On Machine Learning Knowledge Representation In The Form Of Partially Unitary Operator. Knowledge Generalizing Operator

Vladislav Gennadievich Malyshkin*

Ioffe Institute, Politekhnicheskaya 26, St Petersburg, 194021, Russia

(Dated: December, 22, 2022)

$Id: KnowledgeRepresentationAsUnitaryOperator.tex,v 1.465 2025/01/03 17:31:18 mal Exp $

A new form of ML knowledge representation with high generalization power is developed and implemented numerically. Initial IN attributes and OUT class label are transformed into the corresponding Hilbert spaces by considering localized wavefunctions. A partially unitary operator optimally converting a state from IN Hilbert space into OUT Hilbert space is then built from an optimization problem of transferring maximal possible probability from IN to OUT, this leads to the formulation of a new algebraic problem. Constructed Knowledge Generalizing Operator $\mathcal{U}$ can be considered as a IN to OUT quantum channel; it is a partially unitary rectangular matrix of the dimension $\dim(OUT)\times\dim(IN)$ transforming operators as $A^{OUT}=\mathcal{U}A^{IN}\mathcal{U}^{\dagger}$. Whereas only operator $\mathcal{U}$ projections squared are observable $\langle OUT|\mathcal{U}\,|\,IN\rangle^2$ (probabilities), the fundamental equation is formulated for the operator $\mathcal{U}$ itself. This is the reason of high generalizing power of the approach; the situation is the same as for the Schrödinger equation: we can only measure ψ^2, but the equation is written for ψ itself.

* malyshki@ton.ioffe.ru

I. INTRODUCTION

There are four key elements in any ML approach[1]:

- Attributes selection.
- Knowledge representation.
- Quality criteria (norm).
- Search algorithm to find the solution in knowledge representation space.

Knowledge representation is the most important element as it determines generalization power of a ML system. The progress in knowledge representation from linear regression coefficients, perceptron weights[2], statistical learning[3, 4], and logical approaches[5] to support vector machines[6], rules and decision trees[7], fuzzy logic[8, 9], and deep learning[10] has been the direction of ML development within the last two decades.

These approaches, however, share one common feature that limits their applicability. All of them typically construct a norm, loss function, penalty function, metric, distance function, etc. on class label (attributes to predict) difference from the target and perform it's optimization on training data. Selection of the norm is a complex task, moreover, the concept of "norm" is of statistical type and cannot be applied in every situation. In our earlier works[11, 12] we introduced a "norm-free" approach where the norm was replaced by projection operators. The idea takes inspiration in quantum mechanics where the outcomes of an observable f (obtained as an operator's spectrum $\left|f|\psi^{[i]}\right\rangle = \lambda^{[i]}\left|\psi^{[i]}\right\rangle$) and the probabilities of outcomes are separated; for a given state $|\psi\rangle$ the probabilities of $\lambda^{[i]}$ outcomes are obtained as projections to $\left|\psi^{[i]}\right\rangle$ eigenvectors $\left\langle\psi \,\middle|\, \psi^{[i]}\right\rangle^2$. This approach comes in two "flavors"[13]: interpolatory type (where the outcome is obtained as regular Radon–Nikodym derivative) and classification type (where the outcome is obtained as prior weight adjusted Radon–Nikodym derivative, a "Bayesian" style).

While these results are of great interest as they overcome one of the most difficult problem in ML (norm selection) and produce gauge-invariant solutions, they, as the other approaches to ML, still have a limitation in generalization power. The problem with this our approach[13] is that it is still of "joint distribution generalization" type. Effectively it constructs a joint distribution of (attributes, class label) pairs and then is trying to generalize from it. Some

ML approaches, such as statistical learning, support vector machines, rules and decision trees, Bayesian learning, etc. do this "joint distribution generalization" explicitly; the others, such as neural networks, hidden Markov model, almost all logic models, etc. in fact also do a "joint distribution generalization", but do it implicitly.

The problem with "joint distribution generalization" approaches is that they can only predict the outcomes that already have corresponding (attributes, class label) observations in training data. For example if we apply such an approach to periodic planetary motion – we obtain an accurate prediction, but applying it to a hyperbolic comet would be a failure as the comet only travel through the Solar system once. However, both (planet and hyperbolic comet) are governed by the same Newtonian laws and their motion is the phenomena of the same kind. Newtonian mechanics has a more powerful generalization than the "joint distribution generalization".

This work is the first work where we go beyond the "joint distribution generalization" in ML knowledge representation.

II. INPUT DATA AND SIMPLE MODELS

Whereas the developed approach can be applied to input data of various forms, for the purpose of comparison with well known models we will be considering only the data of supervised learning form[1]:

$$\begin{aligned}(x_0, x_1, \ldots, x_k, \ldots, x_{n-1})^{(l)} &\to (f_0, f_1, \ldots, f_j, \ldots, f_{m-1})^{(l)} \qquad \text{weight } \omega^{(l)} \\ \mathbf{x}^{(l)} &\to \mathbf{f}^{(l)}\end{aligned} \tag{1}$$

where an attributes vector $\mathbf{x}$ of the dimension n is mapped to a class label vector $\mathbf{f}$ of the dimension m for all $l = 1 \ldots M$ observations. An average $\langle\cdot\rangle$ is defined as the sum over all M observations sample:

$$\langle 1 \rangle = \sum_{l=1}^{M} \omega^{(l)} \tag{2}$$

$$\langle h(\mathbf{f}) g(\mathbf{x}) \rangle = \sum_{l=1}^{M} h(\mathbf{f}^{(l)}) g(\mathbf{x}^{(l)}) \omega^{(l)} \tag{3}$$

[1] The data can be possibly "producted" to some order $\mathcal{D}$. For example take n initial x_k and construct $x_{\mathbf{k}} = x_0^{k_0} x_1^{k_1} \ldots x_{n-1}^{k_{n-1}}$ with multi-index $\mathbf{k} = (k_0, k_1, \ldots, k_{n-1})$ subject to $\mathcal{D} = \sum_{j=0}^{n-1} k_j$. From initial n attributes x_k we now obtained $\mathcal{N}(n, \mathcal{D}) = C^{\mathcal{D}}_{n+\mathcal{D}-1}$ attributes $x_{\mathbf{k}}$ producted to the order $\mathcal{D}$, see [13].

