Making Archetypal Analysis Practical

Christian Bauckhage[1,2] and Christian Thurau[1]

[1] Fraunhofer IAIS, Sankt Augustin, Germany
[2] B-IT, University of Bonn, Bonn, Germany
{christian.bauckhage,christian.thurau}@iais.fraunhofer.de

Abstract. Archetypal analysis represents the members of a set of multivariate data as a convex combination of extremal points of the data. It allows for dimensionality reduction and clustering and is particularly useful whenever the data are superpositions of basic entities. However, since its computation costs grow quadratically with the number of data points, the original algorithm hardly applies to modern pattern recognition or data mining settings. In this paper, we introduce ways of notably accelerating archetypal analysis. Our experiments are the first successful application of the technique to large scale data analysis problems.

1 Introduction

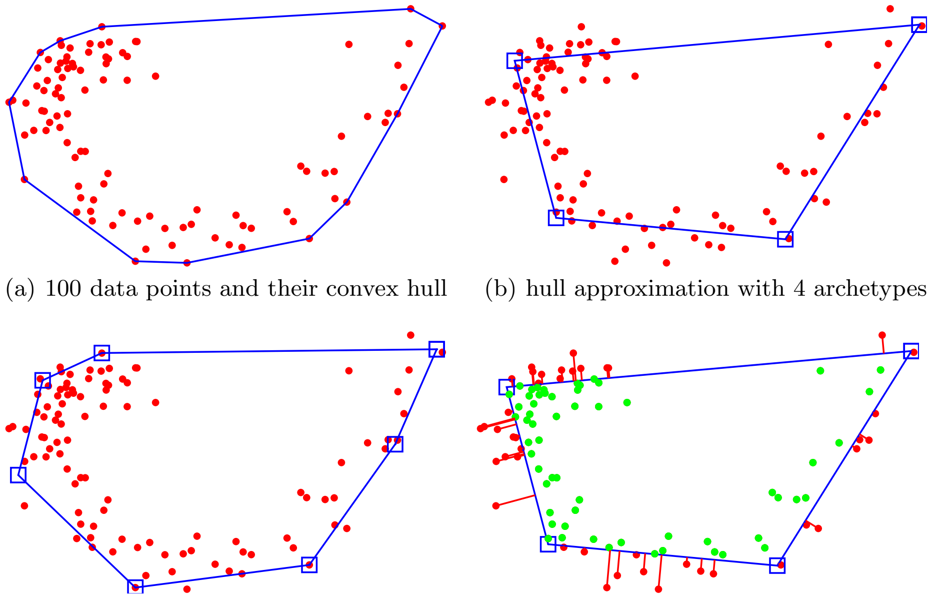
Archetypal Analysis (AA) was introduced by Cutler and Breiman [1] as a new way of dimensionality reduction for multivariate data. The basic idea is to approximate each point in a data set as a *convex combination* of a set of *archetypes.* The archetypes themselves are restricted to being sparse mixtures of individual data points and are thus supposed to be easily interpretable by human experts. This contrasts with familiar techniques such as (kernel) PCA [2,3] where the resulting basis elements often lack physical meaning. And while NMF [4,5] yields characteristic parts, AA yields archetypal composites.

In order to identify suitable archetypes, Cutler and Breiman minimize the squared error in representing each data point as a mixture of archetypes (see Fig. 1). They show that minima are attained if the archetypes are extreme data points lying on the convex hull of the data. Their minimization algorithm consists of an alternating least squares procedure where each iteration requires the solution of several constrained quadratic optimization problems.

Representations based on convex combinations of archetypal elements offer interesting possibilities for pattern recognition. As the coefficient vectors of a convex combination reside in a simplex, AA lends itself to subsequent soft clustering, probabilistic ranking, or classification using latent class models. So far, however, AA has found application in physics and biology [6,7,8] but did not prevail as a commodity tool for pattern analysis or classification. We assume this to be a consequence of the complexity of the algorithm proposed in [1].

