Low-cost scalable discretization, prediction and feature selection for complex systems

Authors: S. Gerber[1,†], L. Pospisil[2,†], M. Navandar[1], I. Horenko[2,†,*].

Affiliations:
[1]Center of Computational Sciences, Johannes-Gutenberg-University of Mainz, PhysMat / Staudingerweg 9, 55128 Mainz, Germany

[2]Faculty of Informatics, Universita della Svizzera Italiana, Via G. Buffi 13, 6900 Lugano Switzerland.

*Correspondence to: horenkoi@usi.ch.

†These authors contributed equally to the paper.

Abstract: Finding reliable discrete approximations of complex systems is a key prerequisite when applying many of the most popular modeling tools. Common discretization approaches (for example, the very popular K-means clustering) are crucially limited in terms of quality and cost. We introduce a low-cost improved-quality Scalable Probabilistic Approximation (SPA) algorithm, allowing for simultaneous data-driven optimal discretization, feature selection and prediction. Cross-validated applications of SPA to a range of large realistic data classification and prediction problems reveal drastic cost and performance improvements. For example, SPA allows the unsupervised next-day surface temperature predictions for Europe with the mean cross-validated one-day prediction error of 0.75°C on a common PC (being around 40% better in terms of errors and five to six orders-of-magnitude cheaper than the next-day surface temperature predictions calculated on supercomputers and provided by the weather services).

One Sentence Summary: Introduced computational tool allows obtaining drastic cost and quality gains for a broad range of science applications.

Computers are finite discrete machines. Computational treatment and practical simulations of real world systems rely on the approximation of any given system's state X(t) (where t=1,...,T) in terms of a finite number K of discrete states $S=\{S_{1,\ldots,} S_K\}$[1,2]. Of particular importance are discretization methods that allow the representation of the system's states X(t) as a vector of K probabilities for the system to be in some particular state S_i at the instance t. Components of such a vector - $\Gamma^X(t)=(\Gamma_1{}^X(t), \Gamma_2{}^X(t),\ldots, \Gamma_K{}^X(t))$ - sum-up to one and are particularly important since they are necessary for Bayesian and Markovian modeling of these systems[3-5].

Bayesian and Markovian models belong to the most popular tools for mathematical modeling and computational data analysis problems in science (with over one million literature references each, according to Google Scholar). They were applied to problems ranging from social and network sciences[6] to a biomolecular dynamics and drug design[7-9], fluid mechanics[10] and climate[11]. These models dwell on the law of the total probability, saying that the exact relation between the given probabilistic representations $\Gamma^Y(t)$ and $\Gamma^X(t)$ of any two processes Y and X is given as a linear model:

$$\Gamma^Y(t) = \Lambda\Gamma^X(t), \qquad [1]$$

where Λ is a stochastic matrix of conditional probabilities between the discrete states of Y and X. This linear model is exact in a probabilistic sense – meaning that it does not impose a modeling error, even if the underlying dynamics of X and Y is arbitrarily complex and nonlinear. If Λ is known, then [1] provides the best relation between the two information sources $\Gamma^Y(t)$ and $\Gamma^X(t)$[2-4].

A particular – and very important - case of the Bayesian models [1] emerges when choosing Y(t) as X(t+1), where t is a time index. The relation matrix Λ is then a left-stochastic square matrix of transition probabilities between two discrete states, formally known as transfer operator. A Bayesian model [1] in this particular case is called a Markov model[2-4]. Besides of their direct relation to the exact law of total probability, another reason for their popularity - especially in the natural sciences - is the fact that these models automatically satisfy important physical conservation laws. They e.g. exactly conserve probability and herewith lead to stable simulations[2,7,9]. Various efficient computational methods allow to estimate conditional probability matrices Λ for real-world systems[7-15].

In practice, all these methods require a priori availability of discrete probabilistic representations. Obtaining such representations/approximations $\Gamma^X(t)$ by means of common methods from the original system's states X(t), is subject to serious quality and cost limitations. For example, applicability of grid discretization methods - covering original system's space with a regular mesh of boxes $\{S_{1,\ldots,} S_K\}$ is limited in terms of cost - since the required number of boxes K grows exponentially with the dimension n of X(t)[1].

Because of that, the most common approaches for tackling these kinds of problems are so-called meshless methods. They attempt to find a discretization by means of grouping the states X(t) into K clusters according to some similarity criteria. The computational costs for popular clustering algorithms[16] as well as for most mixture models[17] scale linearly with the dimensionality *n* of the problem and the amount of data, *T*. This cheap computation made clustering methods the most popular meshless discretization tools – even despite of the apparent quality limitations they entail. For example, K-means clustering (the most popular clustering method, with over 3 million Google Scholar citations) can only provide probabilistic approximations with binary (zero/one) $\Gamma^X(t)$ elements, excluding any other approximations and not guaranteeing optimal approximation quality. Mixture models are subject to similar quality issues when the strong assumptions that they impose (like Gaussianity in Gaussian Mixture Models) are not fulfilled.

Closely related to clustering methods are various approaches for matrix factorization – like the non-negative matrix factorization methods (NMF) that attempt to find an optimal approximation of the given (non-negative) n-times-T data matrix X with a product of the n-times-K matrix S and the K-times-T matrix Γ^X[18-24].

In situations where K is smaller than T, such non-negative reduced approximations $S\Gamma^X$ are computed by means of the fixed-point-iterations[19,21] or by alternating least-squares algorithms and projected gradient methods[22]. However, due to the computational cost issues, probabilistic factorizations (i.e., such approximations $S\Gamma^X$ that the columns of Γ are probability distributions) are either excluded explicitly[22] or they are obtained by means of the spectral decomposition of the data similarity matrices (like the X^TX matrix in the Euclidean space TxT). Such probabilistic NMF variants like the Left-Stochastic Decomposition (LSD)[24] – as well as the closely-related spectral decomposition methods[25] and the robust Perron cluster analysis[8,12] – are subject to cost limitations.

These cost limitations are induced by the fact that even the most efficient tools for eigenvalue problem computations (where all these methods rely on) scale polynomial with the similarity matrix dimension T. If the similarity matrix does not exhibit any particular structure (i.e., if it is not sparse), the overall numerical cost of the eigenvalue decomposition scales as $O(T^3)$. For example, considering twice as much data will lead to a two- to three-fold increase of cost.

Similar scalability limitations are also characteristic for the density-based clustering methods (such as the mean shifts[42], the DBSCAN[43] and the algorithms based on t-SNE[44]), having an iteration complexity in the orders between $O(Tlog(T))$ and $O(T^2)$. Practical applicability of such methods is restricted to relatively-small systems - or relies on the ad hoc data reduction steps (i.e., T cannot routinely exceed 10,000 or 20,000 when working on commodity hardware, see for example the green surface in the Fig. 1a)[9,44].

Cost and quality comparison for the probabilistic approximation methods is shown in the Fig. 1. Cost factor becomes decisive when discretizing very large systems, for example in biology and geosciences, leading to the necessity of some ad hoc data pre-processing, by means of computationally-cheap methods like K-means, Principal Component Analysis (PCA) and other pre-reduction steps [26,27].

In the following, we present a method not requiring such ad hoc reductional data pre-processing, having the same leading order computational iteration complexity O(nKT) as the cheap K-means algorithm, and allowing simultaneously finding discretizations that are optimal for models [1].

Cost, quality and parallelizability in Scalable Probabilistic Approximation (SPA)

Construction and derivation of many computational methods can be frequently approached by casting the problem into the optimization framework. For example, an approximation quality of a discretization can be expressed as a sum of all distances $dist_S(X(t), \Gamma^X(t))$ between the original state X(t) and its probabilistic discrete representation $\Gamma^X(t)$ that is obtained for $S=\{S_{1,\dots,} S_K\}$. For example, minimizing the sum of the squared Euclidean distances $\mathrm{dist}_S\left(X(t), \Gamma^X(t)\right) = \|X(t) - \sum_{k=1}^{K} \Gamma_k^X(t) S_k\|_2^2$ with respect to Γ and S for a fixed given X would allow to find the optimal probabilistic approximations $\sum_{k=1}^{K} \Gamma_k^X(t) S_k$ of the original n-dimensional data points X(t) in the Euclidean space[18-24]. S_k is an n-dimensional vector with coordinates of the discrete state k and $\Gamma_k^X(t)$ is the probability that X(t) belongs to this discrete state (referred to also as a "cluster k" or a "box k" in the following).

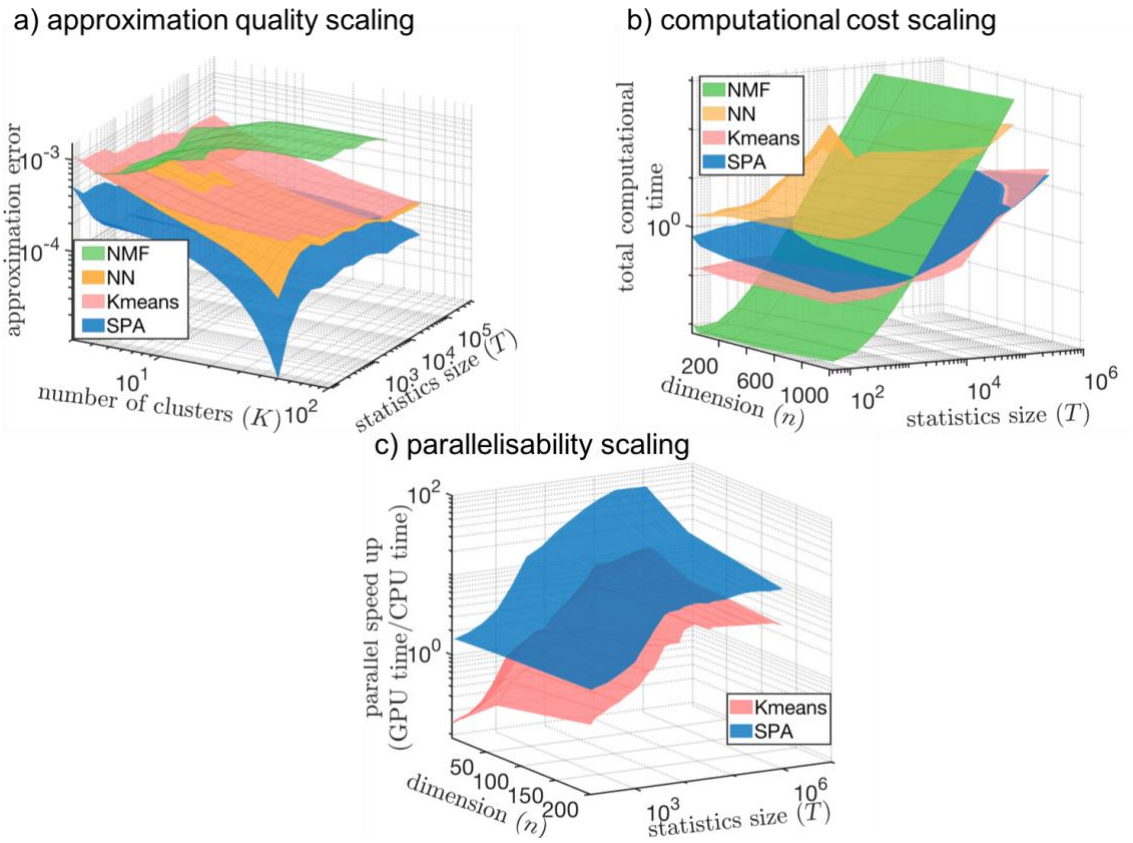


Fig. 1 | Comparing computational cost (a), discretization quality (b) and parallelizability (c) for SPA (blue surfaces) and for common discretization methods: K-means clustering[16,17] (red), Nonnegative Matrix Factorisation[19-24] (in its probabilistic variant called Left-Stochastic Decomposition[24] (LSD), green surfaces) and the Self-Organising Maps[33] (SOM, a special form of unsupervised neuronal networks used for discretization, orange surfaces). For every combination of data dimension *n* and the data statistics length *T*, methods are applied to 25 same randomly-generated data sets and the results in each of the curves represent averages over these 25 problems. Parallel speed-up in (c) is measured as the ratio of the average times *time(GPU)/time(CPU)* needed to reach the same relative tolerance threshold of 10^{-5} on a single Graphics Processing Unit (GPU, ASUS TURBO-GTX1080TI-11G, with 3584 CUDA cores) for *time (GPU)* versus a single CPU core (Intel Core i9-7900X CPU) for *time(CPU)*. Further comparisons can be found in the Fig. S2 from the Supplement. MATLAB script *Fig1_reproduce.m* reproducing these results is available in the repository *SPA* at *https://github.com/SusanneGerber*.

To the resulting expression measuring the approximation error we can add another quality measure, for example the $\Phi_S(S)$ (that measures a quality of discrete states S) and the $\Phi_\Gamma(\Gamma^X)$, measuring the quality of Γ^X. For example, persistence of the obtained discretization can be controlled by $\Phi_\Gamma(\Gamma^X) = \frac{1}{T}\sum_t ||\Gamma^X(t+1) - \Gamma^X(t)||$[30-32], whereas $\Phi_S(S)$ can be chosen as a discrepancy between the actual S and some a priori available knowledge about it[28,29].Then, the best possible probabilistic approximation can be approached by a minimization of the following quality function L with respect to the variables S and Γ^X:

$$L(S,\Gamma^X) = \sum_{t=1}^{T} \mathrm{dist}_S\big(X(t),\Gamma^X(t)\big) + \epsilon_S\Phi_S(S) + \epsilon_\Gamma\Phi_\Gamma(\Gamma^X), \qquad [2]$$

subject to the constraints that enforce that the approximation Γ^X is probabilistic

$$\sum_{k=1}^{K}\Gamma_k^X(t) = 1, \text{ and } \Gamma_k^X(t) \geq 0 \text{ for all k and t}, \qquad [3]\square$$

where $\epsilon_S, \epsilon_\Gamma \geq 0$ regulate the relative importance of the quality criteria Φ_S and Φ_Γ with respect to the approximation quality.

As proven in the Theorem 1 in the Supplement, minima of problem [2-3] can be found in linear time by means of an iterative algorithm alternating optimization for variables Γ^X (with fixed S) and for variables S (with fixed Γ^X). In the following we provide a summary of the most important properties of this algorithm. Detailed mathematical proofs of these properties can be found in the Theorems 1-3 (as well as in the Lemma 1-15 and in the Corollaries 1-11) from the Supplement.

In terms of cost, it can be shown that the computational time of the average iteration for the proposed algorithm grows linearly with the size T of the available data statistics in X – if $\Phi_\Gamma(\Gamma^X)$ is an additively separable function (meaning that it can be represented as $\Phi_\Gamma(\Gamma^X) = \sum_{t=1}^{T} \varphi_\Gamma(\Gamma^X(t))$). We will refer to the iterative methods for minimization of [2-3] - satisfying this property - as Scalable Probabilistic Approximations (SPA). Further, if the distance metrics $dist_S(X(t), \Gamma^X(t))$ is either an Euclidean distance or a Kullback-Leibler divergence, then the overall iteration cost of SPA grows as O(nKT) (where n is system's original dimension and K is the number of discrete states). In another words, computational cost scaling of SPA is the same as the cost scaling of the computationally-cheap K-means clustering[16](please see a Corollary 6 in the Supplement for a proof). Moreover, in such a case it can be shown that the amount of communication between the processors in the case of the Euclidean distance $dist_S(X(t), \Gamma^X(t))$ during one iteration in a parallel implementation of SPA will be independent of the size T of system's output – and will change proportionally to O(nK) and to the number of the used computational cores. Fig. 2 illustrates these properties and shows a principal scheme of the SPA parallelization.

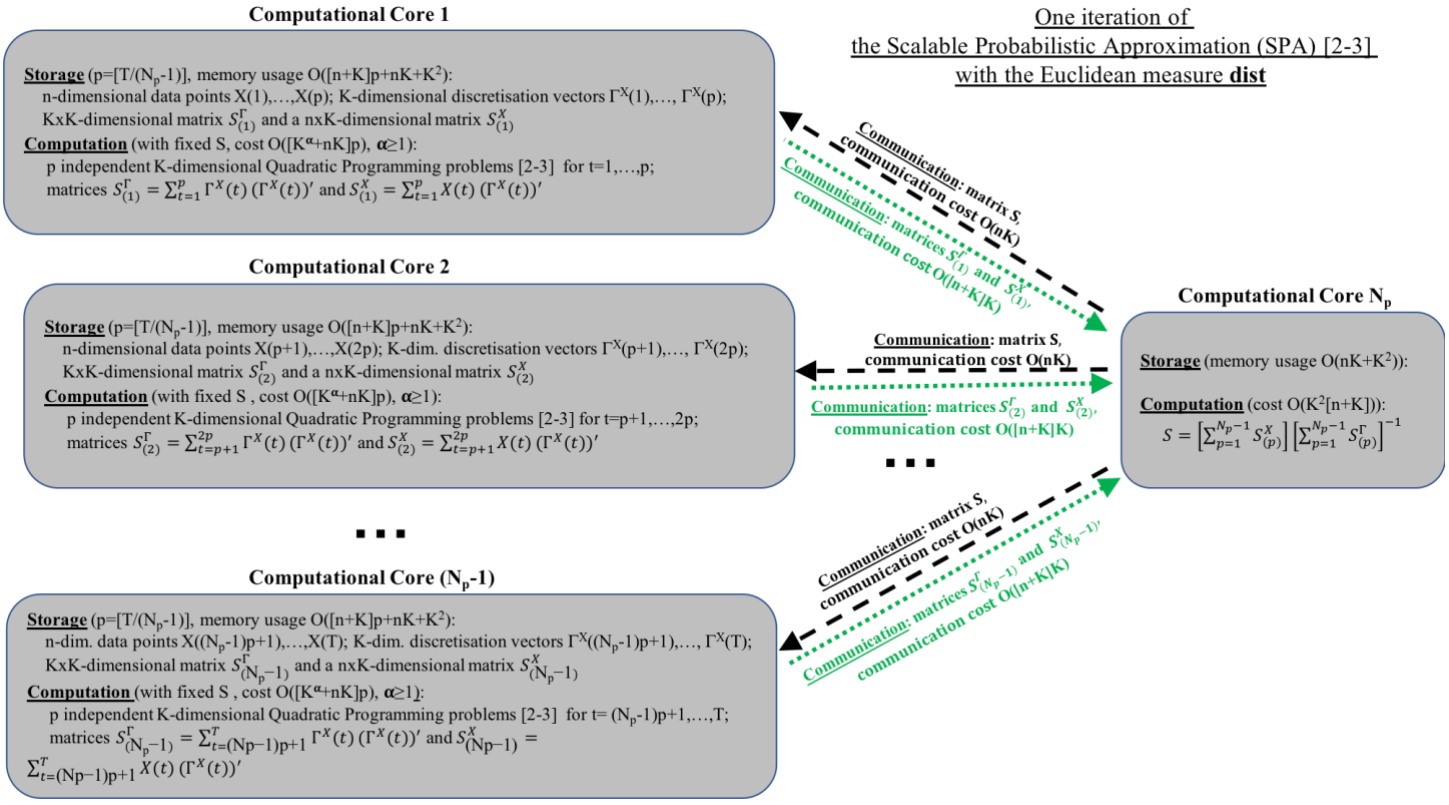


Fig. 2 | Parallelization of the Scalable Probabilistic Approximation (SPA) algorithm: communication cost of SPA for every channel is independent of the data size T and is linear with respect to the data dimension n.