Here $h(\mathbf{f})$ and $g(\mathbf{x})$ are some functions on $\mathbf{f}$ and $\mathbf{x}$, for example a polynomial or Christoffel function $K(x)$ from (10). In this paper we will be considering the models built on "moments" — some average of a polynomial function on x_k and f_j; an example of such an average is $\langle x_k x_{k'} f_j f_{j'} \rangle$. As a constant has always to be present in $\mathbf{x}$ and $\mathbf{f}$ bases the tensor $\langle x_k x_{k'} f_j f_{j'} \rangle$ includes all lower order averages such as $\langle x_k x_{k'} \rangle$ and $\langle f_j f_{j'} \rangle$. Introduce Gram matrices $G^{\mathbf{x}}_{kk'}$ and $G^{\mathbf{f}}_{jj'}$ for $\mathbf{x}$– and $\mathbf{f}$– spaces respectively:

$$G^{\mathbf{x}}_{kk'} = \langle x_k x_{k'} \rangle \tag{4}$$

$$G^{\mathbf{f}}_{jj'} = \langle f_j f_{j'} \rangle \tag{5}$$

We will assume that Gram matrices are non–degenerated, otherwise a regularization to be applied to $\mathbf{x}$ and $\mathbf{f}$ bases, see "*Appendix A: Regularization Example*" of [13].

A few familiar examples. Least squares solution of $\mathbf{f}$ on $\mathbf{x}$ requires Gram matrix $G^{\mathbf{x}}_{kk'}$ and $\langle f_j x_k \rangle$ moments as input to obtain $f_j(\mathbf{x}) = \sum_{k=0}^{n-1} \beta_k x_k$ as linear system solution:

$$\left\langle \left[f_j - \sum_{k=0}^{n-1} \beta_k x_k \right]^2 \right\rangle \to \min \tag{6}$$

$$f_j(\mathbf{x}) \approx \sum_{k,k'=0}^{n-1} x_k G^{\mathbf{x};-1}_{kk'} \langle f_j x_{k'} \rangle \tag{7}$$

Here $G^{\mathbf{x};-1}_{kk'}$ is Gram matrix (4) inverse. The (7) is m different predictors each one is applied to it's own class label component f_j, $j = 0 \ldots m-1$. Least squares knowledge representation model has limited predictive power and low outlier stability but it is very easy to implement numerically and obtained solution is gauge-invariant relatively an arbitrary non–degenerated linear transform of $\mathbf{x}$ and $\mathbf{f}$:

$$x'_k = \sum_{k'=0}^{n-1} T_{kk'} x_{k'} \tag{8a}$$

$$f'_j = \sum_{j'=0}^{m-1} T_{jj'} f_{j'} \tag{8b}$$

This often makes the least squares model the first choice to start data analysis despite all the drawbacks. The model has the properties similar to "joint distribution generalization" on the support of $\langle \cdot \rangle$ and typically diverges for $\mathbf{x}$ outside of the support interval; it has low generalization power.

Radon–Nikodym model consists in constructing a weight density $\psi^2_{\mathbf{y}}(\mathbf{x})$ localized at $\mathbf{x}=\mathbf{y}$ and then averaging $\mathbf{f}$ with it:

$$\psi_{\mathbf{y}}(\mathbf{x}) = \sqrt{K(\mathbf{y})}\sum_{i,k=0}^{n-1} y_i G^{\mathbf{x};\,-1}_{ik} x_k = \frac{\sum\limits_{i,k=0}^{n-1} y_i G^{\mathbf{x};\,-1}_{ik} x_k}{\sqrt{\sum\limits_{i,k=0}^{n-1} y_i G^{\mathbf{x};\,-1}_{ik} y_k}} = \frac{\sum\limits_{i=0}^{n-1} \psi^{[i]}(\mathbf{y})\psi^{[i]}(\mathbf{x})}{\sqrt{\sum\limits_{i=0}^{n-1}\left[\psi^{[i]}(\mathbf{y})\right]^2}} \tag{9}$$

$$K(\mathbf{x}) = \frac{1}{\sum\limits_{i,k=0}^{n-1} x_i G^{\mathbf{x};\,-1}_{ik} x_k} = \frac{1}{\sum\limits_{i=0}^{n-1}\left[\psi^{[i]}(\mathbf{x})\right]^2} \tag{10}$$

$$f_j(\mathbf{x}) \approx \frac{\langle\psi^2_{\mathbf{x}} f_j\rangle}{\langle\psi^2_{\mathbf{x}}\rangle} = \frac{\sum\limits_{i,q,s,k=0}^{n-1} x_i G^{\mathbf{x};\,-1}_{iq}\left\langle x_q x_s f_j\right\rangle G^{\mathbf{x};\,-1}_{sk} x_k}{\sum\limits_{i,k=0}^{n-1} x_i G^{\mathbf{x};\,-1}_{ik} x_k} = \frac{\sum\limits_{i,k=0}^{n-1}\psi^{[i]}(\mathbf{x})\left\langle\psi^{[i]}\,\middle|\, f_j\,\middle|\,\psi^{[k]}\right\rangle\psi^{[k]}(\mathbf{x})}{\sum\limits_{i=0}^{n-1}\left[\psi^{[i]}(\mathbf{x})\right]^2} \tag{11}$$

