Chapter 8
Dimensionality Reduction Optimization

8.1 Introduction

Optimization in continuous solution spaces can be a tedious task. Various methods have been proposed to allow the optimization in multimodal fitness landscapes. In this chapter, we introduce a further method based on dimensionality reduction. The search takes place in a redundant solution space $\mathbb{R}^{\hat{d}}$ with a higher dimensionality $\hat{d} > d$ than the original solution space. To allow the fitness function evaluation, the high-dimensional candidate solution representation must be mapped back to the solution space with a dimensionality reduction mapping $F : \mathbb{R}^{\hat{d}} \rightarrow \mathbb{R}^{d}$. We call this approach dimensionality reduction evolution strategy (DR-ES). Employing more dimensions than the original solution space specification requires can be beneficial in some solution spaces. The search in the original solution space may be more difficult due to local optima and unflattering solution space characteristics that are not present in the high-dimensional pendants. The assumption of the DR-ES is that the additional dimensions offer a degree of freedom that can better be exploited with the usual evolutionary operators, i.e., intermediate recombination and self-adaptive Gaussian mutation in the case of the (μ, λ)-ES. The mapping from $\mathbb{R}^{\hat{d}}$ to $\mathbb{R}^{d}$ is computed after a new population has been generated. Any dimensionality reduction method is potentially a good choice for the dimensionality reduction process. We concentrate on PCA, which assumes linearity between variables. But the approach is not restricted to PCA and can be combined with any other point-wise dimensionality reduction approach.

The approach is related to the concept of bloat, which is the phenomenon that parts of the genome do not encode for functional meaningful parts of species. However, bloat is supposed to play an important role in the evolutionary development. Evolution gets the freedom to develop the unused parts that are not under evolutionary pressure to potentially useful functional parts. By introducing more variables for the optimization process than actually necessary to encode the problem, the redundancy is supposed to support overcoming local optima and difficult solution space conditions.

O. Kramer, *Machine Learning for Evolution Strategies*,
Studies in Big Data 20, DOI 10.1007/978-3-319-33383-0_8

This chapter is structured as follows. In Sect. 8.2, we introduce the dimensionality reduction problem. In Sect. 8.3, we briefly present the idea of PCA. The DR-ES approach is introduced in Sect. 8.4. Related work is presented in Sect. 8.5. Section 8.6 presents an experimental analysis of the DR-ES. Conclusions are presented in Sect. 8.7.

8.2 Dimensionality Reduction

Due to an enormous growth of the number of sensors installed in various computer science-oriented domains, Big Data became a relevant research issue in the last years. The sensor resolution has grown steadily leading to the situation of very large data sets with high-dimensional patterns. Dimensionality reduction has an important part to play in solving these Big Data challenges. Putting it in a more formal definition, the dimensionality reduction problem is to find low-dimensional representations $\mathbf{x}_i \in \mathbb{R}^d$ of high-dimensional patterns $\hat{\mathbf{x}}_i \in \mathbb{R}^{\hat{d}}$ for $i = 1, \ldots, N$. To tackle this challenge, numerous dimensionality reduction techniques have been proposed in the past. Famous ones are self-organizing maps introduced by Kohonen [1]. PCA is an excellent method for linear data. It will be introduced in Sect. 8.3. If the data is non-linear, methods like ISOMAP [2] and locally linear embedding (LLE) [3] may be the proper choice. In Chap. 9, we will focus on a non-linear dimensionality reduction method to visualize high-dimensional optimization processes.

Dimensionality reduction approaches perform a mapping F of patterns from a high-dimensional space to a low-dimensional space while maintaining important information. Distance information and neighborhoods should be preserved. For point-wise embeddings, this means that for each high-dimensional pattern, a vector with less dimensions is computed that servers as its low-dimensional pendant. A frequent approach that is known as feature selection concentrates on the search on a smaller number of features. Based on quality measures like entropy or on pure blackbox search, feature selection methods reduce the number of features simply by leaving them out. A good feature selection method finds an optimal combination of features to solve a particular machine learning problem. Due to the curse of dimensionality problem, see Chap. 4, the concentration on a subset of features is an important pre-processing step. Evolutionary search can be employed to select the optimal feature combination. Examples are introduced for various domains, see [4].