In terms of quality, it is straightforward to validate that several of the common methods are guaranteed to be sub-optimal when compared to SPA – meaning that they cannot provide approximations better than SPA on the same system's data X. This can be shown rigorously for example for different forms of K-means[16] (please see Corollary 1 from the Supplement) and for the different variants of Finite Element clustering Methods on

multivariate Autoregressive Processes with external factors (FEM-VARX, in the Corollary 2 from the Supplement)[30-32].

Fig. 1 shows a comparison of SPA (blue surfaces) to the most common discretization methods, for a set of artificial benchmark problems of different dimensionality n and size T (please see the Supplement for a detailed description of the benchmarks). In comparison with K-means, these numerical experiments illustrate that SPA has the same overall cost scaling (Fig.1b), combined with the significantly better approximation quality and parallelizability scalings (Fig.1a and Fig.1c).

Computing optimal discretization for Bayesian and Markovian models

Common fitting of Bayesian or Markovian models [1] relies on the availability of discrete probabilistic representations $\Gamma^Y(t)$ and $\Gamma^X(t)$ – and requires prior separate discretization of X and Y. There is no guarantee that providing any two of such discretization $\Gamma^Y(t)$ and $\Gamma^X(t)$ as an input for any of the common computational methods[7-15] for Λ identification would result in an optimal model [1]. In another words, Bayesian and Markovian models obtained with common methods[7-15] are only optimal for a particular choice of the underlying discrete representations $\Gamma^Y(t)$ and $\Gamma^X(t)$ (that are assumed to be given and fixed for these methods) – and are not generally optimal with respect to the change of these discretization.

As proven in the Theorem 2 from the Supplement, optimal discretization of the continuous variables X and Y for the model [1] can be obtained jointly, from the family of SPA solutions by minimizing the function L [2-3] for the transformed variable $\hat{X}^\epsilon = \{Y, \epsilon X\}$. This variable is built as a concatenation (a merge) of the original variables Y and ϵX (where X is multiplied with a tunable scalar parameter $\epsilon > 0$). For any combination of parameter ϵ and the discrete dimension K in some pre-defined range, this SPA-optimization [2-3] is performed with respect to the transformed variables $S_{\epsilon,K} = \{S^Y_{\epsilon,K}\Lambda_{\epsilon,K}, \epsilon S^X_{\epsilon,K}\}$ and $\Gamma_\epsilon^{\hat{X}^\epsilon} = \{\Gamma^Y_{\epsilon,K}, \epsilon\Gamma^X_{\epsilon,K}\}$.

Then, the optimal combination of ϵ and K – and the optimal discretization for the models [1] - can be found applying standard model selection criteria[34] (for example, using information criteria or approaches like multiple cross-validation) to the obtained set of solutions $S_{\epsilon,K}$, $\Gamma_\epsilon^{\hat{X}^\epsilon}$.

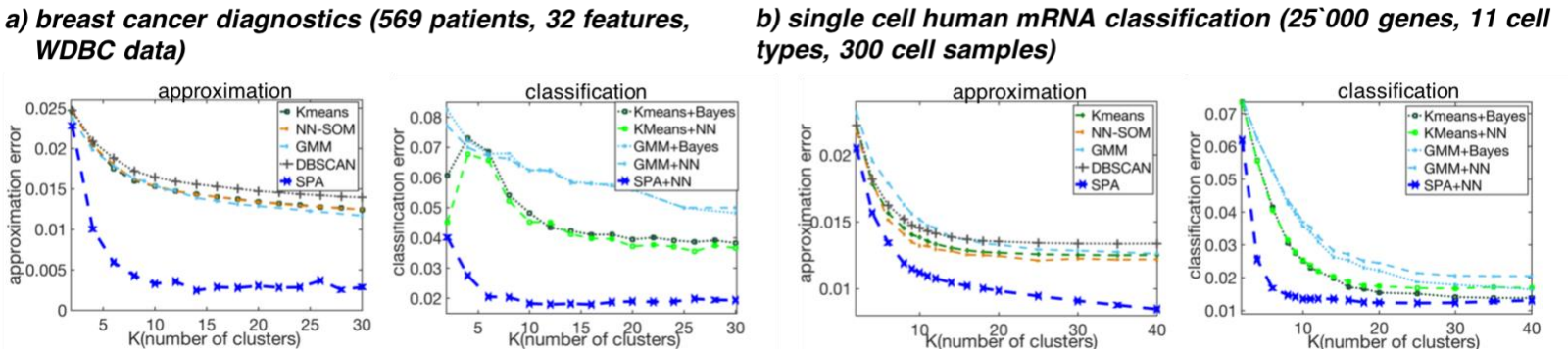


Fig. 3. | <u>Classification problems</u>: comparing approximation and classification performances of SPA (blue curves) to the common methods on biomedical applications[36,37].Common methods include K-means clustering (dotted lines), Self-Organising Maps (SOM, brown), pattern recognition Neuronal Networks (NNs, dashed), Gaussian Mixture Models (GMMs, cyan), density-based DBSCAN clustering (dotted with circles) and Bayesian models [1] (Bayes, dotted lines). Approximation error is measured as the multiply cross-validated average squared *Euclidean norm* of difference between the true and the discretized representations for validation data, classification error is measured as the multiply cross-validated average *Total Variation norm (TV)* between the true and the predicted classifications for validation data.

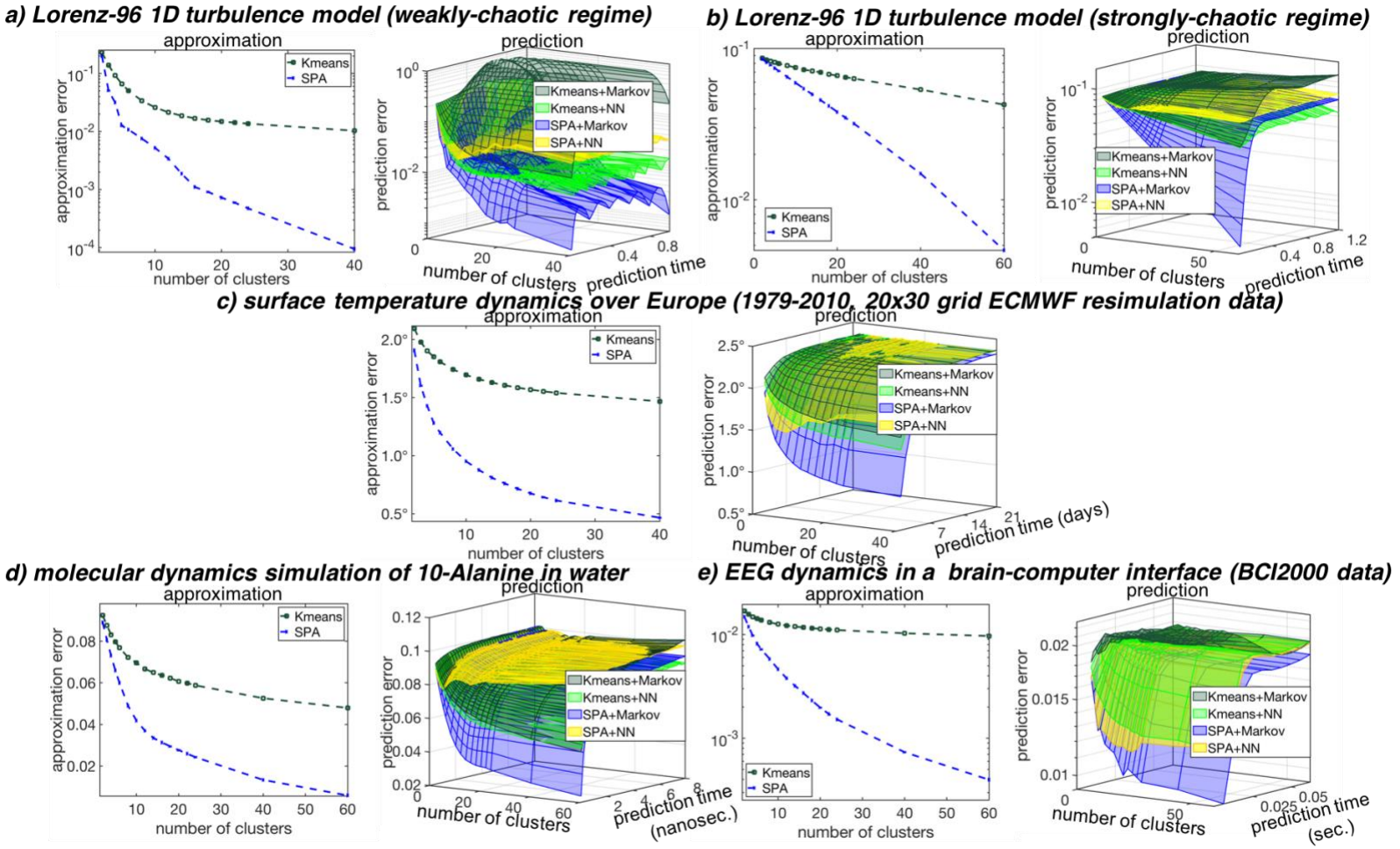


Fig. 4. | <u>Prediction problems in time series analysis</u>: comparing approximation and prediction performances of SPA (blue curves) to the common methods on open-source data sets[38,39,15,40]**:** common methods include K-means clustering (dark green) in combinations with pattern recognition Neuronal Networks (yellow and light green) and Markov models [1] (dark green). Approximation and the prediction errors are measured in the average squared *Euclidean norm* of deviations between the true and the predicted system states for the validation data not used in the model fitting.

Sensitivity analysis and feature selection with SPA

After the discretization problem is solved, an optimal discrete representation $\Gamma^X(t)$ can be computed for any continuous point X(t). Obtained vector $\Gamma^{X(t)}$ contains K probabilities $\Gamma_k^{X(t)}$ for a point X(t) to belong to each particular discrete state S_k – and allows to compute the reconstruction $X^{rec}(t)$ of the point X(t) as $X^{rec}(t)=S\Gamma^X(t)$. In this sense, procedure [2-3] can be understood as the process of finding an optimal discrete probabilistic data compression, such that the average data reconstruction error (measured as a distance between X(t) and $X^{rec}(t)$) is minimized.

In the following, we will refer to the particular dimensions of X as features – and consider a problem of identifying sets of features that are most relevant for the discretization. Importance of any feature/dimension j of X for the fixed discrete states S can be measured as an average sensitivity of the obtained continuous data reconstructions $X^{rec}(t)$ with respect to variations of the original data X(t) along this dimension j. For example, it can be measured by means of the average derivative norm $I(j) = \frac{1}{T}\sum_t \left\| \partial X^{rec}(t)/\partial X_j(t) \right\|_2^2$. For every dimension j of X(t), this quantity $I(j)$ probes an average impact of changes in the dimension j of X(t) on the resulting data reconstructions $X^{rec}(t)$. Dimensions j that have the highest impact on discretization will have the

highest values of $I(j)$, whereas the dimensions j that are irrelevant for assigning to discrete states will have $I(j)$ close to zero.

At a first glance, direct computation of the sensitivities $I(j)$ could seem to be too expensive for realistic applications with large data statistics size T and in high problem dimensions, also due to the a priori unknown smoothness of the derivates $\frac{\partial X^{rec}(t)}{\partial X_j(t)}$ in the multidimensional space of features. However, as proven in the Theorem 3 from the Supplement, in the case of discretizations obtained by solving the problem [2-3] for the Euclidean distance measure $dist$, respective derivatives $\frac{\partial X^{rec}(t)}{\partial X_j(t)}$ are always piecewise-constant functions of $X_j(t)$ if the statistics size T is sufficiently large. This nice property of derivatives allows a straightforward numerical computation of $I(j)$ for $K > 2$ – and an exact analytical computation of I for $K = 2$. It turns out that for $K = 2$ the importance of every original data dimension j can be directly measured as $(S_{2,j} - S_{1,j})^2 / \|S_2 - S_1\|_2^2$. In another words, discretization sensitivity $I(j)$ for the feature j is proportional to the squared difference between the discretization box coordinates $S_{1,j}$ and $S_{2,j}$ in this dimension j. The smaller the difference between the cluster coordinates in this dimension – the less is the impact of this particular feature j on the overall discretization.

It is straightforward to verify (please see Corollary 9 and Theorem 3 in the Supplement for a proof) that the feature sensitivity function $I = \sum_j I(j)$ has a quadratic upper bound $I \leq \sum_{j,k_1,k_2}(S_{k_1}(j) - S_{k_2}(j))^2$. Setting $\Phi_S(S)$ in [2] as $\Phi_S(S) = \sum_{j,k_1,k_2}(S_{k_1}(j) - S_{k_2}(j))^2$, for any given combination of integer K and scalar $\epsilon_S \geq 0$, minimizing [2-3] would then result in a joint simultaneous and scalable solution of the optimal discretization and feature selection problems. Overall numerical cost of this procedure will be again O(nKT). Changing ϵ_S will control the number of features: the larger is ϵ_S the fewer features (i.e., particular dimensions of the original data vector X) will be remaining relevant in the obtained discretization. Optimal value of ϵ_S can again be determined by means of standard model validation criteria[34]. In the SPA results from Fig. 3 and Fig. 4 (blue curves) we use this form of $\Phi_S(S)$ and deploy the multiple cross-validation – a standard model selection approach from machine learning – to determine the optimal ϵ_S and an optimal subset of relevant features for any given number K of discrete states (clusters).

Applications to classification and prediction problems from natural sciences

Next, we compare the discretization performance of SPA by comparing its approximation errors to the approximation errors of the common methods, including such hard-clustering methods as K-means[16], soft clustering methods based on Bayesian mixture models[17] (like Gaussian Mixture Models), density-based clustering[43] (DBSCAN) and neuronal network discretization methods (Self-Organising Maps)[33]. To compare the performances of these methods, obtained discretizations are used in parametrization of the Bayesian/Markovian models [1] – as well as in parametrization of neuronal networks[33] - on several classification and time series analysis problems from different areas. To prevent overfitting, we deploy the same multiple-cross validation protocol[34,35] adopted in machine learning for all of the tested methods. Hereby, the data is randomly subdivided into the training set (75% of the data) where the discretization and classification/prediction models are trained – and performance quality measures (approximation, classification and prediction errors) are then measured on the remaining 25% of validation data (not used in the training). For each of the methods this procedure of random data subdivision, training and validation is repeated 100 times, Fig. 3 and Fig. 4 provide the resulting average performance curves for each of the tested methods. MATLAB scripts reproducing these results are available in a repository *SPA* at *https://github.com/SusanneGerber*. Fig. 3 shows a comparison of approximation and classification performances for two problems of labelled data analysis from biomedicine and bioinformatics: (a) for a problem of breast cancer diagnostics based on X-ray image analysis[36], and (b) for a problem of single cell human mRNA classification[37]. In these problems variable

X(t) is continuous (and real-valued) set of collected features that have to be brought in relation to the discrete set of labels Y(t). In the case of the breast cancer diagnostics example[36] (a), index t denotes patients and goes from 1 to 569, X(t) contains 32 image features and Y(t) can take two values 'benign' or 'malignant'. In the case of the single cell human mRNA classification[37] (b), index t goes from 1 to 300 (there are 300 single cell probes), X(t) contains genetic expression levels for 25'000 genes and Y(t) is a label denoting one of the 11 cell types (e.g., 'blood cell', 'glia cell', etc.).

Fig. 4 summarizes results for five benchmark problems from time series analysis and prediction: for the Lorenz-96 benchmark system[38] modeling turbulent behavior in 1D, in a weakly-chaotic (a) and in the strongly-chaotic (b) regimes; (c) for the dynamics of historical surface temperatures over Europe[39], provided by the European Centre for Medium-Range Weather Forecasts (ECMWF); for the biomolecular dynamics of a 10-alanine peptide molecule in water[15]; and for the electrical activity of the brain measured in various Brain-Computer Interaction (BCI) regimes obtained with the 64 channel Electroencephalograph and provided for open access by the BCI2000-consortium[40].