In Eq. (11) the Radon–Nikodym approximation is presented in two bases: original x_k, for which $\langle x_i x_k\rangle = G^{\mathbf{x}}_{ik}$, and in some orthogonalized basis $\left|\psi^{[i]}\right\rangle$ such that $\left\langle\psi^{[i]}\,\middle|\,\psi^{[k]}\right\rangle = \delta_{ik}$. Whereas in least squares approximation (7) the $f_j(\mathbf{x})$ is a linear combination of basis function x_k, in the Radon–Nikodym approximation (11) it is a ratio of two quadratic forms on basis function x_k with the matrices $\sum_{q,s=0}^{n-1} G^{\mathbf{x};\,-1}_{iq}\left\langle x_q x_s f_j\right\rangle G^{\mathbf{x};\,-1}_{sk}$ and $G^{\mathbf{x};\,-1}_{ik}$. By construction it is an averaging with positive weight[2] $\langle\psi^2 f_j\rangle \,/\, \langle\psi^2\rangle$ thus the bounds of f_j are preserved and the approximation (11) tends to a constant when some $x_k \to \infty$ The calculation requires Gram matrix $G^{\mathbf{x}}_{kk'}$ and $\langle x_k x_{k'} f_j\rangle$ moments as input (compare with $G^{\mathbf{x}}_{kk'}$ and $\langle x_k f_j\rangle$ required for least squares $f(\mathbf{x}) \approx \langle\psi_{\mathbf{x}}\,|\,f\rangle\,\psi_{\mathbf{x}}(\mathbf{x}) = \sum_{i,k=0}^{n-1} x_i G^{\mathbf{x};\,-1}_{ik}\langle x_k f\rangle$ approximation); the result is gauge–invariant relatively (8). The (11) is the solution of "interpolatory" type as it does not take into account "prior probabilities", see [13] for "classification" type solution with prior probabilities taken into account, a "Bayesian style".

A simple demonstration of localized states is presented in Fig. 1. For a simple chart a multi-dimensional vector $\mathbf{x}$ is constructed from 1D variable $x \in [-1:1]$ as $x_k = x^k$. The measure $\langle\cdot\rangle$ is taken as $\langle g\rangle = \int_{-1}^{1} g(x)dx$. Then $\psi^2_{\mathbf{y}}(\mathbf{x})$ can be considered as a function of

[2] For a given ψ the normalizing condition is $1 = \langle\psi^2\rangle$, this is required to properly average an observable $\langle f\psi^2\rangle$. In applications, however, the number of "covered" observations is often also required, for example to estimate possible data overfitting; the total coverage is $\langle 1\rangle$ (2). To estimate the number of observations covered by a given ψ one can use the Christoffel function $K(\mathbf{x})$ (10) to estimate the coverage as: $\text{Coverage}_{\psi} \approx \langle K\psi^2\rangle$. With an expansion of $K(\mathbf{x})$ in spectrum[13] $\left|K|\psi^{[i]}\right\rangle = \lambda^{[i]}\left|\psi^{[i]}\right\rangle$ one can obtain an expansion "by coverage"; this removes the major limitation of the principal components method: it's dependence on the scale of $\mathbf{x}$ attributes.