In this paper, we briefly review AA, discuss some of its characteristics, and point out why it scales quadratically with the size of a data set. The main contribution of this paper is presented in section 3: We propose a *working set*

J. Denzler, G. Notni, and H. Süße (Eds.): DAGM 2009, LNCS 5748, pp. 272–281, 2009.



(a) 100 data points and their convex hull

(b) hull approximation with 4 archetypes

(c) hull approximation with 7 archetypes

(d) points contributing to RSS

Fig. 1. Archetypal analysis approximates the convex hull of a set of data. Increasing the number p of archetypes improves the approximation. Solutions found for different choices of p do not necessarily nest; for instance, none of the 4 archetypes in (b) reoccurs among the 7 archetypes in (c). While points inside of an approximated convex hull can be represented exactly as a convex combination of archetypes, points on the outside are represented by their nearest point on the archetype hull. Suitable archetypes result from iteratively minimizing the residuals of the points outside of the hull (d).

mechanism for AA that accelerates the procedure. In addition, we introduce a workable way of preselecting auspicious archetypal candidates and thus gain further speed up. In section 4, we apply our accelerated AA algorithm to large image collections. To the best of our knowledge, this is the first time that archetypal analysis is being applied to data sets consisting of tens of thousands of elements rather than of just several dozens. Finally, a summary concludes this paper.

2 Archetypal Analysis

Suppose that we are given a set of data $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where $\mathbf{x}_i \in \mathbb{R}^m$. Archetypal analysis (AA) deals with finding a set of archetypes $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_p\}$ with $p \ll n$ that are linear combinations of the data points

$$\mathbf{z}_j = \sum_{i=1}^{n} \mathbf{x}_i b_{ij} \tag{1}$$

where the coefficients $b_{ij} \geq 0$ so that the archetypes *resemble* the data and $\sum_i b_{ij} = 1$ so that they are *convex mixtures* of the data. Then, for a given choice of archetypes, AA minimizes

$$\left\|\mathbf{x}_i - \sum_{j=1}^{p} \mathbf{z}_j a_{ji}\right\|^2 \tag{2}$$

to determine coefficients a_{ji} that allow the data $\mathbf{x}_i$ to be *well represented* by the archetypes. Again, AA imposes the constraints $a_{ji} \geq 0$ so that each data point is a *meaningful* combination of archetypal elements and $\sum_j a_{ji} = 1$ so that the data points are represented as *mixtures* of archetypes. Therefore, a suitable choice of archetypes $\{\mathbf{z}_1, \ldots, \mathbf{z}_p\}$ minimizes the residual sum of squares

$$\mathrm{RSS}(p) \;=\; \sum_{i=1}^{n}\left\|\mathbf{x}_i - \sum_{j=1}^{p} \mathbf{z}_j a_{ji}\right\|^2 \;=\; \sum_{i=1}^{n}\left\|\mathbf{x}_i - \sum_{j=1}^{p}\sum_{k=1}^{n} \mathbf{x}_k b_{kj} a_{ji}\right\|^2. \tag{3}$$

For our discussion in this paper, it is more convenient to write (3) as a matrix equation. To this end, we collect the data points $\mathbf{x}_i \in \mathbb{R}^m$ in an $m \times n$ matrix $\mathbf{X}$ and the archetypes $\mathbf{z}_j \in \mathbb{R}^m$ in an $m \times p$ matrix $\mathbf{Z}$ and cast (3) as

$$\mathrm{RSS}(p) \;=\; \left\|\mathbf{X} - \mathbf{Z}\mathbf{A}\right\|^2 \;=\; \left\|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}\right\|^2 \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{p \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times p}$ are column stochastic matrices.