As can be seen from the Fig. 3 and 4, application of popular discretization methods achieve quality plateaus for all of the considered applications. In another words, increasing the number K of discrete states (clusters) – and increasing the overall computational cost - does not improve the resulting performance accordingly. In contrast, application of methods involving SPA results in drastically improved performances – with a performance improvement factor ranging from two to four (for breast cancer diagnostics example, for single cell mRNA classification, for the temperature data over Europe and for the molecular dynamics application). For the Lorenz-96 turbulence applications[38] and for the brain activity application[40], discretization obtained by SPA are ten to hundred times better than the discretization from common methods – being at the same level of computational cost as the popular K-means clustering.

Evaluating a prediction performance of different models for a particular system, it is important to compare it with the trivial prediction strategies called *mean-value prediction* and *persistent prediction*. The *mean-value prediction* strategy predicts the next state of the system to be an expectation value over the previous already observed states – and is an optimal prediction strategy for stationary independent and identically distributed processes like the Gaussian process. The *persistent prediction* strategy is predicting the next state of the system to be the same as its current state: this strategy is particularly successful and is difficult to be beaten for the systems with more smooth observational time series, like for example for the intraday surface temperature dynamics. As it can be seen from the Fig. S3 from the Supplement, among all other considered methods (K-means, neuronal networks, SOM, mixture models) only the SPA discretization combined with the Markov models [1] allow outperforming both the *mean-value* and the *persistent predictions* for all of the considered systems.

Summary

Computational cost becomes a limiting factor when dealing with big systems. An exponential growth in the hardware performance observed over the last 60 years (the Moore's law) is expected to come to an end in the early 2020's[41]. More advanced machine learning approaches (e.g., neuronal networks) exhibit the cost scaling that grows polynomial with the dimension and with the size of the statistics – making some form of ad hoc pre-processing and pre-reduction with more simple approaches (e.g., clustering methods) unavoidable for the big data situations. However, such ad hoc pre-processing steps might impose a significant bias that is not easy to quantify. At the same time, lover cost of the method typically goes hand-in-hand with the lover quality of the obtained data representations (see Fig.1). Since the amounts of collected data in most of the natural sciences are expected to continue their exponential growth in the foreseeable future, a pressure on a computational performance (quality) and a scaling (cost) of algorithms will increase.

Instead of solving discretization, feature selection and prediction problems separately, the introduced computational procedure (a Scalable Probabilistic Approximation, or SPA) solves them simultaneously. The iteration complexity of SPA scales linearly with data size. The amount of communication between processors in the parallel implementation is independent of the data size and is linear with the data dimension – making it appropriate for big data applications. Hence, SPA did not require any form of data pre-reduction for any of the considered applications. As demonstrated in the Fig. 1, having essentially the same computational cost scaling as the very popular and computationally very cheap K-means algorithm[16-17], SPA allows achieving significantly higher approximation quality and a much higher parallel speed-up with the growing size T of the data.

Applications to large benchmark systems from natural sciences (Fig. 3 and 4) reveal that these features of SPA allow a drastic improvement of approximation and prediction qualities – combined with a massive reduction of computational cost. For example, computing the next-day surface temperature predictions for Europe (e.g., at the European Centre for Medium-Range Weather Forecasts, ECMWF) currently relies on solving equations of atmosphere motion numerically - performed on the supercomputers[39]. Discretization and prediction results for the same online daily temperature data provided in the Fig. 4c were obtained on a standard Mac PC, exhibiting a cross-validated mean error of 0.75 degree Celsius for the one-day-ahead surface air temperature predictions (approximately 40% smaller than the current next-day temperature prediction errors by weather services).

Such probability-preserving and stable predictions $\Gamma^{Y}(t)$ can be done very cheaply with the Bayesian or Markovian model [1] from the available SPA discretization [2,3] – just by computing the product of the obtained Bayesian matrix Λ with the discretization vector $\Gamma^{X}(t)$. The cost of this whole prediction operation scales linearly – resulting in orders of magnitude speed-up as compared to the predictions based on the whole system's simulations. These results indicate a potential to pave the ways to massively-parallel data-driven and error-controlled descriptive models for a robust automated classification and prediction in complex systems.

Acknowledgments: We thank Giovanni Ciccotti (La Sapienza Rome), Martin Weiser (ZIB Berlin), David L. Donoho (U Stanford), Michael Wand (JGU Mainz) and Patrick Gagliardinin (USI Lugano) for helpful comments about the manuscript.

Funding: We acknowledge the financial support from the German Research Foundation DFG ("Mercator Fellowship" of I. Horenko in the CRC 1114 "Scaling cascades in complex systems". S. Gerber acknowledges the Center of Computational Sciences in Mainz);

Author contributions: SG and IH have designed research, wrote the main manuscript and produced results in the Fig. 3 and Fig. 4; LP and IH have produced results in the Fig. 1 and proven Theorems 1-3 in the Supplement; MN has prepared the data and participated in the analysis of single cell mRNA data (Fig. 3b).

Competing interests: Authors declare no competing interests.

Data and materials availability: The description of the model systems used for Fig. 1 is provided in the Supplement. Data sets used for computations in the Fig. 3 and Fig. 4 were published[36,37,38,39,15,40] and are also available in an open access in the directory *Data* of the repository *SPA* at *https://github.com/SusanneGerber*

Model and algorithms availability: We used the standard MATLAB functions *kmeans()*, *fitgmdist()*, *patternnet()* and *som()* to compute the results of the common methods (K-means, GMM, NN and SOM) in the Figs.1+3-4. To avoid getting trapped in the local optima and to enable a unified comparison, of all methods we used 10 random initializations and selected the results with the best approximation quality measure for the training sets. In case of the pattern recognition neuronal networks (NN, evaluated in the classification and prediction performance subfigures of the Fig. 3-4), in each of the instances of the multiple cross-validation procedure we repeated the network fitting for the numbers of neurons in the hidden layer ranging between 1 and 15 and selected the results with the best classification/prediction performances for the training set. The left-stochastic discretization algorithm (LSD) from Fig. 1 was implemented by us in MATLAB according to the

literature description[24] and is provided for an open access. SPA algorithms developed and used during the current study are also available in an open access as MATLAB code at *https://github.com/SusanneGerber*

References and Notes:

1. A. Stuart, A. Humphries, *Dynamical Systems and Numerical Analysis*, vol. 8 in Cambridge Monographs on Applied Mathematics (Cambridge University Press, 1998).

2. A. J. Chorin, O. H. Hald, *Stochastic Tools in Mathematics and Science*, 3rd ed. (Springer, 2013).
3. D. B. Rubin, Bayesian Inference for Causal Effects: The Role of Randomization. *Ann. Statis.* 6(1), 34-58 (1978).

4. D. B. Rubin, *Bayesian Data Analysis*, 3rd ed. (Chapman and Hall/CRC Texts in Statistical Science, 2013).

5. Ch. Schütte, M. Sarich, *Metastability and Markov State Models in Molecular Dynamics: Modeling, Analysis, Algorithmic Approaches*, (in Courant Lecture Notes In Mathematics, vol. 24, American Mathematical Soc., 2013).

6. A. N. Langville, C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings* (Princeton University Press, Princeton, NJ, USA, 2006).
7. Ch. Schütte, W. Huisinga, P. Deuflhard, Transfer operator approach to conformational dynamics in biomolecular systems, in B. Fiedler, ed., *Ergodic theory, analysis, and efficient simulation of dynamical systems*, 191–223 (Elsevier, 2001).

8. P. Deuflhard, M. Weber, Robust Perron cluster analysis in conformation dynamics, *Linear Algebra and Its Applications*, 398, 161–184 (2005).

9. S. Gerber, S. Olsson, F. Noé, I. Horenko, A scalable approach to the computation of invariant measures for high-dimensional Markovian systems, *Nature Sci. Rep.*, 8, 1796 (2018).

10. G. Froyland, K. Padberg, Almost-invariant sets and invariant manifolds: Connecting probabilistic and geometric descriptions of coherent structures in flows. *Physica D: Nonlinear Phenomena* 238, 1507–1523 (2009).

11. A. Majda, R. Abramov, M. Grote, *Information Theory and Stochastics for Multiscale Nonlinear Systems*, (CRM monograph series, American Mathematical Soc., 2005).

12. M. Weber, S. Kube, Robust Perron cluster analysis for various applications in computational life science, *Lecture Notes in Computer Science*, 3695, 55–66 (2005).

13. T. Hofmann, ¨Probabilistic latent semantic analysis¨, in *Proceedings of the 15th Annual Conference on Uncertaintyin Artificial Intelligence* (UAI-99, San Francisco, CA: Morgan Kaufmann Publishers), 289–296 (1999).

14. S. Gerber, I. Horenko, Toward a direct and scalable identification of reduced models for categorical processes, *Proc.Natl.Acad.Sci. U.S.A.*, 114(19), 4863–4868 (2017).

15. S. Gerber, I. Horenko, On inference of causality for discrete state models in a multiscale context, *Proc. Natl. Acad. Sci. U. S. A.*, 111(41), 14651–14656 (2014).

16. J. A. Hartigan, M. A. Wong: Algorithm AS 136: A K-Means Clustering Algorithm. *J. of the Royal Stat. Soc. C* 1(28), 100–108 (1979).

17. P. D. McNicholas *Mixture Model-Based Classification,* 1st ed. (CRC Press, 2016).

18. P. Paatero, U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error. *Environmetrics*, 5, 111–126 (1994).

19. D. D. Lee, H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401, 788–791 (1999).

20. D. L. Donoho, V. Stodden, ¨Learning the parts of objects by nonnegative matrix factorization. When does non-negative matrix factorization give a correct decomposition into parts?¨ in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, B. Schölkopf, Eds., (MIT Press, Cambridge, MA, 2004) vol. 24.

21. C. H. Q. Ding, T. Li, M. I. Jordan, Convex and Semi-Nonnegative Matrix Factorizations, *J. Neur. Comput.*, 19(10), 2756–2779 (2007).

22. C.-J. Lin, Projected Gradient Methods for Nonnegative Matrix Factorization, *J. Neur. Comput.*, 19(10), 2756–2779 (2007).

23. C. Ding, T. Li, W. Peng, ¨Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence, chi-square statistic, and a hybrid method¨, in Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference, vol. 1, 342-347 (2006).

24. R. Arora, M. R. Gupta, A. Kapila, M. Fazel, Similarity-based Clustering by Left-Stochastic Matrix Factorization, *J. Machine. Learn. Res.*, 14, 1417–1452 (2013).

25. A. Ng, M. Jordan, Y. Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems, (NIPS)*, 14 (2002).

26. P. D'haeseleer, How does gene expression clustering work?, *Nature Biotech.*, 23, 1499-1501 (2005).

27. C. Cassou, Intraseasonal interaction between the Madden–Julian Oscillation and the North Atlantic Oscillation, *Nature*, 455, 523-527 (2008).

28. R. Tibshirani, Regression Shrinkage and Selection via the Lasso, *J. Royal Stat. Soc. B*, 58(1), 267-288 (1996).

29. S. Gerber, I. Horenko, Improving clustering by imposing network information, *Science Adv.*, 1(7), e1500163 (2015).

30. Ph. Metzner, L. Putzig, I. Horenko, Analysis of persistent nonstationary time series and applications, *Comm. in Appl. Math. and Comp. Sci.*, 7(2), 175–229 (2012).

31. T. J. O'Kane, R. J. Matear, M. A. Chamberlain, J. S. Risbey, B. M. Sloyan, I. Horenko Decadal variability in an OGCM Southern Ocean: Intrinsic modes, forced modes and metastable states, *Ocean Model.*, 69, 1–21 (2013).

32. N. Vercauteren, R. Klein, A Clustering Method to Characterize Intermittent Bursts of Turbulence and Interaction with Submesomotions in the Stable Boundary Layer, *J. Atmos. Sci.*, 72(4) 1504-1517 (2015).

33. T. Kohhonen, *Self-Organising Maps*, 3rd ed. (Springer Series in Information Sciences, vol. 30, 2001).

34. K. Burnham, D. Anderson, *Model Selection and Multimodel Inference:A Practical Information-Theoretic Approach*, 2nd ed. (Springer, 2002).

35. L. Trippa, L. Waldron, C. Huttenhower, G. Parmigiani, Bayesian nonparametric cross-study validation of prediction methods, *The Annals of Applied Statistics*, 9(1): 402–428, 2010.

36. W.H. Wolberg, W.N. Street, O.L. Mangasarian, Machine learning techniques to diagnose breast cancer from fine-needle aspirates. *Cancer Letters*, 77, 163-171 (1994).

37. A. A. Pollen, T. J. Nowakowski, J. Shuga, X. Wang, A. A. Leyrat, J. H. Lui, N. Li, L. Szpankowski, B. Fowler, P. Chen, N. Ramalingam, G. Sun, M. Thu, M. Norris, R. Lebofsky, D. Toppani, D. W. Kemp II, M. Wong, B. Clerkson, B. N. Jones, S. Wu, L. Knutsson, B. Alvarado, J. Wang, L. S. Weaver, A. P. May, R. C. Jones, M. A. Unger, A. R. Kriegstein, J. A. A. West, Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex, *Nat. Biotech.*, 32, 1053-1058 (2014).

38. E. Lorenz, ¨Predictability: a problem partly solved¨, in *Proceedings of the ECMWF Seminar on Predictability* (Reading, UK, ECMWF), vol. 1, 1-18 (1996).

39. D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, F. Vitart, The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Q.J.R. Meteorol. Soc*., 137, 553–597 (2011).

40. G. Schalk, J. Mellinger, *A Practical Guide to Brain–Computer Interfacing with BCI2000*, 1st ed. (Springer, 2010).

41. H. Khan, D. Hounshell, E. Fuchs, Science and research policy at the end of Moore's law, *Nature Electronics*, 1: 14–21, 2018.

42. Y. Cheng, Mean Shift, Mode Seeking and Clustering. *IEEE Transactions on Pattern Analysis and Machine Learning*, 17(8): 790–799, 1995.

43. M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press.* 226-231, 1996.

44. L.J.P. van der Maaten, Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014.

Low-cost scalable discretization, prediction and feature selection for complex systems

Susanne Gerber,[1,†] Lukáš Pospíšil,[2,†] Mohid Navandar,[1]
Illia Horenko[2,†,*]

[1]Faculty of Biology, Johannes-Gutenberg-University of Mainz,
Anselm-Franz-von-Bentzel-Weg 3, 55128 Mainz, Germany
[2]Faculty of Informatics, Universita della Svizzera Italiana,
Via G. Buffi 13, 6900 Lugano, Switzerland

[*]To whom correspondence should be addressed; E-mail: horenkoi@usi.ch.
[†]These authors contributed equally to the paper.

Description of the synthetic data problems (used in the Figure 1 of the main manuscript)

This section provides the description of the benchmark, whose results are presented in Manuscript in the Figure 1. For a given number of data points $T > 0$ and a data dimension (number of features) $n \geq 2$, we generate the random data $X = [x_1, \ldots, x_T] \in \mathbb{R}^{n,T}$ from multivariate normal distribution with different parameters based on a predefined cluster affiliation.

We choose the cluster affiliation in such a way, that the number of points affiliated to cluters

T_k is approximately the same along the clusters, i.e.,

$$\mathcal{T}_k := \left\{ (k-1) \left\lfloor \frac{T}{K} \right\rfloor + 1, \ldots, \min \left\{ k \left\lfloor \frac{T}{K} \right\rfloor, T \right\} \right\}$$

denotes the set of point indexes affiliated to k-th cluster. Please, notice that these sets are disjoint and union of them forms the set of all point indexes $\{1, \ldots, T\}$. Using this decomposition, we generate corresponding data points for every cluster $k = 1, \ldots, K$ as random realisations from the multivariate normal distributions

$$\forall t \in \mathcal{T}_k : x_t \sim \mathcal{N}(\mu_k, \Sigma_k),$$

where $\mu_k \in \mathbb{R}^n$ denotes the mean value and $\Sigma_k \in \mathbb{R}^{n,n}$ a covariance matrix.

In our benchmark, we choose $K = 4$ with parameters

$$\mu_1 := 0, \quad \Sigma_1 = \begin{bmatrix} 0.1 & 0.05 & & \\ 0.05 & 0.1 & & \\ & & \frac{0.2}{n-2} I_{n-2} & \end{bmatrix}, \quad \mu_2 := \begin{bmatrix} 0.8 \\ 1.6 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0.1 & -0.05 & & \\ -0.05 & 0.1 & & \\ & & \frac{0.2}{n-2} I_{n-2} & \end{bmatrix},$$

$$\mu_3 := \begin{bmatrix} 1.6 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \Sigma_3 = \begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & \frac{0.2}{n-2} I_{n-2} & \end{bmatrix}, \quad \mu_4 := \begin{bmatrix} 0.8 \\ 0.8 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \Sigma_3 = \Sigma_4,$$

where $I_{n-2} \in \mathbb{R}^{n-2,n-2}$ is identity matrix.

General Scalable Probabilistic Approximation (SPA) formulation

The SPA optimization problem is given by

$$[S^*, \Gamma^*] := \arg\min_{\Gamma \in \Omega_\Gamma} L(S, \Gamma), \tag{SPA}$$

where

$$L(S,\Gamma) := \sum_{t=1}^{T} \mathrm{dist}_S(X(t),\Gamma(t)) + \varepsilon_S^2\Phi_S(S) + \varepsilon_\Gamma^2\Phi_\Gamma(\Gamma), \tag{1}$$

$$\Omega_\Gamma := \{\Gamma \in \mathbb{R}^{K\times T} \mid \forall k = 1,\dots,K : \sum_{k=1}^{K} \Gamma_k(t) = 1, \Gamma_k(t) \geq 0, t = 1,\dots,T\}, \tag{2}$$

T denotes the number of data points, $X = \{X(t), t = 1,\dots,T\} \subset \mathcal{X}$ are given data from space $\mathcal{X}$ deployed with the norm $\|\cdot\|$, $K > 1$ denotes the number of discrete states (clusters), $\Gamma = \{\Gamma_k(t), k = 1,\dots,K, t = 1,\dots,T\} \subset \Omega_\Gamma \subset \mathbb{R}^{K\times T}$ are unknown cluster affiliation probability vectors, and $S : \mathbb{R}^K \to \mathcal{X}$ are unknown data representation vectors. We include the possibility of Tikhonov-based regularization of original ill-posed problem using the regularization functions Φ_S, Φ_Γ with corresponding regularization parameters $\varepsilon_S, \varepsilon_\Gamma \geq 0$.