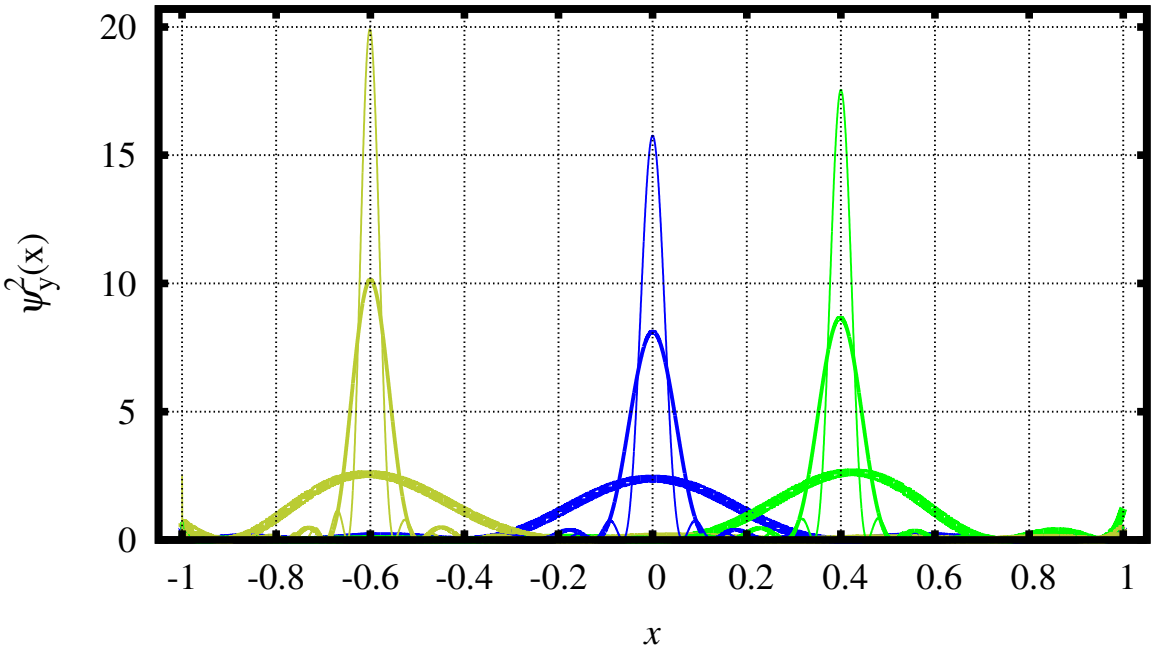

FIG. 1. A simple demonstration of (9) localized states $\psi^2_{\mathbf{y}}(\mathbf{x})$ for the measure $\langle g\rangle = \int_{-1}^{1} g(x)dx$ and the basis $\mathbf{x}$ constructed from 1D variable $x \in [-1:1]$ as $x_k = x^k$. The results for the states localized at $y = \{-0.6, 0, 0.4\}$ are presented as olive, blue, and green lines respectively. Basis dimension n is chosen as $\{7, 25, 50\}$ for thick, middle, and thin lines respectively.

scalar x and y as $\mathbf{x}$ and $\mathbf{y}$ vectors are calculated from the powers of x and y. In Fig. 1 we present $\psi^2_{-0.6}(x)$, $\psi^2_0(x)$, and $\psi^2_{0.4}(x)$. As expected the $\psi^2_y(x)$ density is localized near $x = y$; the localization becomes stronger with n increase. This chart demonstrates the main concept behind Radon–Nikodym type of interpolation which is a two–step process: on the first step a localized state $\psi^2_{\mathbf{y}}(\mathbf{x})$ is built and on the second step the value of an observable f is evaluated at $\mathbf{y}$ by averaging it with the weight obtained on the first step: $f(\mathbf{y}) \approx \langle \psi^2_{\mathbf{y}} f\rangle \,/\, \langle \psi^2_{\mathbf{y}}\rangle$. A trivial example of a square wave interpolation using least squares and Radon–Nikodym is presented in Fig. 2. We see that Radon–Nikodym preserves the bounds of f and has near interval edge oscillations very much suppressed because an interpolation of f at y is obtained by averaging f with always positive weight $d\mu = \psi^2_y(x)dx$.

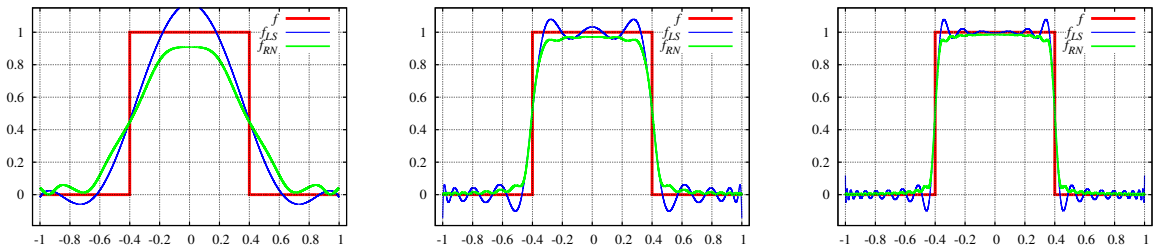

FIG. 2. A demonstration of a square wave interpolation (red) by least squares (blue, Eq. (7)) and Radon–Nikodym (green, Eq. (11)) with the measure $\langle g \rangle = \int_{-1}^{1} g(x)dx$ with $x \in [-1:1]$ for $n = \{7, 25, 50\}$ in the pictures: left, middle, and right respectively.

A. Pure Joint Distribution Model

In the section above we considered a simple problem of recovering $\mathbf{f}$ from $\mathbf{x}$ given sampled data (1). The least squares and Radon–Nikodym estimators (7) and (11) were obtained. They are using individual components of vector $\mathbf{f}$ as separate class labels; vector class label makes the study much more difficult than a scalar one. For further development we need, for attributes $\mathbf{x}$ and class label $\mathbf{f}$ of vector type, to have estimators of joint distribution $P(\mathbf{x}, \mathbf{f})$ probability and corresponding to it coverage.

There are several possible approaches to unify $\mathbf{x}$ and $\mathbf{f}$. In [14] the authors introduced a new vector $\mathbf{z}$ of the dimension $n + m$

$$\mathbf{z} = (x_0, x_1, \ldots, x_k, \ldots, x_{n-1}, f_0, f_1, \ldots, f_j, \ldots, f_{m-1}) \tag{12}$$

and constructed Christoffel function from it (this requires all $\langle x_k x_{k'} \rangle$, $\langle f_j f_{j'} \rangle$, and $\langle x_k f_j \rangle$ moments). Maximizing Christoffel function on $\mathbf{f}$ given $\mathbf{x}$ exhibits very promising results. However, a difficulty with cross-terms arise[13] both in data initial regularization and in interpretation of the final result.

To deal with vector class label $\mathbf{f}$ and, for further generalization of Section III below, we will use $\mathbf{f}$-localized states. For sampled $\mathbf{f}$ data, possibly producted to some order, construct Gram matrix in $\mathbf{f}$-space (5) and, the same as in (9), build a localized state $\psi_{\mathbf{g}}(\mathbf{f})$:

$$\psi_{\mathbf{g}}(\mathbf{f}) = \frac{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};-1}_{jj'} f_{j'}}{\sqrt{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};-1}_{jj'} g_{j'}}} = \frac{\sum\limits_{j=0}^{m-1} \psi^{[i]}(\mathbf{g}) \psi^{[j]}(\mathbf{f})}{\sqrt{\sum\limits_{j=0}^{m-1} \left[\psi^{[j]}(\mathbf{g})\right]^2}} \tag{13}$$