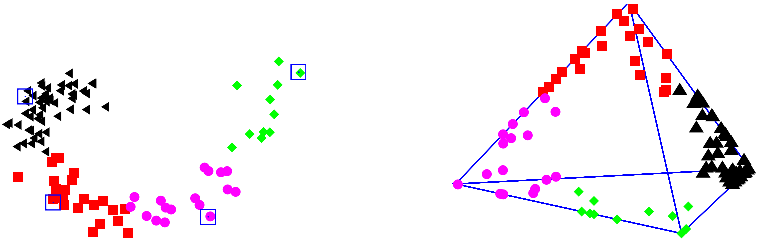
Computing an archetypal representation therefore requires the constrained optimization of two sets of coefficients $\{a_{ji}\}$ and $\{b_{ij}\}$. To accomplish this, Cutler and Breiman [1] present an alternating least squares algorithm which we discuss below; first, however, we summarize some of the characteristics of AA.

2.1 Properties of Archetypal Analysis

In [1], Cutler and Breiman prove that, for $p > 1$, the archetypes $\{\mathbf{z}_1, \ldots, \mathbf{z}_p\}$ are located on the data convex hull. For $p = 1$, the unique minimizer of (3) is the sample mean and for $p = 2$, the vector $\mathbf{v} = \mathbf{z}_1 - \mathbf{z}_2$ corresponds to the first principal axis of the data. If $q \leq n$ points define the convex hull of the data and $p = q$, the global minimizers of (3) are exactly those q points. Increasing the number of archetypes therefore improves the approximation of the data convex hull (see Fig. 1).

Cutler and Breiman point out that archetypes do not nest and need not be orthogonal (see Fig. 1). Once suitable archetypes $\{\mathbf{z}_1, \ldots, \mathbf{z}_p\}$ have been determined, every data point can either be exactly represented or approximated by a convex combination of the $\mathbf{z}_j$ (see Fig. 1). Since $a_{ji} \geq 0$ and $\sum_j a_{ji} = 1$ each data point can be interpreted as a distribution over the archetypes. Therefore, AA readily allows for soft clustering or classification since the coefficients a_{ji} of a data point $\mathbf{x}_i$ can be interpreted as probabilities $p(\mathbf{x}_i|\mathbf{z}_j)$ indicating membership to classes represented by the archetypes $\mathbf{z}_j$ (see Fig. 2).

Since (3) generally does not have a closed form solution, one must resort to optimization. Cutler and Breiman point out that careful initialization improves the speed of convergence and lowers the risk of finding insignificant archetypes.



(a) archetypal clusters

(b) simplex projection

Fig. 2. Example of using AA for clustering; each point $\mathbf{x}_i$ is assigned to an archetype $\mathbf{z}_k$ using $k = \operatorname{argmax}_j a_{ji}$. The coefficient vectors $\mathbf{a}_i$ of the data $\mathbf{x}_i$ are stochastic vectors and therefore reside in a simplex whose vertices correspond to the archetypes $\mathbf{z}_k$.

2.2 The Archetype Algorithm

In order to solve (3) for optimal coefficients a_{ji} and b_{ij}, Cutler and Breiman propose an alternating least squares procedure. Given an initial guess of the archetypes $\{\mathbf{z}_1, \ldots, \mathbf{z}_p\}$, their method iterates the following steps:

1.) determine coefficients a_{ji} by solving n constrained problems as in (2); in matrix notation we have: min $\left\|\mathbf{Z}\mathbf{a}_i - \mathbf{x}_i\right\|^2$ s.t. $a_{ji} \geq 0$ and $\sum_j a_{ji} = 1$. To point out the computational complexity of this step, we recast these n problems as

$$\begin{aligned} \min \ & \frac{1}{2}\mathbf{a}_i^T \mathbf{Q} \mathbf{a}_i - \mathbf{q}^T \mathbf{a}_i \ , \ i = 1, \ldots, n \\ \text{s.t. } & \mathbf{I}\, \mathbf{a}_i \geq \mathbf{0} \\ & \mathbf{1}^T \mathbf{a}_i = 1 \end{aligned} \tag{5}$$

where $\mathbf{Q} = \mathbf{Z}^T\mathbf{Z}$ is a $p \times p$ matrix and $\mathbf{q} = \mathbf{Z}^T\mathbf{x}_i$ is a p-vector.