Set a feasible initial approximation $\Gamma^0 \in \Omega_\Gamma$

while $\|L(S^k,\Gamma^k) - L(S^{k-1},\Gamma^{k-1})\| \geq \varepsilon$

solve $S^k = \arg\min_S L(S,\Gamma^{k-1})$ *(with fixed* Γ^{k-1}*)*

solve $\Gamma^k = \arg\min_{\Gamma\in\Omega_\Gamma} L(S^k,\Gamma)$ *(with fixed* S^k*)*

$k = k + 1$

endwhile

Return an approximation of the data representation vectors S^k *and an approximation of cluster affiliation probability vectors* Γ^k.

Algorithm 1: **General SPA algorithm.**

The problem (SPA) can be solved using the Algorithm 1. The idea is based on the construction of the sequence of split optimization problems. The iteration computational complexity of this algorithm is given by the complexity of the computation of inner optimization problems with fixed variables. The algorithm of this type is well-known as coordinate descent method (*10*) or alternating least-squares method (*1*). The following Lemma presents the basic

convergence properties of the algorithm.

Lemma 1. *If the solutions of inner optimization problems in Algorithm 1 exist, then algorithm generates a sequence of approximations for the optimization problem* (SPA) *with nonincreasing objective function values, i.e.,*

$$L(S^{k+1}, \Gamma^{k+1}) \leq L(S^k, \Gamma^k) \quad \textit{for } k = 1, 2, \ldots \tag{3}$$

Proof. If the solutions of inner optimization problems exist, then the solution process of inner optimization problems provides the approximation with smaller (or the same) function value with respect to non-fixed variable, i.e. (see the Definition 1 in APPENDIX),

$$\forall S: \quad L(S^k, \Gamma^{k-1}) \leq L(S, \Gamma^{k-1}), \quad \text{in the case of fixed } \Gamma^{k-1}, \tag{4}$$

$$\Gamma \in \Omega_\Gamma: \quad L(S^k, \Gamma^k) \leq L(S^k, \Gamma), \quad \text{in the case of fixed } S^k. \tag{5}$$

Choosing $S = S^{k-1}$ in (4) and $\Gamma = \Gamma^{k-1}$ in (5) we get

$$L(S^{k-1}, \Gamma^{k-1}) \geq L(S^k, \Gamma^{k-1}) \geq L(S^k, \Gamma^k).$$

□

Since the objective function (1) is generally non-convex (but bounded from bellow - each distance function is non-negative), the sequence (3) can possibly converge only to the local optimum. To deal with this non-globality, one has to run the algorithm for several random initial Γ^0 and choose the solution with the lowest function value. Such a Monte-Carlo-based approach is commonly used for solving the optimization problem with multiple local optimality points and it can be found in literature as annealing steps (*10*).

However, the convergence of the whole process still highly depends on the solvability of inner optimization problems. Following lemmas present the elementary and the most common situations when the solution exists.

Lemma 2. *If the distance function* dist_S *and the regularization function* Φ_S *in* (SPA) *are convex, bounded from below and continuously differentiable with respect to the variable* S*, then the solution with respect to* S *exists and can be found using the necessary optimality conditions for unconstrained problems.*

Proof. The Lemma is a consequence of optimization theory fundamental results, see for example (*3*). □

Lemma 3. *If the distance function* dist_S *and the regularization function* Φ_Γ *in* (SPA) *are continuous in variable* Γ*, then the solution of the problem with respect to* Γ *exists.*

Proof. Please notice that feasible set Ω_Γ is compact (i.e., closed and bounded) and convex, therefore if L is continuous, then the existence of the solution is a consequence of Weierstrass Extreme Value Theorem (*3*). □

Typically, the largest dimension parameter of the whole problem (SPA) is the number of data points T and the classification data-discretization process (SPA) does not reduce this number. It provides the data representation vectors S, whose size is determined by the size of individual data points (the dimension of vector space $\mathcal{X}$) and the number of them is equal to the number of clusters K. We can conclude, that the optimization problem with respect to S is much smaller in comparison to the optimization problem with respect to the second variable Γ. The unknown Γ consists of cluster affiliation probability vector of each individual data points, i.e., its size is determined by T and K. Fortunately, the objective function L (1) is composed as a sum of local representation errors and therefore if the regularization function $\Phi_\Gamma(\Gamma)$ is also additively separable (the case when it consists of the sum of local regularization functions for individual representations) then the whole minimization problem (SPA) is separable. The following Lemma presents the basic property of additively separable optimization problems.

Lemma 4. *If L in the optimization problem* (SPA) *is additively separable in t (except $\Phi_S(S)$), i.e., there exist functions $L_t(S, \Gamma(t)), t = 1, \ldots, T$ such that*

$$L(S, \Gamma) = \left(\sum_{t=1}^{T} L_t(S, \Gamma(t)) \right) + \Phi_S(S), \tag{6}$$

then the solution of an optimization problem (SPA) *with fixed S can be composed from solutions of individual problems*

$$\Gamma^*(t) = \arg \min_{\Gamma \in \Omega_{\Gamma_t}} L_t(S, \Gamma(t)), \tag{7}$$

where

$$\Omega_{\Gamma_t} = \{\gamma \in \mathbb{R}^K \mid \sum_{k=1}^{K} \gamma_k = 1, \gamma \geq 0\}$$

and $\Omega_{\Gamma_1} \times \cdots \times \Omega_{\Gamma_T} = \Omega_\Gamma$ is the decomposition of the feasible set of the original problem (SPA).

Proof. The definition of optimality point of (7) reads as (see the Definition 1 in APPENDIX)

$$\forall \Gamma(t) \in \Omega_{\Gamma_t} : \; L_t(S, \Gamma^*(t)) \leq L_t(S, \Gamma(t)).$$

Since this inequality can be formulated for all $t = 1, \ldots, T$, we can sum these T inequalities to obtain

$$\sum_{t=1}^{T} L_t(S, \Gamma^*(t)) \leq \sum_{t=1}^{T} L_t(S, \Gamma^*(t)).$$

If we add term $\Phi_S(S)$ (constant in Γ) to both sides of this inequality and use notation (6), we obtain

$$\forall \Gamma \in \Omega_\Gamma : L(S, \Gamma^*) \leq L(S, \Gamma),$$

which is a definition of the optimality point of optimization problem (SPA) with respect to Γ. □

The separability plays crucial role in the embarassingly parallel computations; one can solve the whole set of T optimization problems independently using modern multi-core architectures,

see Figure S2. The Γ-problem can be splitted into smaller subsets and distributed onto separated computational nodes, which is a commonly adopted approach when working on supercomputers. Each node solves the given subset of problems without any communication with the other nodes. Moreover, if the node includes multi-core processors, then (again) each core can solve independently the part of the node subproblem. This "embarrasingly-parallel" hierarchical computation of the large-scaled problem can be exploited even more when using modern GPU architectures; in this case, the relativelly small $\Gamma(t)$ problem (of size K) can be solved using just one computational thread, i.e., one computational core (please see Fig. 2c in the main manuscript).

It is necessary to mention that if the regularization function Φ_Γ is not separable in T (for example when enforcing the persistency of regime/cluster in time, see FEM-H1 and FEM-BV methods (*9*)), then the problem is not embarassingly parallel and computational nodes/cores/threads have to communicate during the solution process. However, as was demonstrated in (*12*), one can still utilize Projected Gradient methods since the projection onto separable simplexes Ω_Γ is still embarassingly parallel.

The following Theorem summarizes the general properties of the Algorithm (1).

Theorem 1 (Properties of the SPA algorithm)**.** *Let $X = \{x(t), t = 1, \dots, T\} \subset \mathcal{X}$ be given data from space $\mathcal{X}$, $K > 1$ is a given number of clusters. Let dist_S, Φ_S, Φ_Γ be such functions that $L(S, \Gamma)$ in (SPA) is convex, bounded from bellow and continuously differentiable with respect to the variable S and continuous in the variable Γ.*

Then the Algorithm 1

(a) is generating a monotonically non-increasing sequence.

Moreover, if $L(S, \Gamma)$ is additively separable problem in Γ, then the Algorithm 1

(b) scales linearly in the size T of the data statistics X,

(c) requires the amount of communication independent of the data size.

Proof. (a) is a consequence of Lemma 2, Lemma 3, and Lemma 1. To prove (b) and (c), please notice that the solution of optimization problem with respect to S is independent of the number of provided data points T. If the assumption of separability is fullfilled, then in the case of solving the problem with respect to Γ, we can using Lemma 4 reformulate the original problem as a set of T independent problems, whose dimension is (again) independent of T. □

Let us present the connection between SPA and some of the commonly used discretization (clustering) methods in following Corollaries.

Corollary 1 (Suboptimality of K-means)**.** *Measured in terms of squared Euclidean distance, discretisations providided by K-means are always suboptimal with respect to the discretisations obtained with* (SPA)*.*

Proof. Let us consider data $X \in \mathbb{R}^{n,T}$. The aim of the K-means clustering algorithm (***?***) is to optimally partition given data into K disjoint clusters based on the Euclidean distance from (unknown) optimal centroids of the clusters. The algorithm computes these cluster centroids $S_k \in \mathbb{R}^n$ and binary affiliation $\Gamma \in \{0,1\}^{K,T}$, where $\Gamma_{k,t} = 1$ if x_t belongs to k-th cluster and $\Gamma_{k,t} = 0$ otherwise. The corresponding optimization problem is formulated as

$$[S^*, \Gamma^*] := \arg\min_{\Gamma \in \Omega_\Gamma} L_{\text{kmeans}}(S, \Gamma), \quad L_{\text{kmeans}}(S, \Gamma) := \sum_{k=1}^{K} \sum_{t=1}^{T} \Gamma_{k,t} \|X(t) - S_k\|_2^2, \tag{8}$$

where $\Omega_\Gamma \subset \{0,1\}^{K,T}$ includes the condition for strict affiliation of a point into exactly one cluster, i.e.,

$$\Omega_\Gamma := \{\Gamma \in \{0,1\}^{K,T} | \forall t = 1, \ldots, T : \sum_{k=1}^{K} \Gamma_{k,t} = 1\rangle.$$

The problem (8) is solved iteratively; the feasible initial approximation of affiliations Γ is chosen randomly (the points are randomly affiliated to clusters) and afterwards, the iterative procedure

solves consecutively the problems with one fixed variable. In the case of K-means, both of the subproblems have analytical solutions

$$S_k^* = \frac{1}{\sum_{t=1}^{T} \Gamma_{k,t}} \sum_{t=1}^{T} \Gamma_{k,t} X(t), \quad \Gamma_{\bar{k},t}^* = \begin{cases} 1 & \text{if } \bar{k} = \arg\min_k \|X(t) - S_k\|, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

In fact, the scheme of the algorithm is the same as in the Algorithm 1 and one can easily check that if Γ is binary variable and we choose $\mathrm{dist}_S(X(t), \Gamma(t)) := \sum_{k=1}^{K} \|X(t) - S\Gamma_k(t)\|_2^2$ in (SPA) (in following text denoted as (SPA$_2$)) then

$$L(S, \Gamma) = \sum_{t=1}^{T} \sum_{k=1}^{K} \|X(t) - S\Gamma_k(t)\|_2^2 = \sum_{k=1}^{K} \sum_{t=1}^{T} \Gamma_{k,t} \|X(t) - S_k\|_2^2 = L_{\mathrm{kmeans}}(S, \Gamma) \tag{10}$$

and therefore K-means algorithm is equivalent to (SPA$_2$).

The variant of K-means algorithm with relaxed binary condition is well-known as soft K-means algorithm (***?***). In this case, $\Gamma_{k,t}$ represents the probability that $X(t)$ is affiliated to the k-th cluster. The feasible set Ω_Γ enforces the rows of Γ to be a corresponding discrete probability density vector, i.e., each element is continuous variable from $[0, 1]$ and because of the law of the total probability, the sum of the elements of this vector has to be equal to one. One can easily check that Ω_Γ defined by (2) represents these conditions. However in the case of continuous Γ, the equality (10) does not hold. Using the Jensen's inequality (*10*) we get

$$L(S, \Gamma) = \sum_{t=1}^{T} \sum_{k=1}^{K} \|X(t) - S\Gamma_k(t)\|_2^2 \leq \sum_{k=1}^{K} \sum_{t=1}^{T} \Gamma_{k,t} \|X(t) - S_k\|_2^2 = L_{\mathrm{kmeans}}(S, \Gamma)$$

and therefore soft K-means algorithm produces only the upper estimation of the (SPA$_2$) optimization problem. □

Corollary 2 (Suboptimality of FEM-BV and FEM-H1.)**.** *Measured in terms of a squared Euclidean distance, discretisations providided by FEM-BV and FEM-H1 methods are always suboptimal with respect to the discretisations obtained with* (SPA).

Proof. The family of FEM-BV and FEM-H1 methods consists of methods used for time series analysis (*9*), (*12*). The idea is to extend stationary models with clustering and additional time regularization for enforcing the model time persistency.

In time series modelling, we suppose that the measured data $x_1, x_2, \ldots, x_T \in \mathbb{R}^n$ are described by the parametric model ψ and include the additive noise, i.e.,

$$x_t = \psi(t, \Theta) + \varepsilon_t. \tag{11}$$

For instance one can consider autoregessive models, e.g., the Var-X model defined as

$$\psi(t, \Theta) = \mu + \sum_{i=0}^{p} A_i x_{t-i\tau} + \sum_{j=0}^{q} B_j u_{t-j\tau}, \tag{12}$$

where $\Theta = (\mu, A_0, \ldots, A_p, B_0, \ldots, B_q)$ includes all model parameters, $\tau > 0$ is a discretisation time step, $p, q \geq 0$ represent the size of memory, and u_t denote the external factors or controls. The aim of the analysis is to find parameters of the model which fit the given data x_t, u_t in an optimal way, for example, one can utilize minimum least square error to formulate optimization problem

$$\Theta^* := \arg\min_{\Theta} \sum_{t=1}^{T} \|x_t - \psi(t, \Theta)\|_2^2. \tag{13}$$

In the case of Var-X model (12) the optimization problem (13) is unconstrained quadratic programming problem and the necessary optimality conditions formulate the corresponding system of linear equations which has to be solved.

FEM-BV and FEM-H1 belong to the non-stationary models; here we suppose that the parameters of model Θ are non-stationary, i.e., they are changing (can change) in time. In general, non-stationary model without any additional assumptions, e.g., restriction of the set of permissible parameters, lead to ill-posed and biased results. In the case of FEM-BV and FEM-H1, we include the assumption of the time persistency of model parameters introducing the finite number of regimes (i.e., clusters) in which the model parameters are stationary. The switching

between those regimes is realized by a hidden regime-switching process, which describes the activity of each regime in a given time. For example, if we consider stationary Var-X model (12) on each of the K regimes, then the corresponding optimization problem is formulated as

$$[\Theta^*, \Gamma^*] := \arg \min_{\Theta, \Gamma \in \Omega_\Gamma} \sum_{t=1}^{T} \sum_{k=1}^{K} \Gamma_{k,t} \|x_t - \psi(t, \Theta_k)\|_2^2 + \varepsilon^2 \Phi_\Gamma(\Gamma), \tag{14}$$

where $\Theta = [\Theta_1, \ldots, \Theta_K]$ includes (unknown) parameters of local models on regimes and $\Gamma_{k,:}$ are model indicator functions defined in similar as in the case of K-means, i.e., $\Gamma_{k,t} = 1$ if the time series in time t is in k-th regime and and $\Gamma_{k,t} = 0$ otherwise. Regularization function $\Phi_\Gamma(\Gamma)$ with regularization parameter $\varepsilon^2 \geq 0$ enforces the time persistency of a regime-switching process. In the case of FEM-BV, we consider Bounded variation (BV) norm defined as

$$\Phi_\Gamma(\Gamma) := \sum_{k=1}^{K} \sum_{t=1}^{T-1} |\Gamma_{k,t+1} - \Gamma_{k,t}|.$$

If we consider binary Γ then this value is equal to the number of switches between regimes and the regularization by this function decreases the global number of switches in the solution. The optimization problem (14) is solved using Algorithm 1, however, in this case the Γ subproblem is not separable due to non-separable regularization term and this problem of dimension KT has to be solved using linear programming algorithm. For extended details on the method see (*9*).

It is straightforward to verify that the formulation of FEM-BV corresponds to (SPA) with distance function defined as a local Euclidean distance between given data $X(t)$ and the local value of model ψ

$$\mathrm{dist}_\Theta(X(t), \Gamma(t)) := \|X(t) - \psi(t, \Theta_\Gamma(t))\|^2, \quad \Theta_\Gamma(t) = \sum_{k=1}^{K} \Gamma_{k,t} \Theta_k. \tag{15}$$

Similarly to the soft K-means clustering case considered in the Corollary 1 above, we can relax the hard clustering property (i.e., the property that each data point is exclusively affiliated to exactly one regime) considering $\Gamma_{k,t}$ to be probability of affiliation of $X(t)$ to k-th regime.

Each $\Gamma_{:,t}$ forms the discrete probability density vector of affiliation of $X(t)$ to regimes and a corresponding feasible set is given by (2). To include the assumption of time persistency, one can adopt the H1 half-norm

$$\Phi_\Gamma(\Gamma) := \sum_{k=1}^{K}\sum_{t=1}^{T-1}(\Gamma_{k,t+1} - \Gamma_{k,t})^2$$

to get the FEM-H1 method, see (*9*). The problem is solved by an Algorithm 1, the corresponding Γ subproblem is non-separable convex quadratic programming problem of size KT, see (*12*).