For each observation $l = 1 \dots M$ consider (14) projection of $\mathbf{x}^{(l)}$–localized state (9) to $\mathbf{f}^{(l)}$–localized state (13) then sum it over the entire $l = 1 \dots M$ sample to obtain the number of covered observations $\mathcal{F}^{JDG}$ (note: there is a "projective" factor $\langle f_j x_k \rangle$ in the expression)

$$\langle \psi_{\mathbf{g}} \,|\, \psi_{\mathbf{y}} \rangle^2 = \frac{\left| \sum\limits_{k,k'=0}^{n-1} \sum\limits_{j,j'=0}^{m-1} g_{j'} G^{\mathbf{f};-1}_{j'j} \langle f_j x_k \rangle G^{\mathbf{x};-1}_{kk'} y_{k'} \right|^2}{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};-1}_{jj'} g_{j'} \sum\limits_{k,k'=0}^{n-1} y_k G^{\mathbf{x};-1}_{kk'} y_{k'}} \tag{14}$$

$$\langle 1 \rangle \geq \mathcal{F}^{JDG} = \sum_{l=1}^{M} \langle \psi_{\mathbf{f}^{(l)}} \,|\, \psi_{\mathbf{x}^{(l)}} \rangle^2 \omega^{(l)} \tag{15}$$

If $\mathbf{x}$ and $\mathbf{f}$ form the same vector space then $\mathcal{F}^{JDG} = \langle 1 \rangle$. Otherwise, for example when $\mathbf{x}$ contains the entire $\mathbf{f}$ plus one more completely random attribute, $\mathcal{F}^{JDG} < \langle 1 \rangle$. Since (14) has normalizing terms containing $G^{\mathbf{f};-1}_{jj'}$ and $G^{\mathbf{x};-1}_{kk'}$ matrices in the denominator, to calculate (15) the secondary sampling technique[15] is required. The Gram matrices $G^{\mathbf{x}}_{kk'}$ (4) and $G^{\mathbf{f}}_{jj'}$ (5) are calculated first then the projection (14) is calculated for every observation $l = 1 \dots M$ and used in (15) as it were plain observed at observation l. Technically this means we need to calculate the moments of $\mathbf{x}$- and $\mathbf{f}$- Christoffel functions product: $\left\langle x_k f_j \,\middle|\, K^{(\mathbf{x})} K^{(\mathbf{f})} \,\middle|\, x_{k'} f_{j'} \right\rangle$ (33).

The $\langle \psi_{\mathbf{g}} \,|\, \psi_{\mathbf{y}} \rangle^2$ can be viewed as joint distribution of $\mathbf{f}$ and $\mathbf{x}$. For a given $\mathbf{x}$ the probabilities of various $\mathbf{f}$ can be estimated as

$$P(\mathbf{f}, \mathbf{x}) \approx \langle \psi_{\mathbf{f}} \,|\, \psi_{\mathbf{x}} \rangle^2 \tag{16}$$

The (16) estimates the probability of possible outcome $\mathbf{f}$ given some fixed value of $\mathbf{x}$; the estimation is based on (attributes, class label) pairs observed in the training sample. A typical step from here is to find a subspace of $\mathbf{x}$ providing the best prediction of $\mathbf{f}$, optimal clustering [13] is a typical approach in this direction. However, we want to go beyond "joint distribution generalization", beyond finding a subspace of $\mathbf{x}$ providing the best prediction of $\mathbf{f}$ in terms of the probability $P(\mathbf{f}, \mathbf{x})$ estimated on *training sample*. We need a more powerful generalization method, possibly applicable to not yet observed values of $\mathbf{x}$ and $\mathbf{f}$.

III. ON KNOWLEDGE GENERALIZING OPERATOR

In Section II A above we transformed original $\mathbf{x}^{(l)} \to \mathbf{f}^{(l)}$ data sample (1) to a sequence of $\mathbf{x}^{(l)}$- and $\mathbf{f}^{(l)}$- localized states according to (9) and (13):

$$\psi_{\mathbf{x}^{(l)}} \to \psi_{\mathbf{f}^{(l)}} \qquad\qquad \text{weight } \omega^{(l)} \tag{17}$$

As $\psi_{\mathbf{x}}$ / $\psi_{\mathbf{f}}$ are defined by n / m coefficients before x_k / f_j the (17) is nothing more than a transform of the original data (1). This is not a regular linear transform of $\mathbf{x}$ / $\mathbf{f}$ bases, this is a linear transform with $G^{\mathbf{x};-1}_{kk'}$ / $G^{\mathbf{f};-1}_{jj'}$ matrices *followed by* normalization to 1 with Christoffel function as in (9) / (13).

The purpose of this transform is to obtain the states we can project to each other or to some other basis; for example: $|\psi_{\mathbf{x}^{(l)}}\rangle = \sum_{k=0}^{n-1} \left|\psi^{[k]}\right\rangle \left\langle\psi^{[k]} \,\middle|\, \psi_{\mathbf{x}^{(l)}}\right\rangle$ with $\left|\psi^{[k]}\right\rangle$ being an orthogonal full basis in $\mathbf{x}$-space, $1 = \langle\psi_{\mathbf{x}^{(l)}} \,|\, \psi_{\mathbf{x}^{(l)}}\rangle^2$, $1 = \langle\psi_{\mathbf{f}^{(l)}} \,|\, \psi_{\mathbf{f}^{(l)}}\rangle^2$, $1 \geq \langle\psi_{\mathbf{x}^{(l)}} \,|\, \psi_{\mathbf{f}^{(l)}}\rangle^2$, etc. The main result of Section II A was to obtain joint $(\mathbf{x}, \mathbf{f})$ distribution (16) and then trying to generalize from it.