2.) given the updated a_{ji}, compute intermediate archetypes that account for the update, i.e. solve (2) for the $\mathbf{z}_j$ to obtain $\tilde{\mathbf{Z}} = \mathbf{X}\mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T\right)^{-1}$.

3.) determine the coefficients b_{ij} as the minimizers of p constrained problems min $\left\|\mathbf{X}\mathbf{b}_j - \tilde{\mathbf{z}}_j\right\|^2$ s.t. $b_{ij} \geq 0$ and $\sum_i b_{ij} = 1$ or equivalently

$$\begin{aligned} \min \ & \frac{1}{2}\mathbf{b}_j^T \mathbf{R} \mathbf{b}_j - \mathbf{r}^T \mathbf{b}_j \ , \ j = 1, \ldots, p \\ \text{s.t. } & \mathbf{I}\, \mathbf{b}_j \geq \mathbf{0} \\ & \mathbf{1}^T \mathbf{b}_j = 1 \end{aligned} \tag{6}$$

where $\mathbf{R} = \mathbf{X}^T\mathbf{X}$ is a $n \times n$ matrix and $\mathbf{r} = \mathbf{X}^T\tilde{\mathbf{z}}_j$ is a corresponding n-vector.

4.) update the archetypes by setting $\mathbf{Z} = \mathbf{X}\mathbf{B}$

5.) compute the new RSS; unless it falls below a threshold or only marginally improves the old RSS, continue with ***1.)***
Apparently, computation and memory costs of this algorithm do not primarily depend on the dimension m of the data but are dominated by the optimization problems of the order of $O(n^2)$ in step ***3.)*** where n denotes the size of the data set. Given the growing amount of data that characterizes modern data analysis problems, naïvely implementing AA therefore impedes its application in most practical settings. Next, we suggest improvements to alleviate this.

3 Making Archetypal Analysis Practical

Both our modifications of the original AA algorithm exploit the fact that suitable archetypes reside on the data convex hull. In a nutshell, the basic idea is that, since archetypes are sparse mixtures of data points, data points $\mathbf{x}_i$ inside of the convex hull do not contribute to these mixtures. Therefore, the corresponding coefficients b_{ij} need not be estimated but can be set to 0.

3.1 Focusing on a Working Set

Data contained within the convex hull of a set of archetypal estimates do not contribute to the residual that is being minimized by the archetype algorithm (compare again Fig. 1). In each iteration of the algorithm, the data set can therefore be decomposed into $X = X^+ \cup X^-$ where $X^- = \{\mathbf{x}_i \in X | \mathbf{x}_i = \mathbf{Z}\mathbf{a}_i\}$ and $X^+ = \{\mathbf{x}_i \in X | \mathbf{x}_i \neq \mathbf{Z}\mathbf{a}_i\}$. That is X consists of a set of points that can be represented exactly as a convex combination of archetypes and a *working set* (cf. [9]) containing points that can only be approximated. Applying a suitable permutation, the matrix $\mathbf{X}$ in (4) then reads $\mathbf{X} = \left[\mathbf{X}^+ \, \mathbf{X}^-\right]$ where $\mathbf{X}^+$ and $\mathbf{X}^-$ are $m \times n'$ and $m \times (n - n')$ matrices, respectively, and $n' < n$.

Under this premise the residual in (4) becomes

$$\begin{aligned} \left\|\mathbf{X} - \mathbf{Z}\mathbf{A}\right\|^2 &= \left\|\left[\mathbf{X}^+ \, \mathbf{X}^-\right] - \mathbf{Z}\left[\mathbf{A}^+ \, \mathbf{A}^-\right]\right\|^2 \\ &= \underbrace{\left\|\mathbf{X}^+ - \mathbf{Z}\mathbf{A}^+\right\|^2}_{\neq 0} + \underbrace{\left\|\mathbf{X}^- - \mathbf{Z}\mathbf{A}^-\right\|^2}_{=0} \end{aligned} \tag{7}$$