Please notice that Θ depends linearly on variable Γ, the Var-X model depends linearly on parameters Θ, and the distance function dist_Θ is convex in variable ψ. Summarizing these properties we can state that distance function is convex in Γ (see (*3*) for the list of operations which preserve convexity). Using the Jensen's inequality we get

$$L(S,\Gamma) = \sum_{t=1}^{T}\sum_{k=1}^{K}\|X(t) - \psi(t,\Theta_\Gamma(t))\|_2^2 \leq \sum_{t=1}^{T}\sum_{k=1}^{K}\Gamma_{k,t}\|X(t) - \psi(t,\Theta_k)\|_2^2 = L_{\mathrm{FEM}}(S,\Gamma).$$

This inequality holds also when we add any regularization $\Phi_\Gamma(\Gamma)$ to the both sides. Hence, FEM-BV and FEM-H1 algorithms produce only the upper estimation of the (SPA) optimization problem with a corresponding choice of distance function and regularization. □

SPA in the Euclidean space

We consider the data from real n-dimensional vector space $\mathcal{X} := \mathbb{R}^n$ and Euclidean distance measure on $\mathcal{X}$ defined by

$$\mathrm{dist}_S(X(t),\Gamma(t)) := \sum_{k=1}^{K}\|X(t) - S\Gamma_k(t)\|_2^2.$$

For simplicity, we compose the vectors into matrices

$$X := [X(1),\ldots,X(T)] \in \mathbb{R}^{n,T},\ \Gamma := [\Gamma(t),\ldots,\Gamma(T)] \in \mathbb{R}^{K,T},\ S \in \mathbb{R}^{n,K}$$

and afterwards, the corresponding optimization problem (SPA) without regularization can be written in a form

$$[S^*, \Gamma^*] := \arg\min_{\Gamma \in \Omega_\Gamma} \|X - S\Gamma\|_F^2, \tag{SPA$_2$}$$

where F denotes Frobenius norm and the feasible set is defined by

$$\Omega_\Gamma := \{\Gamma \in \mathbb{R}^{K,T} \mid \forall t = 1, \ldots, T \; \forall k = 1, \ldots, K : \sum_{k=1}^{K} \Gamma_{k,t} = 1, \Gamma_{k,t} \geq 0\}. \tag{16}$$

Lemma 5. *The solutions of problem* (SPA$_2$) *are always non-unique for any* $K > 1$.

Proof. Let us consider an arbitrary solution $[S^*, \Gamma^*]$ and nonsingular matrix $R \in \mathbb{R}^{K,K}, R \neq I_{K,K}$ such that $R\Gamma \in \Omega_\Gamma$. Such a matrix always exists, e.g., we can consider a permutation matrix which permutes the rows of Γ, i.e., the indexes of clusters. Since we can write

$$L(S^*, \Gamma^*) = \|X - S^*\Gamma^*\|_F^2 = \|X - S^* \underbrace{R^{-1}R}_{=I} \Gamma^*\|_F^2 = L(S^*R^{-1}, R\Gamma^*),$$

we can state that feasible $[S^*R^{-1}, R\Gamma^*] \neq [S^*, \Gamma^*]$ has the same (minimal) function value and therefore it also solves the problem. □

Optimality conditions

We define the Lagrange function (*10*) corresponding to the optimization problem (SPA$_2$) by

$$\mathcal{L}(S, \Gamma, \lambda^E, \lambda^I) := \|X - S\Gamma\|_F^2 + \sum_{t=1}^{T} \lambda_t^E \left(\sum_{k=1}^{K} \Gamma_{k,t} - 1 \right) - \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_{k,t}^I \Gamma_{k,t}.$$

Here $\lambda^E \in \mathbb{R}^T$ are Lagrange multipliers corresponding to equality constraints defined by the feasible set (16) and $\lambda^I \in \mathbb{R}^{K,T}$ denotes the Lagrange multipliers corresponding to the non-negativity bound constraints in (16).

The full system of Karush-Kuhn-Tucker (KKT) optimality conditions for this system will

be:

$$\nabla_S \mathcal{L}(S,\Gamma,\lambda^E,\lambda^I) = \quad -2X\Gamma^T + 2S\Gamma\Gamma^T \qquad = 0, \tag{17}$$

$$\nabla_\Gamma \mathcal{L}(S,\Gamma,\lambda^E,\lambda^I) = \quad -2S^TX + 2S^TS\Gamma + (\lambda^E)^T \otimes \mathbb{1}_K - \lambda^I \quad = 0, \tag{18}$$

$$\nabla_{\lambda_E} \mathcal{L}(S,\Gamma,\lambda^E,\lambda^I) = \quad \Gamma^T\mathbb{1}_K - \mathbb{1}_T \qquad = 0, \tag{19}$$

$$\nabla_{\lambda_I} \mathcal{L}(S,\Gamma,\lambda^E,\lambda^I) = \quad -\Gamma \qquad \leq 0, \tag{20}$$

$$\lambda^I \qquad \geq 0, \tag{21}$$

$$\forall k,t: \quad \lambda^I_{k,t}\Gamma_{k,t} \qquad = 0, \tag{22}$$

where $\mathbb{1}_K \in \mathbb{R}^K, \mathbb{1}_T \in \mathbb{R}^T$ denotes the vectors of ones. Equations (17) and (18) are first-order optimality conditions, equation (19) and inequality (20) are constraints given by the definition of the feasible set (16), inequality (21) preserves the non-negativity of inequality Lagrange multipliers, and equations (22) represent the so-called complementarity conditions for inequality constraints.

The solution of S subproblem

Lemma 6 (The solution of S-problem)**.** *Let $\Gamma \in \Omega_\Gamma$ in problem (SPA$_2$) be fixed. Then the system of all solutions of optimization problem (SPA$_2$) with respect to S is given by*

$$S^* = X\Gamma^T\left(\Gamma\Gamma^T\right)^+ + \alpha^T R^T, \quad \textit{with parameter } \alpha \in \mathbb{R}^{r,n}, \tag{23}$$

where $\left(\Gamma\Gamma^T\right)^+ \in \mathbb{R}^{K,K}$ denotes a pseudoinverse[1] of the matrix $\Gamma\Gamma^T$, $R \in \mathbb{R}^{K,r}$ is a matrix whose columns form the basis of the null space of Γ^T, i.e.

$$\operatorname{Im} R = \operatorname{Ker} \Gamma^T, \tag{24}$$

and $r = \dim \operatorname{Ker} \Gamma^T$ denotes the nullity of matrix Γ^T.

[1] i.e. the matrix such that $AA^+A = A$, $A^+AA^+ = A^+$, $(AA^+)^T = AA^+$, and $(A^+A)^T = A^+A$

Proof. Please notice that the objective function of (SPA_2) in terms of variable S is continuously differentiable convex matrix quadratic function. The necessary optimality condition of given unconstrained optimization problem is given by (17). This system of linear equations with multiple right-hand side vectors with symmetric positive semi-definite system matrix always has a solution. If the system matrix is non-singular, then the unique solution is given by

$$S^* = X\Gamma^T(\Gamma\Gamma^T)^{-1}.$$

However, the non-singularity of system matrix $\Gamma\Gamma^T \in \mathbb{R}^{K,K}$ (and consequently, the existence of inverse matrix) is not guaranteed[2], the system of all solutions is given by (23) where all solutions differ by the vector from $\operatorname{Ker}\Gamma\Gamma^T$, see (*8*) or (*7*). □

Next we deal with the eventual ill-posedness of the optimization problem (SPA_2) in variable S, or equivalently, with the ill-posedness of the system of linear equations (17). Deploying Tykhonov-regularization, we reformulate the original (SPA) problem choosing the regularization function

$$\Phi_S(S) := \frac{1}{nK(K-1)} \sum_{i=1}^{n} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} (S_{i,k_1} - S_{i,k_2})^2 \tag{25}$$

and consider regularization parameter $\varepsilon_S^2 > 0$. Please notice that the solution of the optimization problem in term of variable S is independent on the choice of regularization function Φ_Γ. The following Lemma proves that (25) guarantees the unique solvability of S-problem.

Lemma 7. *The computational complexity of solving S subproblem in (SPA_2) is $\mathcal{O}(K^3 + KnT)$, with the memory complexity of $\mathcal{O}(K^2 + nK)$.*

Proof. The first step in solving the S subproblem is the assembly of the matrix $\Gamma\Gamma^T$ and of the matrix of the right-hand side vectors $X\Gamma^T$ in an equation (17). Let us remind that the complexity

[2]Since $\operatorname{Ker}\Gamma\Gamma^T = \operatorname{Ker}\Gamma^T$ (see (*8*)) we can see that if and only if Γ has linearly independent rows, then matrix $\Gamma\Gamma^T$ is non-singular (invertible).

of computing matrix-matrix multiplication of general (non-sparse) matrices $A \in \mathbb{R}^{n,m}$ and $B \in \mathbb{R}^{m,p}$ is $\mathcal{O}(nmp)$, therefore in our case, the overall complexity of assembling the problem is $\mathcal{O}(TK^2)+\mathcal{O}(nTK)$. The memory required to store these two new matrices is $\mathcal{O}(K^2)+\mathcal{O}(nK)$.

In general, the direct methods for solving a system of linear equation $Ax = b, A \in \mathbb{R}^{m,m}$ have the complexity of order $\mathcal{O}(m^3)$. Iterative methods, like Krylov subspace algorithms, are based on the iterations where the computational complexity scaling in the leading order is dominated by the multiplication with a system matrix A, which is of order $\mathcal{O}(m^2)$. Number of iterations needed for the convergence, when using a suitable preconditioner, is usually much less than $\mathcal{O}(n)$. Therefore, the overall work for solving the system of linear equations is less than $\mathcal{O}(m^3)$. In general, numerical linear algebra algorithms for this purpose are using the auxiliary vectors of dimension $\mathbb{R}^m$, whose number is independent on the dimension of the problem. Therefore, the amount of additional memory used for solving the system of linear equations is of the order $\mathcal{O}(m)$.

Applying these general results to S subproblem which consists of T linear systems of dimension K, we obtain the total computational complexity $\mathcal{O}(TK^3)$ and a memory complexity $\mathcal{O}(TK)$. Since the system matrix is the same for all subsystems, therefore one can compute pseudoinverse and use (23) directly, which will lead to the total computational complexity of $\mathcal{O}(n^3) + \mathcal{O}(K^2T)$. In practial applications the computation of pseudoinverse is typically much slower than solving the system of linear equations.

□

Corollary 3. *In the case of K-means algorithm, evaluation of an analytical solution* S^* (9) *consists of computing two sums with the computational complexity* $\mathcal{O}((n + K)T)$. *To compute the sums, one has to use an additional auxiliary vector of dimension* $\mathcal{O}(K)$.

Lemma 8 (S-problem with regularization)**.** *Let* $\Gamma \in \Omega_\Gamma$ *in a problem* (SPA_2) *with an additional regularization function* (25) *be fixed. Then, for any* $\varepsilon_S^2 > 0$ *the problem with respect to* S *has a*

unique solution given by

$$S^* = X\Gamma^T H_\epsilon^{-1}, H_\varepsilon := \Gamma\Gamma^T + \frac{2\varepsilon^2}{nK(K-1)}(KI_{K,K} - \mathbb{1}_{K,K}), \tag{26}$$

where $I_{K,K} \in \mathbb{R}^{K,K}$ *is an identity matrix and* $\mathbb{1}_{K,K} \in \mathbb{R}^{K,K}$ *is a matrix full of ones. Moreover the spectrum of regularized Hessian matrix* H_ε *can be estimated by*

$$\begin{array}{rcl} \lambda_{\min}(H_\varepsilon) & \geq & \min\left\{\frac{T}{K}, \frac{2\varepsilon^2}{n(K-1)}\right\}, \\ \lambda_{\max}(H_\varepsilon) & \leq & \|\Gamma\Gamma^T\|_2 + \frac{2\varepsilon^2}{n(K-1)}. \end{array} \tag{27}$$

Proof. The gradient of the original objective function L in (SPA_2) without regularization is given by the left-hand side of (17). Let us focus on the gradient of regularization function whose components are given by (for every $i \in \{1, \ldots, n\}, k \in \{1, \ldots, K\}$)

$$\begin{array}{rcl} [\nabla\Phi_S(S)]_{i,k} & = & \frac{1}{nK(K-1)}\left(\sum\limits_{k_2=1}^{K} 2(S_{i,k} - S_{i,k_2}) - \sum\limits_{k_1=1}^{K} 2(S_{i,k_1} - S_{i,k})\right) \\ & = & \frac{2}{nK(K-1)}\left(2KS_{i,k} - 2\sum\limits_{k_1=1}^{K} S_{i,k_1}\right) = \frac{4}{nK(K-1)}\left(KS_{i,k} - S_{i,:}\mathbb{1}_K\right) \end{array}$$

where $\mathbb{1}_K \in \mathbb{R}^K$ is a column vector of ones. It is easy to see that the whole gradient can be written as

$$\nabla\Phi_S(S) = \frac{4}{nK(K-1)}\left(KS - S\mathbb{1}_{K,K}\right)$$

and therefore the necessary optimality condition of the regularized problem is given by the solution of a regularized linear system of equations

$$-2X\Gamma^T + 2S\left(\Gamma\Gamma^T + \frac{2\varepsilon_S^2}{nK(K-1)}(KI_{K,K} - \mathbb{1}_{K,K})\right) = 0. \tag{28}$$

It remains to show that the system matrix is non-singular for any $\varepsilon_S^2 > 0$ and therefore we will be able to multiply the whole equation with the matrix inverse to obtain a unique solution.

Please notice that the matrix $G_K := KI_{K,K} - \mathbb{1}_{K,K}$ is a Laplacian matrix of a complete graph on K nodes, therefore it is symmetric positive semidefinite and $\operatorname{Ker} G_K = \operatorname{span}\{\mathbb{1}_K\}$, see (*5*).

For the simplicity, let us denote $\hat{\varepsilon} := \frac{2\varepsilon_S^2}{nK(K-1)} > 0$. For any non-zero $y \in \mathbb{R}^K$ we can differentiate two cases

- if $y \notin \operatorname{Ker} G_K$ then $y^T G_K y = K y^T y$ (the spectrum of complete graph Laplace matrix is composed from one zero eigenvalue and eigenvalues of value K with multiplicity $K-1$, see (5)) and

$$y^T \left(\Gamma\Gamma^T + \hat{\varepsilon} G_K\right) y = \underbrace{y^T \Gamma\Gamma^T y}_{\geq 0} + \hat{\varepsilon} \underbrace{y^T G_K y}_{= K y^T y} \geq \hat{\varepsilon} K y^T y > 0. \tag{29}$$

- if $y \in \operatorname{Ker} G_K = \operatorname{span}\{\mathbb{1}_K\}$ then there exists a non-zero $\alpha \in \mathbb{R}$ such that non-zero y can be written as $y = \alpha \mathbb{1}_K$. Using the equality constraints of the feasible set Ω_Γ (16) written in a form $\Gamma^T \mathbb{1}_K = \mathbb{1}_T$ we can state that

$$y^T \Gamma\Gamma^T y = \alpha^2 \mathbb{1}_K^T \Gamma\Gamma^T \mathbb{1}_K = \alpha^2 \mathbb{1}_T^T \mathbb{1}_T = \alpha^2 T = \frac{T}{K} \alpha^2 \mathbb{1}_K^T \mathbb{1}_K = \frac{T}{K} y^T y > 0$$

and consequently

$$y^T \left(\Gamma\Gamma^T + \hat{\varepsilon} G_K\right) y = \underbrace{y^T \Gamma\Gamma^T y}_{= \frac{T}{K} y^T y} + \underbrace{\hat{\varepsilon} y^T G_K y}_{=0} = \frac{T}{K} y^T y > 0. \tag{30}$$

This proves that $y^T(\Gamma\Gamma^T + \hat{\varepsilon} G_K)y > 0$ for any $y \neq 0$, i.e., that the system matrix in (28) is symmetric positive definite and therefore there exists a unique solution of this system given by (26). This also proves that the original objective function of a problem (SPA_2) with regularization (25) with respect to S is for any fixed $\varepsilon_S^2 > 0$ strictly convex and the optimization problem with bounded closed convex feasible set (16) has a unique minimizer. Since for any symmetric matrix and any non-zero y it holds $y^T A y \geq \lambda_{\min}(A) y^T y$, we can combine (29) and (30) to prove the lower estimation in (27). To prove upper estimation, one can use the property of norm and eigenvalues of complete graph Laplace matrix

$$\|H_\varepsilon\|_2 = \|\Gamma\Gamma^T + \hat{\varepsilon}(K I_{K,K} - \mathbb{1}_{K,K})\|_2 \leq \|\Gamma\Gamma^T\|_2 + \frac{2\varepsilon^2}{n(K-1)}.$$

□

Lemma 9 (Uniqueness of a reconstruction with the fixed Γ)**.** *Let* $[S^{1*}, \Gamma^{1*}]$ *and* $[S^{2*}, \Gamma^{2*}]$ *be two solutions of* (SPA$_2$) *for given data* X*. Let us denote the appropriate reconstructions by* $X^{\text{rec1}} := S^{1*}\Gamma^{1*}$ *and* $X^{\text{rec2}} := S^{2*}\Gamma^{2*}$*. If* $\Gamma^{1*} = \Gamma^{2*}$ *then* $X^{\text{rec1}} = X^{\text{rec2}}$*.*

Proof. From the optimality conditions, S^{1*} and S^{2*} solves (SPA$_2$) with fixed $\Gamma := \Gamma^{1*} = \Gamma^{2*}$. All solutions of corresponding QP differ by a vector from kernel of Hessian matrix (see (*7*), (*12*), and (23)) and using Lemma 21 we get

$$X^{\text{rec1}} - X^{\text{rec2}} = \underbrace{(S^{1*} - S^{2*})}_{\in \text{Ker}\,\Gamma\Gamma^T = \text{Ker}\,\Gamma^T} \Gamma = 0.$$