Consider a different form of generalization. Let $|\psi_{\mathbf{x}^{(l)}}\rangle$, before being used in calculation of joint distribution, is transformed by a unitary operator $\|\mathcal{U}\|$:

$$\mathcal{F} = \sum_{l=1}^{M} \langle\psi_{\mathbf{f}^{(l)}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{x}^{(l)}}\rangle^2 \, \omega^{(l)} \tag{18}$$

Contrary to (16) this expression is transforming $\psi_{\mathbf{x}^{(l)}}(\mathbf{x})$ to some other function in $\mathbf{x}$-space $|\psi(\mathbf{x})\rangle = |\mathcal{U}|\psi_{\mathbf{x}^{(l)}}(\mathbf{x})\rangle$ and only then projecting the result to actual realization $\psi_{\mathbf{f}^{(l)}}(\mathbf{f})$ in $\mathbf{f}$-space. In some sense the $\|\mathcal{U}\|$ can be viewed as a Scattering Amplitude Matrix, as it relates the *IN* state $|\psi_{\mathbf{x}^{(l)}}\rangle$ with the *OUT* state $|\psi_{\mathbf{f}^{(l)}}\rangle$. All the information about what combinations of attributes x_k to be used for prediction now contains in operator $\|\mathcal{U}\|$. It is called *Knowledge Generalizing Operator*. The operator is unitary (to preserve normalizing) $1 = \left\langle\psi \left| \mathcal{U}^\dagger \right| \mathcal{U} \right| \psi\right\rangle$.

$$\mathcal{U}^\dagger\mathcal{U} = \mathbb{1} \tag{19}$$

In our model the knowledge is represented in the form of a unitary operator. This is a very common form in physics: the dynamics in classical mechanics, electrodynamics, quantum mechanics can be represented as a sequence of infinitesimal unitary transformations determined by Hamiltonian (or Lagrangian) of the system. The (18) is an inverse problem: given (1) data find unitary operator $\|\mathcal{U}\|$ maximizing (18) coverage subject to (19) constraints.

Whereas the (18) is of fourth order in $|\psi\rangle$, it is of second order in $\|\mathcal{U}\|$. The constraints (19) is also of second order in $\|\mathcal{U}\|$. Thus the problem of finding the *Knowledge Generalizing Operator* is a variant of quadratically constrained quadratic program[16] (QCQP).

Before we go further, let us consider a simplified version of (18) to obtain $\mathcal{F}$ upper limit for "projective" interpretation of operator $\|\mathcal{U}\|$. Consider the problem of finding (in $\mathbf{x}$-space) orthogonal basis $\phi^{[i]}$, a subset of full basis $D \leq n$, that maximizes $\mathcal{F}^{TOT}$:

$$\phi^{[i]} = \sum_{k=0}^{n-1} \alpha_k^{\phi;[i]} x_k \qquad i = 0 \dots D-1; \;\; D \leq n \tag{20}$$

$$\delta_{ii'} = \left\langle \phi^{[i]} \,\middle|\, \phi^{[i']} \right\rangle = \sum_{k,k'=0}^{n-1} \alpha_k^{\phi;[i]} G^{\mathbf{x}}_{kk'} \alpha_{k'}^{\phi;[i']} \tag{21}$$

$$\mathcal{F}^{TOT} = \sum_{l=1}^{M} \sum_{i=0}^{D-1} \left\langle \psi_{\mathbf{f}^{(l)}} \,\middle|\, \phi^{[i]} \right\rangle^2 \omega^{(l)} \tag{22}$$

Substituting (20) to (13) obtain:

$$K^{(\mathbf{f})}(\mathbf{g}) = \frac{1}{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};-1}_{jj'} g_{j'}} \tag{23}$$

$$\left\langle f_t \,\middle|\, K^{(\mathbf{f})} \,\middle|\, f_s \right\rangle = \sum_{l=1}^{M} \frac{f_t^{(l)} f_s^{(l)}}{\sum\limits_{j,j'=0}^{m-1} f_j^{(l)} G^{\mathbf{f};-1}_{jj'} f_{j'}^{(l)}} \omega^{(l)} \tag{24}$$

$$K^{(\mathbf{f}\to\mathbf{x})}_{ik} = \sum_{k',t',s',j'=0}^{m-1} \langle x_i f_{k'} \rangle \, G^{\mathbf{f};-1}_{k't'} \left\langle f_{t'} \,\middle|\, K^{(\mathbf{f})} \,\middle|\, f_{s'} \right\rangle G^{\mathbf{f};-1}_{s'j'} \langle x_k f_{j'} \rangle \quad i,k = 0 \dots n-1 \tag{25}$$

The (24) is $\mathbf{f}$–Christoffel function (23) moments[3]. The (25) allows to present (22) in the form:

$$\mathcal{F}^{TOT} = \sum_{i=0}^{D-1} \sum_{k,k'=0}^{n-1} \alpha_k^{\phi;[i]} K^{(\mathbf{f}\to\mathbf{x})}_{kk'} \alpha_{k'}^{\phi;[i]} \tag{26}$$

From which we can spectrally expand the $\mathcal{F}^{TOT}$ by solving a generalized eigenvalue problem with the matrices $K^{(\mathbf{f}\to\mathbf{x})}_{kk'}$ and $G^{\mathbf{x}}_{kk'}$ in left- and right- hand sides:

$$\sum_{k'=0}^{n-1} K^{(\mathbf{f}\to\mathbf{x})}_{kk'} \alpha_{k'}^{\phi;[i]} = \lambda^{[i]} \sum_{k'=0}^{n-1} G^{\mathbf{x}}_{kk'} \alpha_{k'}^{\phi;[i]} \tag{27}$$

$$\mathcal{F}^{TOT} = \sum_{i=0}^{D-1} \lambda^{[i]} \tag{28}$$

---

[3] One can also consider $\left\langle \frac{\partial R}{\partial f_t} \frac{\partial R}{\partial f_s} \right\rangle$ with $R(\mathbf{f}) = 1/\sqrt{K(\mathbf{f})} = \sqrt{\sum\limits_{j,j'=0}^{m-1} f_j G^{\mathbf{f};-1}_{jj'} f_{j'}}$.