and after expanding $\mathbf{Z} = \mathbf{X}\mathbf{B}$ it further reduces to

$$\begin{aligned} \left\|\mathbf{X}^+ - \mathbf{Z}\mathbf{A}^+\right\|^2 &= \left\|\mathbf{X}^+ - \left[\mathbf{X}^+ \, \mathbf{X}^-\right] \begin{bmatrix}\mathbf{B}^+ \\ \mathbf{B}^-\end{bmatrix} \mathbf{A}^+\right\|^2 \\ &= \left\|\mathbf{X}^+ - \left(\mathbf{X}^+\mathbf{B}^+ + \mathbf{X}^-\mathbf{B}^-\right)\mathbf{A}^+\right\|^2 \\ &= \left\|\mathbf{X}^+ - \mathbf{X}^+\mathbf{B}^+\mathbf{A}^+\right\|^2. \end{aligned} \tag{8}$$

Here, the last step exploits that $\mathbf{X}^-$ only contains data points inside of the convex hull of the currently estimated archetypes; as the archetypes themselves are mixtures of data points on the convex hull of $\mathbf{X}$, the data points in $\mathbf{X}^-$ do not contribute to $\mathbf{Z}$, which, in turn, is tantamount to $\mathbf{B}^- = \mathbf{0}$.

Consequently, the effort required in the third step of the archetype algorithm reduces to $O(n'^2) < O(n^2)$. Moreover, as the algorithm improves archetypal estimates, the number of points outside of their convex hull decreases. This also decreases the size of the optimization problems and automatically accelerates the algorithm in later iterations.

3.2 Preselecting Archetypal Candidates

The overall gain in speed that can be achieved by the above approach depends on the initial choice of archetypes. If these were close to data convex hull, already the first couple of iterations of the algorithm would have to consider small working sets only. In fact, if we were to know the points on the convex hull of $\mathbf{X}$, we could restrict the optimization procedure to just these points. Unfortunately, the so called Upper Bound Theorem (cf. e.g. [10]) states that the worst case combinatorial complexity of computing the convex hull of n points in m dimensions is $\Theta(n^{(m/2)})$. Although more sophisticated methods for computing the convex hull of m dimensional data exist, the problem seems ill posed for typical pattern recognition tasks such as the analysis of image databases.

Our second contribution in this paper consists in a workable solution that avoids this problem. Instead of trying to compute the data convex hull directly, we propose to consider a sub-sample $\mathbf{X}^H$ of points a on the convex hull of $\mathbf{X}$. Since the optimization procedure in archetypal analysis will usually converge to approximations of the optimal choice of archetypes, we effectively narrow the number of possible solutions. This step is crucial for making archetypal analysis practical for very large data sets, i.e. cases where $n > 50000$.

To obtain $\mathbf{X}^H$ we exploit that the original data matrix $\mathbf{X}$ contains only finitely many points so that its convex hull forms a polytope in $\mathbb{R}^m$. The main theorem of polytope theory states that every image of a polytope P under an affine map $\pi : \mathbf{x} \rightarrow \mathbf{M}\mathbf{x} + \mathbf{t}$ is a polytope [11]. In particular, every vertex of an affine image of P corresponds to a vertex of P. This allows us to sample the convex hull of $\mathbf{X}$ as the union of points found on convex hulls of different 2D projections of the data. We project the data onto the $\frac{h(h-1)}{2}$ 2D subspaces spanned by pairwise combinations of the first h eigenvectors of the covariance matrix of $\mathbf{X}$ where h is chosen such that the first h eigenvectors account for 95% of the data variation.