8.3 Principal Component Analysis

PCA by Pearson [5, 6] is a method for linear dimensionality reduction. It computes principal components from the training data set. Given a set of $\hat{d}$-dimensional patterns $\hat{\mathbf{x}}_i \in \mathbb{R}^{\hat{d}}$ with $i = 1, \ldots, N$, the objective of PCA is to find a linear manifold of a

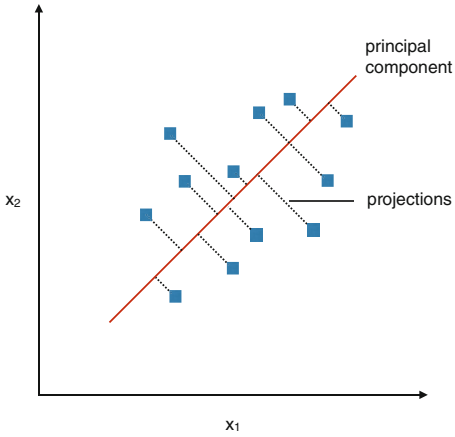


Fig. 8.1 Illustration of PCA. The first principal component captures the largest possible variance. Each further component employs the next highest variance and must be orthogonal to the preceding components

lower dimension $d < \hat{d}$ that captures the most variance of the patterns. Figure 8.1 illustrates the PCA concept. For this sake, PCA computes the covariance matrix of the patterns

$$\mathbf{C} = \frac{1}{N-1} \sum_{i=1}^{N} (\bar{\mathbf{x}} - \hat{\mathbf{x}})(\bar{\mathbf{x}} - \hat{\mathbf{x}})^T \tag{8.1}$$

with mean

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \hat{\mathbf{x}}_i . \tag{8.2}$$

If $\lambda_1 \geq \cdots \geq \lambda_{\hat{d}}$ are the eigenvalues of the covariance matrix $\mathbf{C}$, and if $\mathbf{e}_1, \ldots, \mathbf{e}_{\hat{d}}$ are the corresponding eigenvectors, we can define a $\hat{d} \times d$-matrix

$$\mathbf{V} = [\mathbf{e}_1, \ldots, \mathbf{e}_d]. \tag{8.3}$$

With this matrix, the mapping $\mathbf{F} : \mathbb{R}^{\hat{d}} \rightarrow \mathbb{R}^d$,

$$\mathbf{F}(\hat{\mathbf{x}}_i) = \mathbf{V}^T (\hat{\mathbf{x}}_i - \bar{\mathbf{x}}) \tag{8.4}$$

from data space to the d-dimensional space can be performed. The inverse mapping

$$\mathbf{f}(\mathbf{x}_i) = \mathbf{V}\mathbf{x}_i + \bar{\mathbf{x}} \tag{8.5}$$

back to the $\hat{d}$-dimensional data space is the projection of pattern $\mathbf{x}_i$ onto the linear manifold.

Often, data does not show linear characteristics. Examples are wind time series or image recognition data, where the patterns often live in high dimensions. In this case, non-linear dimensionality reduction methods like ISOMAP and LLE are

good choices. The quality of the dimensionality reduction result can be evaluated with measurements that concentrate on the maintenance of neighborhoods like the co-ranking matrix [7], the nearest neighbor classification error for labeled data [8], or by inspection of visualized embeddings.

SCIKIT- LEARN allows an easy integration of PCA with the steps introduced in the following.

- `from sklearn import decomposition` imports the SCIKIT- LEARN decomposition package that contains PCA variants.
- `decomposition.PCA(...).fit_transform(X)` fits PCA to the list of patterns `X` and maps them to a q-dimensional space. Again, further methods can be employed.