The (28) is a spectral decomposition of (22), it has at most m non–zero eigenvalues (the rank of (25) is m or lower, we also assume $m \leq n$). If $\mathbf{f}$ belongs to a subspace of $\mathbf{x}$ then the sum of these m eigenvalues in (28) is equal to $\langle 1 \rangle$. The (28) takes all possible vectors from $\mathbf{x}$-space and project them to all $|\psi_{\mathbf{f}^{(l)}}\rangle$ summing the coverage, this operation does not make any inference, it estimates the coverage (18) upper limit for any norm–preserving projective transform[17], such as $|\mathcal{U}|\psi\rangle$ (53) or, more generally, (54). The estimation can be obtained from $K_{kk'}^{(\mathbf{f}\to\mathbf{x})}$ and $G_{kk'}^{\mathbf{x}}$ matrices even without solving the eigenvalue problem (27):

$$\mathcal{F}^{TOT} = \sum_{k,k'=0}^{n-1} K_{kk'}^{(\mathbf{f}\to\mathbf{x})} G_{k'k}^{\mathbf{x};\,-1} \tag{29}$$

For calculation see `com/polytechnik/kgo/KGOSolutionVectorXVectorF.java:FTOT` which is used in unit tests.

A simpler approach to construct contributing to coverage subspace $\left|\phi^{[i]}\right\rangle$ is to notice that in (18) there are scalar products $\langle f_j x_k \rangle$ of the vectors from $\mathbf{x}$ and $\mathbf{f}$ spaces. Thus we can project the $\mathbf{f}$-space to $\mathbf{x}$-space; to split $\mathbf{x}$ into two subspaces: $\left|\phi^{P;[j]}\right\rangle$ "projected" (of the dimension $D \leq m$) and $\left|\phi^{O;[k]}\right\rangle$ "orthogonal" to $\mathbf{f}$ (of the dimension $n - D$), all vectors from the second one have zero scalar product with a state in $\mathbf{f}$-space $\left\langle \phi^{O;[k]} \,\middle|\, \psi_{\mathbf{f}} \right\rangle = 0$, thus the $\left|\phi^{O;[k]}\right\rangle$ does not contribute to coverage (18). For this reason it is sufficient to consider operator $\|\mathcal{U}\|$ to have the dimension $D \times n$ converting a vector from $\mathbf{x}$-space to $\left|\phi^{P;[j]}\right\rangle$, i.e. to use $\left|\phi^{[i]}\right\rangle = \left|\phi^{P;[i]}\right\rangle$ as contributing subspace, see `com/polytechnik/kgo/TestKGO.java:orthogonalizeU` for an implementation.

Let us define operator $\|\mathcal{U}\|$ to be a matrix (in this paper u_{sk} is considered to be a real matrix, a generalization to a complex matrix is straightforward) of $D \times n$, $D \leq m$, such that:

$$|\mathcal{U}|x_k\rangle = \sum_{s=0}^{D-1} \left|\phi^{[s]}\right\rangle u_{sk} \tag{30}$$

Then (note: there is a "projective" factor $\langle f_j x_k \rangle$ in the expression, from $\left\langle f_j \phi^{[s]} \right\rangle$)

$$\langle \psi_{\mathbf{g}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{y}} \rangle^2 = \frac{\left| \sum\limits_{k,k'=0}^{n-1} \sum\limits_{j,j'=0}^{m-1} \sum\limits_{s=0}^{D-1} g_{j'} G_{j'j}^{\mathbf{f};\,-1} \left\langle f_j \phi^{[s]} \right\rangle u_{sk} G_{kk'}^{\mathbf{x};\,-1} y_{k'} \right|^2}{\sum\limits_{j,j'=0}^{m-1} g_j G_{jj'}^{\mathbf{f};\,-1} g_{j'} \sum\limits_{k,k'=0}^{n-1} y_k G_{kk'}^{\mathbf{x};\,-1} y_{k'}} \tag{31}$$

$$\left\langle \phi^{[s]} \,\middle|\, \phi^{[q]} \right\rangle = \sum_{k,k'=0}^{n-1} u_{sk} \langle x_k \,|\, x_{k'} \rangle u_{qk'} \qquad s, q = 0 \ldots D-1 \tag{32}$$