□

Lemma 10 (Derivative of solution with a fixed Γ)**.** *Let* $\Gamma \in \Omega_\Gamma$ *in problem* (SPA$_2$) *with additional regularization function* (25) *be fixed and let* $S^*(X)$ *be solution* (26) *for any* X*. Then for any* $j = 1, \ldots, n$ *and* $t = 1, \ldots, T$

$$\left\| \frac{\partial S^*(X)}{\partial X_{j,t}} \right\|_2 \leq \frac{1}{\lambda_{\min}(H_\varepsilon)} \leq \frac{1}{\min\left\{ \frac{T}{K}, \frac{2\varepsilon^2}{n(K-1)} \right\}}, \tag{31}$$

where $\lambda_{\min}(H_\varepsilon)$ *is the smallest eigenvalue of regularized Hessian matrix* H_ε *given by* (26) *and further estimated using* (27)*.*

Proof. We use the derivative definition

$$\frac{\partial S^*(X)}{\partial X_{j,t}} = \lim_{\delta \to 0} \frac{S^*(X + \delta e_{j,t}) - S^*(X)}{\delta \|e_{j,t}\|_2},$$

where $e_{j,t} \in \mathbb{R}^{n,T}$ is a standard basis vector with elements defined by

$$i = 1, \ldots, n, \tau = 1, \ldots, T: \quad [e_{j,t}]_{i,\tau} := \begin{cases} 1, & \text{if } i = j \text{ and } \tau = t, \\ 0, & \text{elsewhere.} \end{cases}$$

Using the solution (26), the norm can be estimated by

$$\left\| \frac{\partial S^*(X)}{\partial X_{j,t}} \right\|_2 = \lim_{\delta \to 0} \frac{\|S^*(X + \delta e_{j,t}) - S^*(X)\|_2}{\delta \|e_{j,t}\|_2} = \lim_{\delta \to 0} \frac{\delta \left\| e_{j,t}\Gamma^T H_\varepsilon^{-1} \right\|_2}{\delta \|e_{j,t}\|_2} = \left\| e_j \gamma_t^T H_\varepsilon^{-1} \right\|_2,$$

where $e_j \in \mathbb{R}^n$ is vector of standard basis and $\gamma_t := \Gamma_{:,t}$. Using the property of the norm, we can further estimate

$$\left\|e_j \gamma_t^T H_\varepsilon^{-1}\right\|_2 \leq \|e_j\|_2 \|\gamma_t\|_2 \left\|H_\varepsilon^{-1}\right\|_2 \leq \left\|H_\varepsilon^{-1}\right\|_2 = \frac{1}{\lambda_{\min}(H_\varepsilon)}.$$

□

Corollary 4. *In the case of K-means, the indicator functions* Γ *are binary and*

$$H_0 = \Gamma\Gamma^T = \begin{bmatrix} N_1 & & \\ & \ddots & \\ & & N_K \end{bmatrix} \in \mathbb{R}^{K,K}, \quad N_k := \sum_{t=1}^{T} \Gamma_{k,t},$$

where $N_k \geq 0$ *denotes the number of points affiliated to* k*-th cluster. The eigenvalues of diagonal matrix* H_0 *are equal to the values on the diagonal, therefore upper estimation* (31) *depends only on the inverse value of the smallest cluster size; it is independent on both of the data size and number of clusters.*

The solution of Γ subproblem

In this Section, we suppose that in the optimization problem (SPA_2) the variable S is fixed and it remains to solve the problem in a variable Γ only (the second optimization problem of Algorithm 1). In this case, the objective function is additively separable and it can be written in the form of separable Quadratic Programming (QP) problems with linear equality and bound constraints.

Lemma 11. *The solution of* (SPA_2) *with fixed* S *is equivalent to the solution of* T *independent QP problems*

$$\gamma_t^* := \arg\min_{\gamma \in \Omega_\gamma} \frac{1}{2}\gamma^T A\gamma - b_t^T\gamma, \quad \Omega_\gamma := \{\gamma \in \mathbb{R}^K \mid B\gamma = c, \gamma \geq 0\}, \tag{32}$$

where

$$A := 2S^TS,\ b_t := S^Tx_t,\ B := \mathbb{1}_K^T,\ c := 1,$$
$$X = [x_1, \ldots, x_T] \in \mathbb{R}^{n,T},$$

and the original solution of (SPA_2) *can be composed as*

$$\Gamma^* := [\gamma_1^*, \dots, \gamma_T^*] \in \mathbb{R}^{K,T}.$$

Proof. From the definition of Frobenius norm and matrix-matrix multiplication we have

$$\begin{aligned}\|X - S\Gamma\|_F^2 = \sum_{t=1}^{T} \|x_t - S\gamma_t\|_2^2 = \sum_{t=1}^{T} \left(x_t^T x_t - 2x_t^T S\gamma_t + \gamma_t^T S^T S\gamma_t\right) \\ \propto \sum_{t=1}^{T} \tfrac{1}{2}\gamma_t^T (2S^T S)\gamma_t - (S^T x_t)^T \gamma_t.\end{aligned}$$

Moreover, it is easy to check that the composition of Ω_γ for all $\gamma_t, t = 1, \dots, T$ forms the original feasible set Ω_Γ. Then using Lemma 4 the problem can be rewritten as the solution of the separated subproblems. □

From the computational point of view, the Γ-problem is more challenging since one has to deal with optimization problems on the fasible set described by the combination of linear equality constraints and bound constraints. In the case of QP (32), the subproblems can be solved by the Interior-Point methods or by the Augumented Lagrangian methods combined with Active-set approach (*10*), (*7*). In our implementation we use the fact that the feasible set Ω_γ is the simplex of size K. Since the objective function is continuously differentiable, then one can use Projected Gradient Descent methods, for example Spectral projected gradient method for QP (*2*), (*12*).

Lemma 12. *The computational complexity of decreasing the objective function in* Γ *for a fixed* A *in* (SPA_2) *is* $\mathcal{O}(nK^2 + nKT + TK^2)$*, with a memory complexity of* $\mathcal{O}(K^2 + KT)$.

Proof. The complexity of assembling this QP problem is given by the complexity of a matrix-matrix multiplications $S^T S$ and $S^T X$, which is $\mathcal{O}(nK^2 + nKT)$. These objects require a memory of the order $\mathcal{O}(K^2 + KT)$.

The number of iterations required for solving this QP problem on convex sets depends on the spectral properties of its Hessian matrix (*7*). Let us focus on one iteration, which will decrease

the value of an objective function (32). Such a decrease can be obtained using a projected gradient descend step

$$\gamma^{k+1} = P_{\Omega_\gamma}(\gamma^k - \bar{\alpha}\nabla f(\gamma^k)), \tag{33}$$

with a step-length $\bar{\alpha} \in (0, \|A\|^{-1})$. Decrease of the function value for a convex QP on a general closed convex set has been proven in (*6*), (*11*).

The computational complexity of computing the gradient in (33) is $\mathcal{O}(K^2)$ because of the Hessian matrix multiplication. Computational iteration complexity of the projection onto a simplex is of order $\mathcal{O}(K^2)$ (*4*), (*12*). Since the step has to be performed for all γ_t, the overal complexity is $\mathcal{O}(TK^2)$. The step for each γ_t requires auxiliary vectors of additional memory $\mathcal{O}(K)$, therefore a computation of the whole Γ takes additional $\mathcal{O}(KT)$ of memory. □

Corollary 5. *In the case of K-means algorithm, the evaluation of analytical solution* Γ^* (9) *consists of evaluation of local error and finding the maxima for all data points. The computational complexity is* $\mathcal{O}(nKT)$ *and the size of auxiliary vectors is* $\mathcal{O}(KT)$.

Lemma 13. *The computational complexity of one iteration of* (SPA$_2$) *is* $\mathcal{O}(nKT+(n+T)K^2+K^3)$, *with a memory complexity of* $\mathcal{O}(K^2+(n+T)K)$.

Proof. The Lemma is a direct combination of Lemma 7 and Lemma 12. □

Corollary 6. *The complexity of one iteration of K-means algorithm can be obtained combining Corollary 3 and Corollary 5. The computational complexity is* $\mathcal{O}(nKT + (n + K)T)$ *and the memory complexity* $\mathcal{O}(KT+K+n)$. *In practical big data applications the dimension* n *and the statistics size* T *are much larger then the discretisation dimension* K. *It means that in such situations both K-means and SPA will have the same leading order of the computational iteration complexity* $\mathcal{O}(nkT)$ *and the same leading order of the required memory in* T, *being* $\mathcal{O}(KT)$. *In contrast, spectral clustering methods (like LSD, PCCA+) and density-based clustering meth-*

ods (like DBSCAN and "mean shift") will have the leading order in both the computational complexity and in the required memory scaling ranging between $\mathcal{O}(T\log(T))$ *and* $\mathcal{O}(T^2)$.

Lemma 14. *Let* $S \in \mathbb{R}^{n,K}$ *be fixed. Function* $\gamma^* : \mathbb{R}^n \to \Omega_\gamma$ *defined as*

$$\gamma^*(x) := \arg\min_{\gamma\in\Omega_\gamma} \|x - S\gamma\|_2^2$$

is a continuous piecewise linear function.

Proof. Let us consider arbitrary $x_1, x_2 \in \mathbb{R}^n$ and corresponding $\gamma_1 := \gamma^*(x_1), \gamma_2 := \gamma^*(x_2)$. Since both of these values solve the optimization problem, there exist appropriate Lagrange multipliers $\lambda_1^I, \lambda_1^E, \lambda_2^I, \lambda_2^E$ such that the KKT optimality conditions (18), (19), (20), (21), (22) are satisfied in the form

$$-2S^T x_t + 2S^T S\gamma_t + \lambda_t^E \mathbb{1}_K - \lambda_t^I = 0, \tag{34}$$

$$\gamma_t^T \mathbb{1}_K = 1, \tag{35}$$

$$\gamma_t, \lambda_t^I \geq 0, \tag{36}$$

$$\forall k : \{\lambda_t^I\}_k\{\gamma_t\}_k = 0 \tag{37}$$

for both of the given $t \in \{1, 2\}$. Let us consider parameter $\alpha \in [0, 1]$, build a convex combination of equations (34) and get

$$-2S^T x_\alpha + 2S^T S\gamma_\alpha + \lambda_\alpha^E \mathbb{1}_K - \lambda_\alpha^I = 0, \tag{38}$$

where we denoted

$$\begin{aligned} x_\alpha &:= (1-\alpha)x_1 + \alpha x_2, \\ \gamma_\alpha &:= (1-\alpha)\gamma_1 + \alpha\gamma_2, \\ \lambda_\alpha^E &:= (1-\alpha)\lambda_1^E + \alpha\lambda_2^E, \\ \lambda_\alpha^I &:= (1-\alpha)\lambda_1^I + \alpha\lambda_2^I. \end{aligned} \tag{39}$$

It is easy to see that (38) can be considered as the first KKT optimality condition for any x_α which lies on the line connecting x_1, x_2. In this case, the solution $\gamma_\alpha = \gamma^*(x_\alpha)$ of the corresponding optimization problem can be built as a linear combination of γ_1, γ_2 with the same coeficient. The conditions (35) and (36) for γ_α are also satisified since the feasible set Ω_γ is convex (and every convex combination of points inside the convex set is also in this set) and/or one can directly check that for any $\alpha \in [0, 1]$

$$\gamma_\alpha^T \mathbb{1}_K = (1-\alpha)\underbrace{\gamma_1^T \mathbb{1}_K}_{=1} + \alpha \underbrace{\gamma_2^T \mathbb{1}_K}_{=1} = 1,$$
$$\gamma_\alpha = \underbrace{(1-\alpha)\gamma_1}_{\geq 0} + \underbrace{\alpha\gamma_2}_{\geq 0} \geq 0,$$
$$\lambda_\alpha^I = \underbrace{(1-\alpha)\lambda_1^I}_{\geq 0} + \underbrace{\alpha\lambda_2^I}_{\geq 0} \geq 0.$$

The reason why the function γ^* is not linear for general x_1, x_2 is the complementarity condition. If we substitute (39) into (37) for α, we obtain

$$\forall k : \{\lambda_\alpha^I\}_k \{\gamma_\alpha\}_k = \alpha(1-\alpha)\left(\{\lambda_1^I\}_k\{\gamma_2\}_k + \{\lambda_2^I\}_k\{\gamma_1\}_k\right) = 0.$$

Since (36) and (37) such a condition is satisfied for all $\alpha \in [0, 1]$ if and only if for all k

$$\{\lambda_1^I\}_k = \{\lambda_2^I\}_k = 0 \quad \text{and/or} \quad \{\gamma_1\}_k = \{\gamma_2\}_k = 0.$$

The line connecting x_1, x_2 can be splitted into the segments which satisfied these conditions and therefore the function γ^* is piecewise linear. □

Corollary 7. *Let S be fixed and let us define a function*

$$X^{\text{rec}}(X) := S\Gamma^*(X), \text{ where } \Gamma^*(X) := \arg\min_{\Gamma \in \Omega_\Gamma} \|X - S\Gamma\|_F.$$

It is easy to see that this function linearly depends on $\Gamma^*(X)$ and since this separable function is composed from linear functions (see Lemma 14) the derivative

$$\frac{\partial X^{\text{rec}}}{\partial X}$$

is a piecewise constant function.

Lemma 15. *Let $K = 2$, $S \in \mathbb{R}^{n,2}$, $x \in \mathbb{R}^n$ be given. Then the optimization problem*

$$\gamma^* := \arg\min_{\gamma \in \Omega_\gamma} L(\gamma), \quad L(\gamma) := \|x - S\gamma\|_2^2,$$
$$\Omega_\gamma := \{\gamma \in \mathbb{R}^2 \ | \ \gamma_1 + \gamma_2 = 1, \gamma_1, \gamma_2 \geq 0\}$$

has a solution

$$\gamma^* = [P_{[0,1]}(\alpha_1), P_{[0,1]}(\alpha_2)]^T, \ \ \alpha_1 = \frac{\langle x - S_2, S_1 - S_2 \rangle}{\|S_1 - S_2\|_2^2}, \alpha_2 = -\frac{\langle x - S_1, S_1 - S_2 \rangle}{\|S_1 - S_2\|_2^2}, \tag{40}$$

where $P_{[0,1]}(\alpha)$ is a projection of $\alpha \in \mathbb{R}$ onto interval $[0, 1]$ given by

$$P_{[0,1]}(\alpha) := \arg\min_{\beta \in [0,1]} (\alpha - \beta)^2 = \max\{0, \min\{1, \alpha\}\}. \tag{41}$$

Proof. Let us denote the columns of matrix $S = [S_1, S_2]$. The KKT optimality conditions (18), (19), (20), (21), (22) form the system

$$-2 \begin{bmatrix} S_1^T \\ S_2^T \end{bmatrix} x + 2 \begin{bmatrix} \langle S_1, S_1 \rangle & \langle S_1, S_2 \rangle \\ \langle S_2, S_1 \rangle & \langle S_2, S_2 \rangle \end{bmatrix} \gamma + \begin{bmatrix} \lambda_E \\ \lambda_E \end{bmatrix} - \begin{bmatrix} \lambda_{I_1} \\ \lambda_{I_2} \end{bmatrix} = 0, \tag{42}$$

$$\gamma_1 + \gamma_2 = 1, \tag{43}$$

$$\gamma_1, \gamma_2, \lambda_{I_1}, \lambda_{I_2} \geq 0, \tag{44}$$

$$\lambda_{I_1}\gamma_1 = \lambda_{I_2}\gamma_2 = 0, \tag{45}$$

Using the equality (43), we can eliminate variable $\gamma_2 = 1 - \gamma_1$ in (42). Additionally, we can substract the equations and after some manipulations we obtain

$$-\langle x - S_2, S_1 - S_2 \rangle + \gamma_1 \langle S_1 - S_2, S_1 - S_2 \rangle - \frac{\lambda_{I_1} - \lambda_{I_2}}{2} = 0.$$

Using the notation (40) for α_1 and including the remaining KKT conditions (44) and (45), we end up with the equivalent system

$$\gamma_1^* = \alpha_1 + \frac{\lambda_{I_1} - \lambda_{I_2}}{2}, \ \ 0 \leq \gamma_1^* \leq 1, \ \lambda_{I_1}, \lambda_{I_2} \geq 0, \ \lambda_{I_1}\gamma_1^* = \lambda_{I_2}(1 - \gamma_1^*) = 0. \tag{46}$$

The same system of equations and inequalities can be obtained as KKT system of projection optimization problem (41); here the Lagrange function is given by

$$\mathcal{L}(\beta, \lambda_I) := \alpha^2 - 2\alpha\beta + \beta^2 - \lambda_{I_2}\beta - \lambda_{I_1}(1-\beta)$$

and the KKT optimality conditions can be derived and modified as

$$\begin{gathered} \frac{\partial L}{\partial \beta} = -2\alpha + 2\beta - \lambda_{I_2} + \lambda_{I_1} = 0 \;\; \Rightarrow \;\; \beta^* = \alpha - \frac{\lambda_{I_1} - \lambda_{I_2}}{2}, \\ 0 \leq \beta^* \leq 1, \;\; \lambda_{I_1}, \lambda_{I_2} \geq 0, \;\; \lambda_{I_1}\beta^* = \lambda_{I_2}(1-\beta^*) = 0. \end{gathered} \tag{47}$$

We see that if we denote the output of projection as $\gamma_1^* = \beta^* = P_{[0,1]}(\alpha_1)$ (like in the presented solution (40)) then systems (47) and (46) are the same.