8.4 Algorithm

In this section, we introduce the DR-ES that is based on dimensionality reduction methods that allow the point-wise mapping from $\mathbb{R}^{\hat{d}}$ to $\mathbb{R}^d$ like briefly introduced in the previous section. The DR-ES is a modified self-adaptive population-based (μ, λ)-ES, i.e., it typically employs dominant or intermediate recombination and any form of Gaussian mutation with self-adaptive step size control.

Algorithm 6 shows the DR-ES pseudocode. At the beginning, the candidate solutions $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_\mu \in \mathbb{R}^{\hat{d}}$ and the step sizes $\sigma_1, \ldots, \sigma_\mu$ are initialized. Each solution carries its own step size vector. In the generational loop, λ candidate solutions are produced like in the conventional (μ, λ)-ES. The dimensionality reduction method F, in the experimental part we employ PCA, is applied to the offspring population $\{\hat{\mathbf{x}}'_i\}_{i=1}^{\lambda}$ resulting in d-dimensional solution candidates $\{\mathbf{x}'_i\}_{i=1}^{\lambda}$ that are evaluated on f. In the last step, the $\hat{d}$-dimensional μ best solutions $\{\hat{\mathbf{x}}_i\}_{i=1}^{\mu}$ are selected w.r.t. the fitness the d-dimensional counterparts achieve. The optimization process stops, when a termination condition is met. The ES uses self-adaptive step size control [9], which works as follows. Each solution is equipped with its own step size. It is recombined using the typical recombination operators and is mutated with the log-normal mutation operator

$$\sigma' = \sigma \cdot e^{\tau \mathcal{N}(0,1)} \tag{8.6}$$

and mutation strength τ. As the step sizes are inherited with the solutions, good step sizes spread in the course of evolution.

To summarize, after application of the dimensionality reduction method, a complete individual is a tuple $(\hat{\mathbf{x}}_i, \sigma_i, \mathbf{x}_i, f(\mathbf{x}_i))$ of a high-dimensional *abstract* solution $\hat{\mathbf{x}}_i$, step size σ_i, which may also be a step size vector, depending on the employed Gaussian mutation type, the candidate solution $\mathbf{x}$ in the original solution space, and its fitness $f(\mathbf{x})$.

Algorithm 6 DR-ES

initialize $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_\mu$ and $\sigma_1, \ldots, \sigma_\mu$
repeat
  for $j = 1$ **to** λ **do**
    recombination$(\{\hat{\mathbf{x}}_i\}_{i=1}^{\mu}) \rightarrow \hat{\mathbf{x}}'_j$
    recombination$(\sigma_1, \ldots, \sigma_\mu) \rightarrow \sigma'_j$
    log-normal mutation $\rightarrow \sigma'_j$
    Gaussian mutation $\rightarrow \hat{\mathbf{x}}'_j$
  end for
  dim. red. (PCA) $F(\hat{\mathbf{x}}'_i) \rightarrow \{\mathbf{x}'_i\}_{i=1}^{\lambda}$
  evaluate $\{\mathbf{x}_i\}_{i=1}^{\lambda} \rightarrow \{f(\mathbf{x}_i)\}_{i=1}^{\lambda}$
  select $\{\hat{\mathbf{x}}_i\}_{i=1}^{\mu}$ w.r.t. f
until termination condition

The question how to choose the dimension $\hat{d}$ depends on the problem. In the experimental section, we will experiment with $\hat{d} = 3/2d$, e.g. $\hat{d} = 15$ for $d = 10$. Further method-specific parameters like the neighborhood size of ISOMAP and LLE may have to be chosen according to the optimization problem.

8.5 Related Work

To the best of our knowledge, the evolutionary search in a high-dimensional space with mapping back to the solution space via PCA or other dimensionality reduction methods has not been applied yet.

Related to the idea of employing more dimensions than required for the original problem is related to the discussion mapping from genotypes to phenotypes and of bloat. Often, a function is required that maps the genetic representation of the solution, i.e. the genotype, to the actual solution, i.e. the phenotype. In continuous solution spaces, this is often not required as the genotype can directly be represented as phenotype. A recent contribution to this discussion comes from Simões et al. [10], who employ feed-forward neural networks for defining genotype-phenotype maps of arbitrary continuous optimization problems and concentrate on the analysis of locality and redundancy, and the self-adaptation of the mappings.