In our experiments with computer vision benchmark data, we found that the number of points n'' obtained from several 2D projections is much smaller than the set size n. Note that this is not a general property of convex hulls. On the contrary in extremely high dimensions all points of a normally distributed data set come to lie on the hull [12,13]. The fact that our experiments always revealed a computational complexity $O(n''^2) \ll O(n'^2)$ (see Fig. 4 for an exemplary quantitative analysis on synthetic 3D data) indicates that, at least in the feature spaces we considered, large collections of natural images are not normally distributed but reside on lower dimensional manifolds of different structure.

```
Input: data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$
Output: matrix of archetypes $\mathbf{Z} \in \mathbb{R}^{m \times p}$ and coefficient matrices $\mathbf{A}$ and $\mathbf{B}$

preselect archetypal candidates $\mathbf{X}^H$
initialize matrices $\mathbf{Z}$, $\mathbf{A}$, and $\mathbf{B}$, and compute $\mathrm{RSS}_{t=0}$
repeat
    optimize $\mathbf{A} = \min_{\mathbf{A}} \|\mathbf{X}^H - \mathbf{Z}\mathbf{A}\|^2$ s.t. $a_{ji} \geq 0$ and $\sum_j a_{ji} = 1$
    determine working set $X^+$ and matrices $\mathbf{X}^+$, $\mathbf{A}^+$, and $\tilde{\mathbf{Z}}^+$ and set $\mathbf{B}^- = \mathbf{0}$
    optimize $\mathbf{B}^+ = \min_{\mathbf{B}^+} \|\tilde{\mathbf{Z}}^+ - \mathbf{X}^+\mathbf{B}^+\|^2$ s.t. $b_{ij} \geq 0$ and $\sum_i b_{ij} = 1$
    update the archetypes $\mathbf{Z} = \mathbf{X}^+\mathbf{B}^+$
until $\mathrm{RSS}_{t+1} < \theta$ or $|\mathrm{RSS}_{t+1} - \mathrm{RSS}_t| < \epsilon$
```

Fig. 3. Summary of the archetype algorithm combining both proposed accelerations

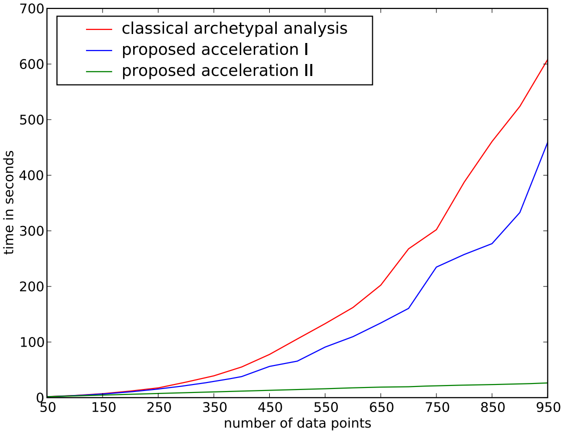


Fig. 4. Runtime behavior of archetypal analysis applied to a growing number n of 3D points. All three variant scale quadratically with n. The version confined to points sampled from the convex hull clearly outperforms the other two.

4 Application Examples

In order to verify the practical applicability of the proposed extensions to archetypal analysis, we analyzed two large data sets from the image analysis domain. The first data set consist of sequences of human silhouettes showing various activities [14]. The second data set consists of more than 50.000 images downloaded from *flickr*. For both data sets, we followed the proposal in [15] and re-scaled the RGB/Gray-value images to a resolution of 32 × 32 pixels. Visualizations of the archetypes we found are shown in Figs. 5 through 8. The computation times we measured were reasonable; 50 iterations on the *flickr* data took less than an hour using a Python implementation applying the `cvxopt` optimization library by Dahl and Vandenberghe (`http://abel.ee.ucla.edu/cvxopt/`). Note that, to

