37), 1568–1575 (1989)
6. Scheunders, P.: A genetic c-means clustering algorithm applied to color image quantization. Pattern Recognition 30(6), 859–866 (1997)
7. Chou, C.-H., Su, M.-C., Lai, E.: A new cluster validity measure and its application to image compression. Pattern Analysis and Applications 7(2), 205–220 (2004)
8. Spurek, P., Tabor, J., Misztal, K.: Weighted Approach to Projective Clustering. In: Saeed, K., Chaki, R., Cortesi, A., Wierzchoń, S. (eds.) CISIM 2013. LNCS, vol. 8104, pp. 367–378. Springer, Heidelberg (2013)
9. Agarwal, P.K., Mustafa, N.H.: k-means projective clustering. In: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 155–165. ACM (2004)
10. Kriegel, H.P., Kröger, P., Zimek, A.: Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from Data (TKDD) 3(1), 1–58 (2009)
11. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter 6(1), 90–105 (2004)
12. Vidal, R.: Subspace clustering. IEEE Signal Processing Magazine 28(2), 52–68 (2011)
13. Jolliffe, I.: Principal component analysis. Wiley Online Library (2005)
14. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics 2(4), 433–459 (2010)
15. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and Intelligent Laboratory Systems 2(1), 37–52 (1987)
16. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 245–250. ACM (2001)
17. Ifarraguerri, A., Chang, C.-I.: Unsupervised hyperspectral image analysis with projection pursuit. IEEE Transactions on Geoscience and Remote Sensing 38(6), 2529–2538 (2000)
18. Kaarna, A., Zemcik, P., Kalviainen, H., Parkkinen, J.: Compression of multispectral remote sensing images using clustering and spectral reduction. IEEE Transactions on Geoscience and Remote Sensing 38(2), 1073–1082 (2000)
19. Grim, J.: Multimodal discrete karhunen-loève expansion. Kybernetika 22(4), 329–330 (1986)
20. USC-SIPI, USC-SIPI image database, http://sipi.usc.edu/database/
21. Jolliffe, I.: Principal component analysis. In: Encyclopedia of Statistics in Behavioral Science (2002)
22. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations, John Wiley & Sons, Inc (1990)
23. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack problems. Springer (2004)