The concept of bloat in evolutionary computation is related to our approach of adding additional genes. Bloat are the parts of the genome that are not directly required to encode a solution. This redundancy is supposed to allow walks in solution spaces that overcome local optima and flat fitness landscapes.

Liepins and Vose [11] show that all continuous functions are theoretically easy for EAs given an appropriate representation. But as the mappings are in general unknown and the search space of functions is very large because one has to consider all permutations of mappings, it is difficult to find the optimal representation.

However, representation adaptation on a meta-level is supposed to be attractive in this context.

Wang et al. [12] propose an algorithm that separates the solution space into subspaces by selecting objective variables that are relevant for separate objectives in multi-objective optimization problems. This form of dimensionality reduction is similar to feature selection in machine learning.

To some extend related to the dimensionality reduction idea of the DR-ES is the kernel trick of methods like SVMs. The kernel trick allows implicit operations in a high-dimensional reproducing Hilbert space [13], where, e.g., non-linear data can linearly be separated. The mapping into this kernel feature space is not explicitly computed. Often, only pattern similarities are required that can be computed via the dot product of the original patterns using a kernel function. The DR-ES searches in a high-dimensional abstract space that is mapped back to the low-dimensional solution space.

Decoder functions for constrained solution spaces also share similarities with the DR-ES. Decoder functions map the constrained solution space to an unconstrained, or only box-constrained space, in which the search is easier to carry out. Solutions in this so-called feature space are feasible and the original solutions can easily be repaired. Koziel and Michalewicz [14] employ a homomorphous mapping between the constrained solution space and the one-dimensional cube. Similar, Bremer et al. [15] use kernel functions with support vector description to search for feasible scheduling solutions in the energy domain. To some extend related is the approach by Boschetti [16], who employs LLE to support the evolutionary search with candidate solutions.

Evolutionary computation in dimensionality reduction is also a promising related line of research, e.g., for tuning dimensionality reduction parameters or embedding results with ES. Vahdat et al. [17] proposes to use evolutionary multi-objective algorithms to balance intra-cluster distance and connectedness of clusters. We use a (1+1)-ES to tune clusters of embeddings with ISOMAP, LLE, and PCA by scaling and rotating [18]. In [8], we propose an iterative approach that constructs a dimensionality reduction solution with Gaussian mutation-like steps.

8.6 Experimental Analysis

In the following, the DR-ES is experimentally analyzed. Table 8.1 shows the corresponding results. It shows the mean fitness achieved by a (15,100)-ES and a (15,100)-DR-ES with PCA on Sphere and Rastrigin after 5000 fitness function evaluations. We test the settings $d = 10$ and $d = 20$, while the search takes place in the higher dimensional space $\hat{d} = 15$, and $\hat{d} = 30$, respectively. For example in the case of $d = 20$, the ES searches in the solution space $\mathbb{R}^{30}$, while the PCA maps the best solutions $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_\mu$ back to the original solution space $\mathbb{R}^{20}$. The table shows the medians and corresponding standard deviations of 25 runs for different experimental settings. The results show that the (15,100)-DR-ES with PCA outperforms the (15,100)-ES

Table 8.1 Experimental comparison of (15,100)-ES and (15,100)-DR-ES with PCA on Sphere and Rastrigin

Problem		(15,100)-ES		(15,100)-DR-ES		Wilx.
	$\hat{d}/d$	Median	Dev	Median	Dev	p-value
Sphere	15/10	3.292e-12	4.006e-12	**1.055e-13**	8.507e-14	**1.821e-05**
	30/20	1.931e-06	2.196e-06	**4.960e-08**	4.888e-08	**1.821e-05**
Rastrigin	15/10	2.984	5.65	**1.463e-06**	5.151	**0.0003**
	30/20	56.583	34.43	**0.312**	5.71	**1.821e-05**