The similar process can be performed to obtain γ_2^*, however, in this case, we use $\gamma_1 = 1 - \gamma_2$ to eliminate variable in (42). □

Lemma 16 (Uniqueness of reconstruction with fixed S)**.** *Let* $[S^{1*}, \Gamma^{1*}]$ *and* $[S^{2*}, \Gamma^{2*}]$ *be two solutions of* (SPA$_2$) *for given data* X*. Let us denote the appropriate reconstructions by* $X^{\mathrm{rec1}} := S^{1*}\Gamma^{1*}$ *and* $X^{\mathrm{rec2}} := S^{2*}\Gamma^{2*}$*. If* $S^{1*} = S^{2*}$ *then* $X^{\mathrm{rec1}} = X^{\mathrm{rec2}}$*.*

Proof. From the optimality conditions, Γ^{1*} and Γ^{2*} solves (SPA$_2$) with fixed $S := S^{1*} = S^{2*}$. All solutions of corresponding QP for every $t = 1, \ldots, T$ differ by a vector from kernel of Hessian matrix (see (*7*), (*12*)) and using Lemma 21 we get

$$X^{\mathrm{rec1}} - X^{\mathrm{rec2}} = S \underbrace{(\gamma_t^{1*} - \gamma_t^{2*})}_{\in \mathrm{Ker}\, S^T S = \mathrm{Ker}\, S} = 0.$$

□

Computing optimal discretisations for Bayesian and Markovian models

Theorem 2. *Let $x_t \in \mathbb{R}^n$ and $y_t \in \mathbb{R}^m$ be two time series of length T, $X = [x_1, \dots, x_T] \in \mathbb{R}^{n,T}, Y = [y_1, \dots, y_T] \in \mathbb{R}^{m,T}$. The solution of (SPA$_2$) in the form*

$$[S_\varepsilon^*, \Gamma_x^*] = \arg\min_{\Gamma_x \in \Omega_\Gamma} \|X_\varepsilon - S_\varepsilon \Gamma_x\|_F^2 \tag{48}$$

with

$$X_\varepsilon := \begin{bmatrix} Y \\ \varepsilon X \end{bmatrix}, \quad S_\varepsilon := \begin{bmatrix} S_y \Lambda \\ \varepsilon S_x \end{bmatrix}, \tag{49}$$

and $\varepsilon \geq 0$ is equivalent to the solution of (SPA$_2$) problems

$$[S_x^*, \Gamma_x^*] := \arg\min_{\Gamma_x \in \Omega_\Gamma} \|X - S_x \Gamma_x\|_F^2, \tag{50}$$

$$[S_y^*, \Gamma_y^*] := \arg\min_{\Gamma_y \in \Omega_\Gamma} \|Y - S_y \Gamma_y\|_F^2, \tag{51}$$

in Tikhonov-sense with regularization parameter ε and $\Lambda \in \mathbb{R}^{K,T}$ is left-stochastic matrix of conditional probabilities such that the discrete Bayesian and Markovian model equations

$$\Gamma_y = \Lambda \Gamma_x, \tag{52}$$

are satisfied.

Proof. The combination of problems (50) and (51) into one optimization problem using Tikhonov-based approach is given by

$$[S_x^*, \Gamma_x^*, S_y^*, \Gamma_y^*] = \arg\min_{\Gamma_x, \Gamma_y \in \Omega_\Gamma} \|Y - S_y \Gamma_y\|_F^2 + \varepsilon \|X - S_x \Gamma_x\|_F^2, \tag{53}$$

where $\varepsilon \geq 0$ is a Tykhonov-regularisation parameter, controlling the relative importance of the X-discretisation problem with respect to the Y-discretisation problem. Substituting (52) into (53) and using the properties of Frobenius norm, we can write the objective function in form

$$\|Y - S_y \Gamma_y\|_F^2 + \varepsilon \|X - S_x \Gamma_x\|_F^2 = \left\| \begin{bmatrix} Y \\ \varepsilon X \end{bmatrix} - \begin{bmatrix} S_y \Gamma_y \\ \varepsilon S_x \Gamma_x \end{bmatrix} \right\|_F^2 = \left\| \begin{bmatrix} Y \\ \varepsilon X \end{bmatrix} - \begin{bmatrix} S_y \Lambda \\ \varepsilon S_x \end{bmatrix} \Gamma_x \right\|_F^2.$$

Getting use of (49) we can reformulate optimization problem (53) into form (48). □

Feature selection with SPA in the Euclidean space

Lemma 17. *Let $S \in \mathbb{R}^{n,K}$ be given. We consider $x \in \mathbb{R}^n$ and its small perturbation $x+d \in \mathbb{R}^n$. Let us denote γ_x^* and γ_{x+d}^* the optimal probabilistic discretisations of x and $x+d$ with respect to S, i.e.,*

$$\begin{array}{rcl@{\qquad}rcl} \gamma_x^* & := & \arg\min\limits_{\gamma\in\Omega_\gamma} L_x(\gamma), & L_x(\gamma) & := & \|x - S\gamma\|_2^2, \\ \gamma_{x+d}^* & := & \arg\min\limits_{\gamma\in\Omega_\gamma} L_{x+d}(\gamma), & L_{x+d}(\gamma) & := & \|(x+d) - S\gamma\|_2^2, \end{array} \tag{54}$$

and $\Omega_\gamma = \{\gamma \in \mathbb{R}^K : \sum\limits_{k=1}^{K} \gamma_k = 1 \ \wedge \ \gamma \geq 0\}$ is a feasible set. Then

$$\|\gamma_{x+d}^* - \gamma_x^*\|_{S^TS}^2 \leq \langle d, S(\gamma_{x+d}^* - \gamma_x^*)\rangle, \tag{55}$$

where $\|\gamma\|_{S^TS} = \sqrt{\langle S^TS\gamma, \gamma\rangle}$ is a seminorm on $\mathbb{R}^K$ induced by the scalar product with a symmetric positive semidefinite matrix S^TS.

Proof. Using Lemma 22 we state that the point γ^* is a solution of optimization problem if and only if

$$\langle \nabla L_x(\gamma_x^*), \gamma - \gamma_x^*\rangle \geq 0 \quad \forall \gamma \in \Omega_\gamma, \tag{56}$$

$$\langle \nabla L_{x+d}(\gamma_{x+d}^*), \gamma - \gamma_{x+d}^*\rangle \geq 0 \quad \forall \gamma \in \Omega_\gamma. \tag{57}$$

Since the feasible set is the same for both of optimization problems and consequently $\gamma_x^*, \gamma_{x+d}^* \in \Omega_\gamma$, we can choose $\gamma = \gamma_{x+d}^*$ in (56) and $\gamma = \gamma_x^*$ in (57). We get

$$\begin{array}{rcl} \langle \nabla L_x(\gamma_x^*), \gamma_{x+d}^* - \gamma_x^*\rangle & \geq & 0, \\ \langle \nabla L_{x+d}(\gamma_{x+d}^*), \gamma_x^* - \gamma_{x+d}^*\rangle & \geq & 0. \end{array}$$

and the sum of these inequalities gives us

$$\langle \nabla L_x(\gamma_x^*) - \nabla L_{x+d}(\gamma_{x+d}^*), \gamma_{x+d}^* - \gamma_x^*\rangle \geq 0. \tag{58}$$

The gradient of the continuously differentiable objective functions can be computed as

$$\nabla L_x(\gamma) = -2S^Tx + 2S^TS\gamma, \quad \nabla L_{x+d}(\gamma) = -2S^T(x+d) + 2S^TS\gamma,$$

and substituted into (58) to get

$$\langle S^T d - S^T S(\gamma^*_{x+d} - \gamma^*_x), \gamma^*_{x+d} - \gamma^*_x \rangle \geq 0.$$

Using the properties of a scalar product, we can rewrite this inequality as (55). □

Corollary 8. *Let us consider an arbitrary point $x \in \mathbb{R}^n$ and its perturbation in j-th feature*

$$x_h := x + he_j, \quad \{e_j\}_i := \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

Let us denote a so-called reconstruction of these points by $x^{\text{rec}}_x := S\gamma^*_x$ and $x^{\text{rec}}_{x_h} := S\gamma^*_{x_h}$. Since the seminorm on the left-hand side of (55) is non-negative, we get using simple subtitution

$$0 \leq \langle he_j, S(\gamma^*_{x+d} - \gamma^*_x) \rangle = h \left(\{x^{\text{rec}}_{x_h}\}_j - \{x^{\text{rec}}_x\}_j \right) = \left(\{x_{x_h}\}_j - \{x_x\}_j \right) \left(\{x^{\text{rec}}_{x_h}\}_j - \{x^{\text{rec}}_x\}_j \right)$$

We can conclude that the sign of the feature change in the data is the same as the sign of the feature change in corresponding reconstructions.

Corollary 9. *Using Cauchy-Bunyakovsky-Schwarz inequality we can further estimate (55) to form*

$$\|\gamma^*_{x+d} - \gamma^*_x\|^2_{S^T S} \leq \langle d, S(\gamma^*_{x+d} - \gamma^*_x) \rangle \leq \|d\| . \|\gamma^*_{x+d} - \gamma^*_x\|_{S^T S}$$

and therefore

$$\|\gamma^*_{x+d} - \gamma^*_x\|_{S^T S} \leq \|d\|$$

or using the notation for x^{rec}

$$\|x^{\text{rec}}_{x_1} - x^{\text{rec}}_{x_2}\| \leq \|x_1 - x_2\| \tag{59}$$

for any $x_1, x_2 \in \mathbb{R}^n$.

The original optimization problem can be rewritten as a projection problem to the set consisting the all possible reconstructed points $\Omega_{\text{rec}} \subset \mathbb{R}^n$

$$\begin{aligned} &\gamma^* = \arg\min_{\gamma \in \Omega_\gamma} \|x - S\gamma\|, \; x^{\text{rec}} = S\gamma^* \\ &\Updownarrow \\ &x^{\text{rec}} = P_{\Omega_{\text{rec}}}(x) := \arg\min_{y \in \Omega_{\text{rec}}} \|x - y\|, \;\; \Omega_{\text{rec}} := \{S\gamma, \gamma \in \Omega_\gamma\} \end{aligned}$$

and the projection is always non-expansive operator, i.e.,

$$\forall x_1, x_2 \in \mathbb{R}^n : \|P_{\Omega_{\text{rec}}}(x_1) - P_{\Omega_{\text{rec}}}(x_2)\| \leq \|x_1 - x_2\|.$$

Additionally, the distance between any $x_1^{\text{rec}}, x_2^{\text{rec}} \in \Omega_{\text{rec}}$ *can be bounded by the largest distance in the feasible set. In the case of the polytope* Ω_{rec}*, the largest distance is given by the largest distance between the vertices stored in columns of matrix* S*, i.e.,*

$$\|x_1^{\text{rec}} - x_2^{\text{rec}}\|_2 \leq \max_{k_1,k_2} \|S_{k_1} - S_{k_2}\|_2. \tag{60}$$

Theorem 3. *For sufficiently large* T*, let* $[S^*, \Gamma^*]$ *denote the solution of* (SPA$_2$) *for* $X \in \mathbb{R}^{n,T}$*. Let* $X^{\text{rec}}(X) := S^*(X)\Gamma^*(X)$ *denotes a reconstruction of the optimal discrete approximation of data* X*. Then for any dimension* $j = 1, \ldots, n$ *and any* $t = 1, \ldots, T$

1.) if $K = 2$ *then*

$$\left\| \frac{\partial X_{:,t}^{\text{rec}}}{\partial X_{j,t}} \right\|_2 \leq \frac{|S_{j,1}^* - S_{j,2}^*|}{\|S_{:,1}^* - S_{:,2}^*\|_2}, \tag{61}$$

2.) if $K \geq 2$ *then*

$$\left\| \frac{\partial X^{\text{rec}_{:,t}}}{\partial X_{j,t}} \right\|_2 \leq 1. \tag{62}$$

Proof. Using the chain rule we get

$$\frac{\partial X_{:,t}^{\text{rec}}}{\partial X_{j,t}} = \frac{\partial S^*(X)\Gamma_{:,t}^*(X)}{\partial X_{j,t}} = \frac{\partial S^*\Gamma_{:,t}^*(X)}{\partial S^*}\frac{\partial S^*(X)}{\partial X_{j,t}} + \frac{\partial S^*(X)\Gamma_{:,t}^*}{\partial \Gamma_{:,t}^*}\frac{\partial \Gamma_{:,t}^*(X)}{\partial X_{j,t}}$$

The first term represents the norm of derivate of the recontruction with fixed Γ^*. We already proved in Lemma 10 that the upper estimation of the norm of this derivative depends on the smallest eigenvalue of matrix $\Gamma\Gamma^T$. We will suppose that T is sufficiently large in a such way that the smallest eigenvalue is sufficiently large and therefore this norm is sufficiently small. In this case, the norm of derivative depends only on second term, i.e., we approximate

$$\left\| \frac{\partial X_{:,t}^{\text{rec}}}{\partial X_{j,t}} \right\|_2 \approx \left\| \frac{\partial S^*(X)\Gamma_{:,t}^*}{\partial \Gamma_{:,t}^*}\frac{\partial \Gamma_{:,t}^*(X)}{\partial X_{j,t}} \right\|_2 = \left\| S^* \frac{\partial \Gamma_{:,t}^*(X)}{\partial X_{j,t}} \right\|_2.$$

This value represents the norm of derivative of reconstruction with fixed S^*, therefore in the following proof we will suppose that S^* is fixed.

1.) In the case of $K = 2$, we can use an analytical solution of $\gamma^*(x_t) := \Gamma^*_{:,t}(X)$ provided by the Lemma 15. Since (for given $S = [S_{:,1}, S_{:,2}] \in \mathbb{R}^{n,2}$ and for any $x_t \in \mathbb{R}^n$)

$$\gamma_1^*(x_t) = \begin{cases} 0, & \text{if } \alpha_1 < 0 \\ 1, & \text{if } \alpha_1 > 1 \\ \alpha_1, & \text{elsewhere} \end{cases}, \quad \gamma_2^*(x_t) = \begin{cases} 0, & \text{if } \alpha_2 < 0 \\ 1, & \text{if } \alpha_2 > 1 \\ \alpha_2, & \text{elsewhere} \end{cases}$$

the derivatives are given by

$$\frac{\partial \gamma_1^*(x_t)}{\partial X_{j,t}} = \begin{cases} 0, & \text{if } \alpha_1 < 0 \text{ or } \alpha_1 > 1, \\ \frac{\partial \alpha_1}{\partial X_{j,t}}, & \text{elsewhere,} \end{cases} \quad \frac{\partial \gamma_2^*(x_t)}{\partial X_{j,t}} = \begin{cases} 0, & \text{if } \alpha_2 < 0 \text{ or } \alpha_2 > 1, \\ \frac{\partial \alpha_2}{\partial X_{j,t}}, & \text{elsewhere,} \end{cases} \tag{63}$$

where

$$\begin{aligned} \frac{\partial \alpha_1}{\partial X_{j,t}} &= \frac{\partial}{\partial X_{j,t}} \left(\frac{\langle x_t - S^*_{:,2}, S^*_{:,1} - S^*_{:,2} \rangle}{\|S^*_{:,1} - S^*_{:,2}\|_2^2} \right) = \frac{S^*_{j,1} - S^*_{j,2}}{\|S^*_{:,1} - S^*_{:,2}\|_2^2}, \\ \frac{\partial \alpha_2}{\partial X_{j,t}} &= \frac{\partial}{\partial X_{j,t}} \left(-\frac{\langle x_t - S^*_{:,1}, S^*_{:,1} - S^*_{:,2} \rangle}{\|S^*_{:,1} - S^*_{:,2}\|_2^2} \right) = -\frac{S^*_{j,1} - S^*_{j,2}}{\|S^*_{:,1} - S^*_{:,2}\|_2^2}. \end{aligned} \tag{64}$$

From (63), (64), and since $\alpha_1 + \alpha_2 = 1$ we can easily conclude that

$$\frac{\partial \gamma_1^*(x_t)}{\partial X_{j,t}} = -\frac{\partial \gamma_2^*(x_t)}{\partial X_{j,t}}, \quad \left| \frac{\partial \gamma_1^*(x_t)}{\partial X_{j,t}} \right| \leq \left| \frac{\partial \alpha_1^*(x_t)}{\partial X_{j,t}} \right|. \tag{65}$$

Using the linearity of derivative, the partial derivative of reconstruction $X^{\text{rec}}_{:,t}$ can be computed as

$$\frac{\partial X^{\text{rec}}_{:,t}}{\partial X_{j,t}} = \underbrace{\frac{\partial (S^* \gamma^*(x_t))}{\partial X_{j,t}}}_{\in \mathbb{R}^n} = S^* \underbrace{\frac{\partial \gamma^*(x_t)}{\partial X_{j,t}}}_{\in \mathbb{R}^K} = \underbrace{\frac{\partial \gamma_1^*(x_t)}{\partial X_{j,t}}}_{\in \mathbb{R}} \underbrace{S^*_{:,1}}_{\in \mathbb{R}^n} + \underbrace{\frac{\partial \gamma_2^*(x_t)}{\partial X_{j,t}}}_{\in \mathbb{R}} \underbrace{S^*_{:,2}}_{\in \mathbb{R}^n}$$

and using (65) we get

$$\begin{aligned} \left\| \frac{\partial X^{\text{rec}}_{:,t}}{\partial X_{j,t}} \right\|_2^2 &= \sum_{i=1}^n \left(\frac{\partial \gamma_1^*(x_t)}{\partial X_{j,t}} S^*_{i,1} + \frac{\partial \gamma_2^*(x_t)}{\partial X_{j,t}} S^*_{i,2} \right)^2 = \sum_{i=1}^n \left[(S^*_{i,1} - S^*_{i,2}) \left| \frac{\partial \gamma_1^*(x_t)}{\partial X_{j,t}} \right| \right]^2 \\ &\leq \underbrace{\left[\sum_{i=1}^n (S^*_{i,1} - S^*_{i,2})^2 \right]}_{= \|S^*_{:,1} - S^*_{:,2}\|_2^2} \left(\frac{|S^*_{j,1} - S^*_{j,2}|}{\|S^*_{:,1} - S^*_{:,2}\|_2^2} \right)^2 = \frac{(S^*_{j,1} - S^*_{j,2})^2}{\|S^*_{:,1} - S^*_{:,2}\|_2^2} \end{aligned}$$