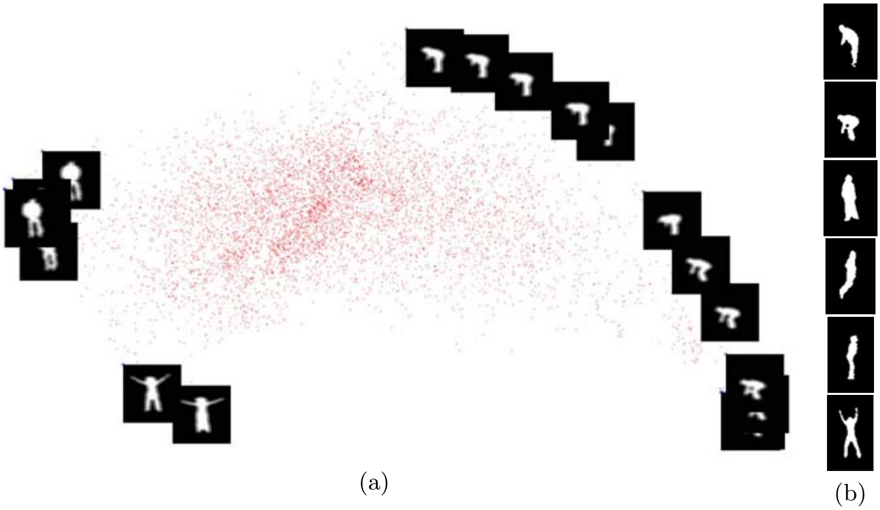


Fig. 5. (a) 2D projection of the Weizman set containing 5.000 body poses; points on the convex hull are shown as pictures. (b) 6 archetypal poses extracted from the data.

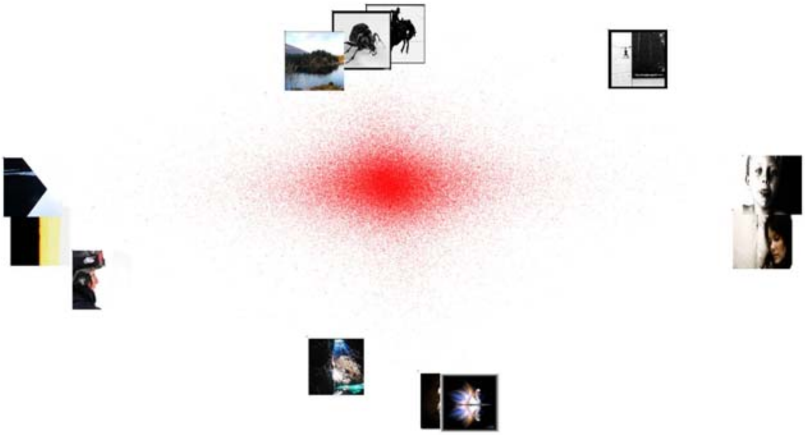


Fig. 6. 2D projections of 50.000 images retrieved from *flickr*; points located on the convex hull are shown as pictures

our knowledge, these are the largest data sets processed with archetypal analysis so far. Following our suggested initialization and optimization steps, the approach scales to millions of images since it no longer depends on the overall set size but rather on the number of data points sampled from the convex hull.

Interestingly, the archetypes found among the *flickr* images display a geometric similarity to the Gabor filters that are found among the principal components of natural images [16]. They show prominent vertical, horizontal, or diagonal line patterns, or they feature blob-like elements in the center of the image. This

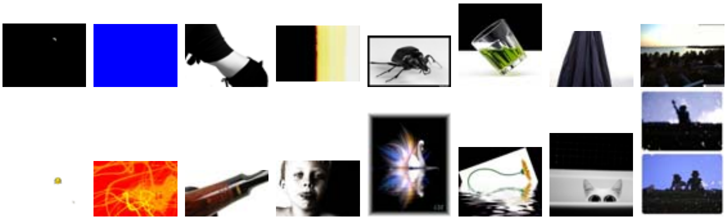


Fig. 7. 16 archetypes determined from a data set of 50.000 *flickr* images

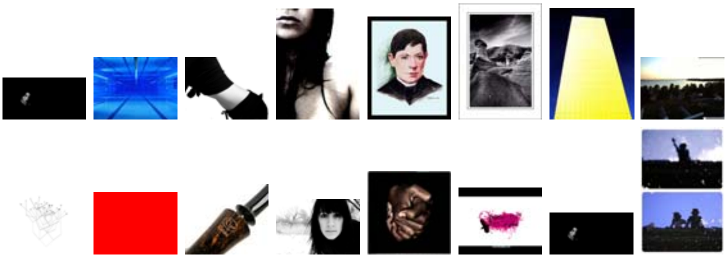


Fig. 8. 16 archetypes resulting from a different initialization of the algorithm. While these archetypes are not completely identical to the ones in Fig. 7, they show similar global geometric structures or brightness gradients.

suggests that the extremal points in this large collection of natural images are located close to the principal axes of the data. Since we do not observe this behavior for the pose images, this finding does not seem to be an artefact of AA but rather a phenomenon of the statistics of natural images.