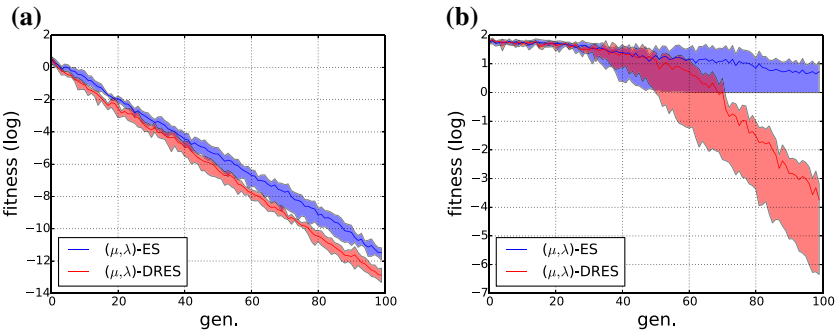
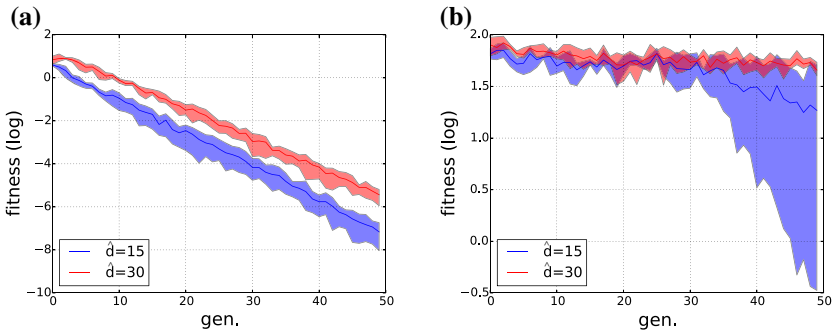


Fig. 8.2 Comparison of evolutionary runs between (15,100)-ES and (15,100)-DR-ES on **a** the Sphere function and **b** Rastrigin employing $\hat{d} = 15$ and $d = 10$

on Sphere and Rastrigin for all dimensions d and corresponding choices $\hat{d}$. The results are confirmed with the Wilcoxon signed rank-sum test. All values lie below a p-value of 0.05 and are consequently statistically significant. On the Sphere function, the DR-ES achieves slight improvements w.r.t. the median result. This observation is remarkable as the standard (μ, λ)-ES with Gaussian mutation and self-adaptive step size control is known to be a strong optimization approach on the Sphere function. Further, the (μ, λ)-ES fails on Rastrigin, where the DR-ES is significantly superior.

Figure 8.2 compares the evolutionary runs of a (15,100)-ES and a (15,100)-DR-ES on (a) the Sphere function and (b) Rastrigin with $\hat{d} = 15$ and $d = 10$. The plots show the mean, best and worst runs. All other runs lie in light blue and light red regions. The plots show that the DR-ES with PCA is superior to the standard (15,100)-ES. On the Sphere function, the DR-ES is significantly faster, on Rastrigin, it allows convergence in contrast to the stagnating standard ES.

Like observed in Table 8.1, this also holds for the Sphere function. As both ES employ comma selection, the fitness can slightly deteriorate within few optimization steps leading to a non-smooth development with little spikes.



Sphere, 15/10 vs. 30/10 Rastrigin, 15/10 vs. 30/10

Fig. 8.3 $\hat{d} = 15$ versus $\hat{d} = 30$ comparison on **a** Sphere and **b** Rastrigin with $d = 10$

The question for the influence of $\hat{d}$ arises. In the following, we compare the optimization process between large $\hat{d}$ and moderate $\hat{d}$. Figure 8.3 shows the results of 25 experimental runs on both problems with $d = 10$. The experiments show that a too large solution space deteriorates the capabilities of the DR-ES to approximate the optimum. On the Sphere, the DR-ES with $\hat{d} = 30$ achieves a log-linear development, but is significantly worse than the DR-ES in all phases of the search. On Rastrigin, the DR-ES with lower $\hat{d}$ is able to achieve better approximation capabilities than the approach with $\hat{3} = 50$. The latter does not perform fast runs, but they differ from each other resulting in the large blue area.