2.) From a definition of the derivative we have

$$\frac{\partial X^{\mathrm{rec}}_{:,t}}{\partial X_{j,t}} := \lim_{h\to 0} \frac{X^{\mathrm{rec}}_{:,t,h} - X^{\mathrm{rec}}_{:,t}}{h},$$

where $X^{\mathrm{rec}}_{:,t,h}$ is reconstruction of point $X_{:,t,h}$ defined as $X_{:,t}$ with perturbated j-th feature, i.e.,

$$X_{:,t,h} := X_{:,t} + he_j, \quad \{e_j\}_i := \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

Since the reconstruction $X^{\mathrm{rec}}_{:,t}$ is continuous function of $X_{:,t}$, we can write

$$\left\| \frac{\partial X^{\mathrm{rec}}_{:,t}}{\partial X_{j,t}} \right\|_2^2 = \left\| \lim_{h\to 0} \frac{X^{\mathrm{rec}}_{:,t,h} - X^{\mathrm{rec}}_{:,t}}{h} \right\|_2^2 = \lim_{h\to 0} \frac{1}{h^2} \|X^{\mathrm{rec}}_{:,t,h} - X^{\mathrm{rec}}_{:,t}\|_2^2$$

The inner norm can be estimated using (59) to get

$$\lim_{h\to 0} \frac{1}{h^2} \|X^{\mathrm{rec}}_{:,t,h} - X^{\mathrm{rec}}_{:,t}\|_2^2 \leq \lim_{h\to 0} \frac{1}{h^2} \|X_{:,t,h} - X_{:,t}\|_2^2 = 1$$

□

Corollary 10. *The previous Lemma motivates for using the regularization of S-problem (25). In the case of $K = 2$, such a regularization minimizes the norm of derivative (61). In the case of general K, this regularization modifies the resulting polytope generated by S^* in a such way that this polytope is distinguishing between the features of reconstructed data, see (60).*

Corollary 11. *Please, notice that the dependence of reconstruction of X^{rec} on data X is linear and the respective derivative is piecewise constant, see Corollary after Lemma 14. In practice, we can estimate the norm in (62) using Euler method. Due to discontinuities in derivatives, such a method is exact for sufficiently small step h.*

Figures

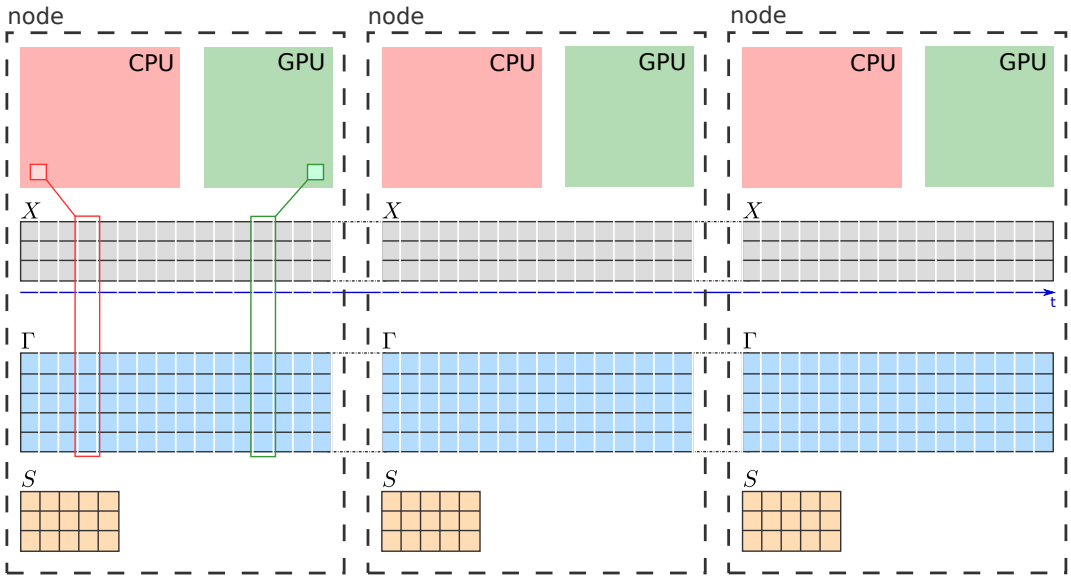
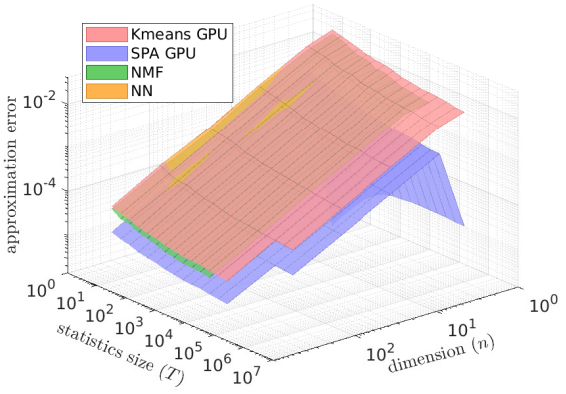
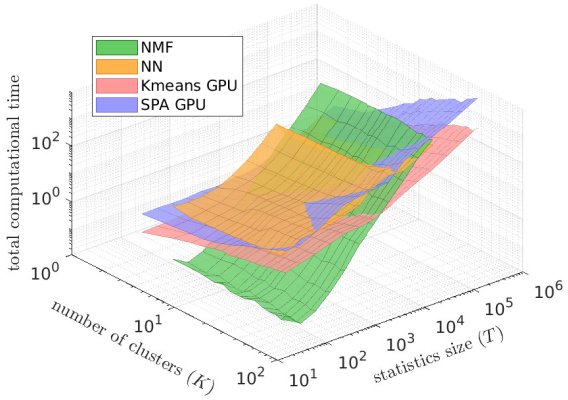


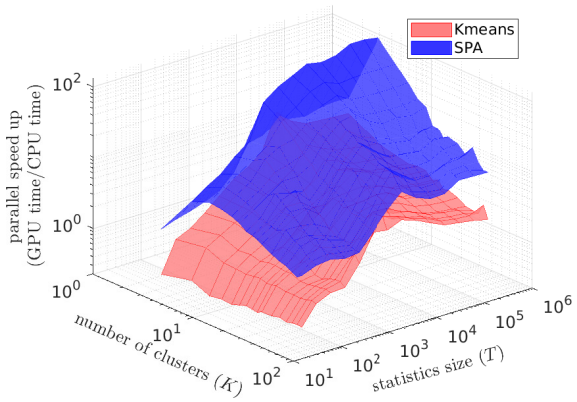
Figure S1: **Distributed solution of Γ-problem:** If objective function in (SPA), (SPA$_2$) is additively separable in t then the solution of optimization problem with fixed S can be composed as a solution of individual problems (see Lemma 4 and Lemma 11). In such a case, we can distribute T independent problems into several computation nodes such that the each node solves its own subset of problems. This local computation can be be performed by local CPU cores and/or using GPU cores, where (again) each core solves its individual subset of local optimization problems. Additionally, if we distribute the data of the problem in the same way, then each computational resource will have an access to its own local part of memory, without any additional communication.



(a) approximation quality scaling

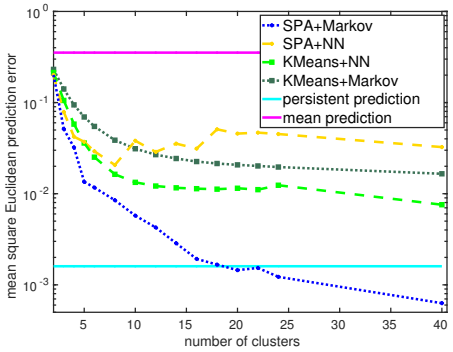


(b) computational cost scaling

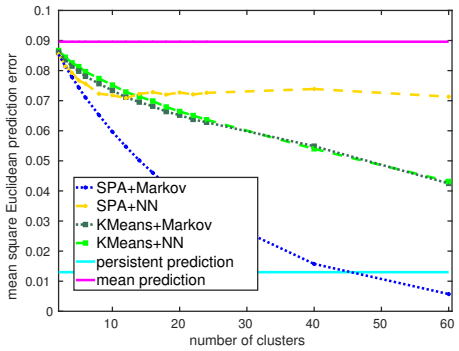


(c) parallelisability scaling

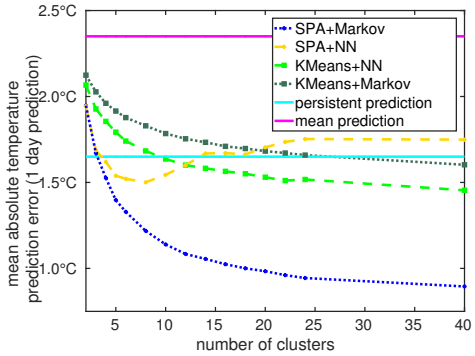
Figure S2: **Comparing computational cost (a), discretization quality (b) and parallelizability (c):** for (SPA_2) (blue surfaces), K-means clustering (dark-green), Nonnegative Matrix Factorisation (in its probabilistic variant called Left-Stochastic Decomposition (LSD), magenta surfaces) and the Self-Organising Maps (SOM, a special form of unsupervised neuronal networks used for discretization, orange surfaces). For every combination of data dimension n and the data statistics length T, methods are applied to 50 same randomly-generated data sets and the results in each of the curves represent averages over these 50 problems. Parallel speed-up in (c) is measured as the ratio of the average times time(GPU)/time(CPU) needed to reach the same relative tolerance threshold of 10^{-5} on a single Graphics Processing Unit (GPU, ASUS TURBO-GTX1080TI-11G, with 3584 CUDA cores) for time(GPU) versus a single CPU core (Intel Core i9-7900X CPU) for time(CPU). MATLAB script Fig1_reproduce.m reproducing these results is available for open access in the repository SPA at `https://github.com/SusanneGerber`.



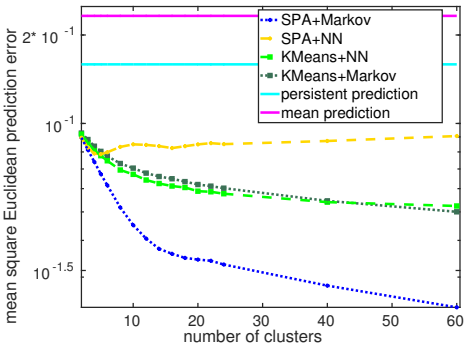
(a) Lorenz-96 1D turbulence model (weakly-chaotic regime)



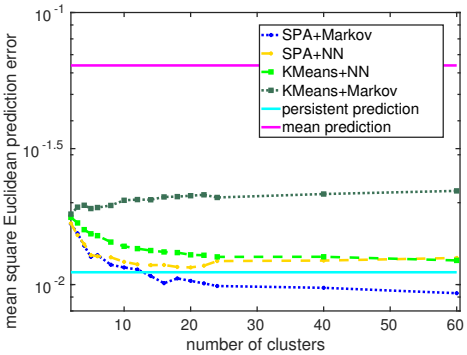
(b) Lorenz-96 1D turbulence model (strongly-chaotic regime)



(c) surface temperature dynamics over Europe (1979-2010, 20x30 grid ECMWF resimulation data)



(d) molecular dynamics simulation of 10-Alanine in water



(e) EG dynamics in a brain-computer interface (BCI2000 data)

Figure S3: Comparison of one-time-step predictions for a combination of SPA with Markov models (based on applications of the Theorem 2, blue lines) to the one-time-step predictions obtained by the standard prediction methods. The combination of SPA with Markov models is the only prediction scheme that outperforms the persistent prediction (i.e., when the next state is predicted to be the same as the current one) for all of the considered systems.

APPENDIX

Definition 1. *We say that point x^* is a minimizer of function f on given feasible set Ω, written as*

$$x^* = \arg\min_{x \in \Omega} f(x),$$

if (and only if) all points from the feasible set have larger or equal function value than $f(x^*)$, i.e.,

$$\forall x \in \Omega : f(x^*) \leq f(x).$$

Lemma 18. *Let $X \in \mathbb{R}^{n,T}, a, x \in \mathbb{R}^n, b \in \mathbb{R}^n, A = A^T \in \mathbb{R}^{n,n}$. Then*

$$\frac{\partial a^T X b}{\partial X} = ab^T, \quad \frac{\partial b^T X^T X b}{\partial X} = 2Xbb^T, \quad \frac{\partial x^T a}{\partial x} = a, \quad \frac{\partial x^T A x}{\partial x} = 2Ax.$$

Lemma 19. *Let $n, K, T \in \mathbb{N}$ and $A \in \mathbb{R}^{n,T}, B \in \mathbb{R}^{K,T}$. Then*

$$\sum_{t=1}^{T} A_{:,t}(B_{:,t})^T = AB^T \in \mathbb{R}^{n,K}.$$

Proof. From the definition of matrix-vector multiplication, the components of the result on left-hand side of the equation can be written in form (for every $i \in \{1, \ldots, n\}, j \in \{1, \ldots, K\}$)

$$\left[\sum_{t=1}^{T} A_{:,t}(B_{:,t})^T\right]_{i,j} = \sum_{t=1}^{T} A_{i,t}(B_{j,t})^T = \langle A_{i,:}, B_{j,:}\rangle = A_{i,:}(B_{j,:})^T,$$

which is a value of the corresponding matrix component on right-hand side of the equation. □

Lemma 20. *(of four fundamental subspaces): for any $B \in \mathbb{R}^{n,m}$ it holds*[3]

$$\operatorname{Ker} B \perp \operatorname{Im} B^T, \quad \operatorname{Im} B \perp \operatorname{Ker} B^T.$$

$$\operatorname{Ker} B \cup \operatorname{Im} B^T = \mathbb{R}^m, \quad \operatorname{Im} B \cup \operatorname{Ker} B^T = \mathbb{R}^n.$$

[3]Let $\mathcal{V}, \mathcal{W}$ be two subspaces of vector space $\mathcal{F}$ with scalar product $\langle ., . \rangle : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$. Then we say that $\mathcal{V} \perp \mathcal{W}$ if $\forall v \in \mathcal{V} \forall w \in \mathcal{W} : \langle v, w \rangle = 0$. Additionally we define $\mathcal{V} \cup \mathcal{W} := \{f \in \mathcal{F} : f \in \mathcal{V} \vee f \in \mathcal{W}\}$.

Proof. See Laub (*8*). □

Lemma 21. *Let* $n, K, T \in \mathbb{N}$ *and* $A \in \mathbb{R}^{n,T}, B \in \mathbb{R}^{K,T}$. *Then*

$$\operatorname{Ker} AA^T = \operatorname{Ker} A^T \subset \mathbb{R}^n, \tag{66}$$

$$\operatorname{Ker} B \subset \operatorname{Ker} AB \subset \mathbb{R}^K. \tag{67}$$

Proof. To prove (66), it is necessary to show that

$$\forall x \in \mathbb{R}^n : \quad AA^T x = 0 \quad \Leftrightarrow A^T A = 0.$$

($\Leftarrow$) Let us consider $x \in \mathbb{R}^m$ such that $A^T x = 0$. Then $AA^T x = A \underbrace{A^T x}_{=0} = 0$ (this also proves (67))

($\Rightarrow$) Let us consider $x \in \mathbb{R}^m$ such that $AA^T x = 0$. Using smart zero, we can write

$$0 = x^T 0 = x^T AA^T x = \|A^T x\|^2.$$

The norm of the vector is equal to zero if and only if the vector is equal to zero, therefore $A^T x = 0$.

□

Lemma 22. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *be a continuously differentiable convex function and let* $\Omega \subset \mathbb{R}^n$ *be closed convex set. Then* $x^* \in \Omega$ *is a solution of optimization problem*

$$x^* := \arg\min_{x \in \Omega} f(x)$$

if and only if

$$\forall x \in \Omega : \langle \nabla f(x), x - x^* \rangle \geq 0.$$

Proof. See (*3*), (*10*). □

References

1. G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26:2133–2159, 2005.

2. E. G. Birgin, J. M. Martínez, and M. M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10:1196–1211, 2000.

3. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, 1st edition, 2004.

4. Y. Chen and X. Ye. Projection onto a simplex. *Unpublished manuscript, arXiv:1101.6081*, 2011.

5. F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

6. Z. Dostál. On the decrease of a quadratic function along the projected-gradient path. *ETNA. Electronic Transactions on Numerical Analysis*, 31:25–29, 2008.

7. Z. Dostál. *Optimal Quadratic Programming Algorithms, with Applications to Variational Inequalities*, volume 23. SOIA, Springer, New York, US, 2009.

8. A. J. Laub. *Matrix Analysis For Scientists And Engineers*. Society for Industrial and Applied Mathematics, 2014.

9. P. Metzner, L. Putzig, and I. Horenko. Analysis of persistent non-stationary time series and applications. *Communications in Applied Mathematics and Computational Science*, 7(2):175–229, 2012.

10. J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2003.

11. L. Pospíšil and Z. Dostál. The projected Barzilai-Borwein method with fall-back for strictly convex QCQP problems with separable constraints. *Mathematics and Computers in Simulation*, 145:79–89, 2018.

12. L. Pospíšil, P. Gagliardini, W. Sawyer, and I. Horenko. On a scalable nonparametric denoising of time series signals. *Communications in Applied Mathematics and Computational Science*, 13:107–138, 2018.