Moreover, for the *flickr* images, we find the archetypes to be rather distant from the vast majority of the data. We currently investigate whether this is a boon or a bane. On the one hand, AA is affected by outliers. On the other hand, the notion of an outlier is not that clearly defined for a large set of natural images. While they are extreme in that they show stark contrasts and dominant structures, none of the archetypes we found can be considered an abnormal picture. Also, representing a data set by means of sparse convex combinations over the members of the set is of course best accomplished, if the basis elements are extreme. Whether or not this improves clustering or content-based classification is examined in an ongoing study.

5 Summary

Archetypal analysis represents each point in a data set as a convex combination of a set of archetypes which themselves are sparse mixtures of individual data points. Unlike most familiar dimensionality reduction or clustering techniques, archetypal analysis therefore yields basis elements that are readily interpretable by human experts.

Optimal archetypes reside on the data convex hull and are determined through a constrained quadratic optimization process. In this paper, we suggested two modifications of the original algorithm in order to notably speed up its runtime. We exploit that archetypes are sparse convex combinations of extremal elements of the data and only apply the procedure to working sets of correspondingly reduced sizes. Consequently, archetypal analysis becomes applicable to a wide range of realistic data analysis problems for it can now cope with data sets whose sizes exceed several hundred elements. The results we presented in this paper are, to the best of our knowledge, the first instances of successful application of archetypal analysis to several tens of thousands of data points.

References

1. Cutler, A., Breiman, L.: Archetypal Analysis. Technometrics 36(4), 338–347 (1994)
2. Jolliffe, I.: Principal Component Analysis. Springer, Heidelberg (1986)
3. Schölkopf, B., Smola, A.J., Müller, K.-R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10(5), 1299–1319 (1998)
4. Lee, D.D., Seung, S.: Learning the Parts of Objects by Non-Negative Matrix Factorization. Nature 401(6755), 788 (1999)
5. Finesso, L., Spreij, P.: Approximate Nonnegative Matrix Factorization via Alternating Minimization. In: Proc. 16th Int. Symp. on Mathematical Theory of Networks and Systems, Leuven (July 2004)
6. Stone, E., Cutler, A.: Archetypal Analysis of Spatio-temporal Dynamics. Physica D 90(3), 209–224 (1996)
7. Chan, B.H.P.: Archetypal Analysis of Galaxy Spectra. Monthly Notices of the Royal Astronomical Society 338(3), 790–795 (2003)
8. Huggins, P., Pachter, L., Sturmfels, B.: Toward the Human Genotope. Bulletin of Mathematical Biology 69(8), 2723–2735 (2007)
9. Joachims, T.: Making Large-Scale Support Vector Machine Learningn Practical. In: Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge (1999)
10. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry. Springer, Heidelberg (2000)
11. Ziegler, G.M.: Lectures on Polytopes. Springer, Heidelberg (1995)
12. Donoho, D.L., Tanner, J.: Neighborliness of Randomly-Projected Simplices in High Dimensions. Proc. of the Nat. Academy of Sciences 102(27), 9452–9457 (2005)
13. Hall, P., Marron, J., Neeman, A.: Geometric representation of high dimension low sample size data. J. of the Royal Statistical Society B 67(3), 427–444 (2005)
14. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. In: Proc. ICCV (2005)
15. Torralba, A., Fergus, R., Freeman, W.T.: 80 Million Tiny Images: A Large Dataset for Non-parametric Object and Scene Recognition. IEEE Trans. on Pattern Analalysis and Machine Intelligence 30(11), 1958–1970 (2008)
16. Heidemann, G.: The principal components of natural images revisited. IEEE Trans. on Pattern Analalysis and Machine Intelligence 28(5), 822–826 (2006)