8.7 Conclusions

To optimize in multimodal solution spaces, the DR-ES employs a high-dimensional abstract solution space, in which the optimization processes are easier to perform. Dimensionality reduction is used to map the abstract solutions back to the original solution space dimensionality, where they are evaluated on the original fitness function. Dimensionality reduction is the mapping from a high-dimensional space to a low-dimensional space while maintaining important information like pattern neighborhoods and pattern distances. This information is used to compute low-dimensional pendants of the high-dimensional data. The search in the high-dimensional abstract space is performed as usually with continuous recombination and Gaussian mutation. Assisting evolutionary continuous search with dimensionality reduction this kind of way is a novel idea to the best of our knowledge.

For the mapping from the abstract space to the solution space, strong classic dimensionality reduction methods have be used like PCA. Our experimental results show that the DR-ES with PCA is superior to the standard ES without dimensionality reduction. PCA detects the main principal components corresponding to variances in

the data. It is based on the computation of the eigenvectors with the largest eigenvalues of the covariance matrix. Obviously, the additional features have an important impact on the search and adding additional degrees of freedoms makes the search easier. In the future, a theoretical analysis will be useful to show the impact of additional features.

References

1. Kohonen, T.: Self-Organizing Maps. Springer (1995)
2. Tenenbaum, J.B., Silva, V.D., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
3. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
4. Treiber, N.A., Kramer, O.: Evolutionary feature weighting for wind power prediction with nearest neighbor regression. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2015, pp. 332–337. Sendai, Japan, 25–28 May 2015
5. Jolliffe, I.: Principal component analysis. Springer Series in Statistics. Springer, New York (1986)
6. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philos. Mag. **2**(6), 559–572 (1901)
7. Lueks, W., Mokbel, B., Biehl, M., Hammer, B.: How to evaluate dimensionality reduction? Improving the co-ranking matrix. CoRR (2011)
8. Treiber, N.A., Späth, S., Heinermann, J., von Bremen, L., Kramer, O.: Comparison of numerical models and statistical learning for wind speed prediction. In: Proceedings of the European Symposium on Artificial Neural Networks, ESANN 2015, pp. 71–76 (2015)
9. Beyer, H., Schwefel, H.: Evolution strategies: a comprehensive introduction. Nat. Comput. **1**(1), 3–52 (2002)
10. Simões, L.F., Izzo, D., Haasdijk, E., Eiben, Á.E.: Self-adaptive genotype-phenotype maps: neural networks as a meta-representation. In: Proceedings of the Parallel Problem Solving from Nature, PPSN 2014, pp. 110–119 (2014)
11. Liepins, G.E., Vose, M.D.: Representational issues in genetic optimization. J. Exp. Theor. Artif. Intell. **2**(2), 101–115 (1990)
12. Wang, H., Jiao, L., Shang, R., He, S., Liu, F.: A memetic optimization strategy based on dimension reduction in decision space. Evol. Comput. **23**(1), 69–100 (2015)
13. Aronszajn, N.: Theory of reproducing kernels. Trans. Am. Math. Soc. **68**(3), 404 (1950)
14. Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol. Comput. **7**(1), 19–44 (1999)
15. Bremer, J., Sonnenschein, M.: Model-based integration of constrained search spaces into distributed planning of active power provision. Comput. Sci. Inf. Syst. **10**(4), 1823–1854 (2013)
16. Boschetti, F.: A local linear embedding module for evolutionary computation optimization. J. Heuristics **14**(1), 95–116 (2008)
17. Vahdat, A., Heywood, M.I., Zincir-Heywood, A.N.: Bottom-up evolutionary subspace clustering. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1–8 (2010)
18. Kramer, O.: Hybrid manifold clustering with evolutionary tuning. In: Proceedings of the 18th European Conference on Applications of Evolutionary Computation, EvoApplications 2015, pp. 481–490. Copenhagen, Denmark (2015)