The (31) is actually (14) but $|\psi_{\mathbf{y}}\rangle$ is replaced by $|\mathcal{U}|\psi_{\mathbf{y}}\rangle$. This is the central concept of knowledge generalizing operator: the state the inference is based on $|\psi_{\mathbf{y}}\rangle$ is transformed by the operator $\|\mathcal{U}\|$ before coupling with the state $|\psi_{\mathbf{f}}\rangle$ we are looking an inference to. Partial unitarity constraint (32) corresponds to the fact that only subspace of the dimension $D \le m$ can possibly contribute to the coverage (18). When only a subspace of $\mathbf{x}$ contributes to (35) the problem to find a unitary matrix u_{jk} becomes highly degenerative. While the algorithm described in the Appendix A below works well with such a degenerative problem, it is beneficial for both: computational complexity and simplicity of result's interpretation to make the problem less degenerative. Consider a $\left|\phi^{[i]}\right\rangle$ $i = 0 \dots D-1$ subspace of the dimension $D \le m$. Let us split considered above unitary operator $\|\mathcal{U}\|$ into $\|\mathcal{U}\| = \|\mathcal{U}^P\| + \|\mathcal{U}^O\|$ such that $\|\mathcal{U}^P\|$ transforms any $\mathbf{x}$-vector to $\left|\phi^{[i]}\right\rangle$ subspace, and $\|\mathcal{U}^O\|$ transforms any $\mathbf{x}$-vector to a subspace orthogonal to $\left|\phi^{[i]}\right\rangle$ (this split is most easy to perform if to convert original $|x_k\rangle$ space into direct sum of $\left|\phi^{[i]}\right\rangle$ and orthogonal to $\left|\phi^{[i]}\right\rangle$ subspaces). Then, because $0 = \left\langle \psi_{\mathbf{f}} \left| \mathcal{U}^O \right| \psi_{\mathbf{x}} \right\rangle$ for any $\mathbf{f}$ and $\mathbf{x}$, optimization result of $\mathcal{F}$ does not depend on $\|\mathcal{U}^O\|$, thus it is sufficient to find an operator $\|\mathcal{U}^P\|$ of the dimension $D \times n$ subject to (32) constraint.

To calculate (18) it is convenient to introduce the moments of Christoffel functions product:

$$\left\langle x_k f_j \left| K^{(\mathbf{x})} K^{(\mathbf{f})} \right| x_{k'} f_{j'} \right\rangle = \sum_{l=0}^{M} \omega^{(l)} \frac{x_k^{(l)} x_{k'}^{(l)}}{\sum\limits_{q,q'=0}^{n-1} x_q^{(l)} G_{qq'}^{\mathbf{x};-1} x_{q'}^{(l)}} \cdot \frac{f_j^{(l)} f_{j'}^{(l)}}{\sum\limits_{s,s'=0}^{m-1} f_s^{(l)} G_{ss'}^{\mathbf{f};-1} f_{s'}^{(l)}} \tag{33}$$

to write $\mathcal{F}$ in the form (note: there is a "projective" factor $\langle f_j x_k \rangle$ in the expression, from $\left\langle f_j \phi^{[s]} \right\rangle$)

$$S_{sk;s'k'} = \sum_{j,j',q,q'=0}^{m-1} \sum_{t,t'=0}^{n-1} \left\langle x_t f_{j'} \left| K^{(\mathbf{x})} K^{(\mathbf{f})} \right| x_{t'} f_{q'} \right\rangle G_{j'j}^{\mathbf{f};-1} \left\langle f_j \phi^{[s]} \right\rangle G_{kt}^{\mathbf{x};-1} G_{q'q}^{\mathbf{f};-1} \left\langle f_q \phi^{[s']} \right\rangle G_{k't'}^{\mathbf{x};-1} \tag{34}$$

$$\mathcal{F} = \sum_{s,s'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{sk} S_{sk;s'k'} u_{s'k'} = \sum_{l=1}^{M} \left\langle \psi_{\mathbf{f}^{(l)}} \left| \mathcal{U} \right| \psi_{\mathbf{x}^{(l)}} \right\rangle^2 \omega^{(l)} \underset{\mathcal{U}}{\longrightarrow} \max \tag{35}$$

The $\mathcal{F}$ is a quadratic function on u_{sk}; the expression for $S_{sk;s'k'}$ can be greatly simplified if $\mathbf{x}$- and $\mathbf{f}$- bases are initially regularized (see [13], "*Appendix A: Regularization Example*"). In an orthogonal basis Gram matrix is a unit matrix, thus the $G^{\mathbf{x};-1}$ and $G^{\mathbf{f};-1}$ get removed in (34).

A. On Knowledge Generalizing Operator With Different *IN* and *OUT* Spaces

In the section above we considered operator $\|\mathcal{U}\|$ as $\mathbf{x} \to \mathbf{x}$ transform. In (18) the $\langle \psi_{\mathbf{f}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{x}} \rangle$ was understood as $\mathbf{x} \to \mathbf{x}$ transform $|\mathcal{U}|\psi_{\mathbf{x}}\rangle$ followed by projection of the result to $|\psi_{\mathbf{f}}\rangle$-space (31); similar "projective" interpretation was used in (14). This interpretation of $\|\mathcal{U}\|$ lead us to "contributing subspace" $\left|\phi^{[s]}\right\rangle$ (20) (which is a subspace of $\mathbf{x}$), equation (31) for $\langle \psi_{\mathbf{f}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{x}} \rangle^2$ (it has $\langle f_j x_k \rangle$ projective factors) and (32) constraints with the meaning of scalar product invariance. Optimization problem (35) for u_{jk} matrix of the dimension $D \times n$ allows to determine partially unitary operator $\|\mathcal{U}\|$. This operator has both *IN* and *OUT* subspaces being a subspace of $\mathbf{x}$.

A natural generalization is to consider an operator $\|\mathcal{U}\|$ with *different* subspaces for *IN* and *OUT*, this way we can avoid any kind of "projection" what would greatly increase generalizing power of the approach. Let us consider $\mathbf{x} \to \mathbf{f}$ transform directly. Now u_{jk} is a $m \times n$ matrix transforming a vector from $\mathbf{x}$-space to $\mathbf{f}$-space

$$f_j = \sum_{k=0}^{n-1} u_{jk} x_k \qquad\qquad j = 0 \ldots m-1 \tag{36}$$

In a common "projective" paradigm the (36) is multiplied by $x_{k'}$, then after taking the average — least squares (7) are obtained. Now it is different — we cannot take scalar products $\langle f_j x_k \rangle$ as $\mathbf{f}$ and $\mathbf{x}$ belong to different Hilbert spaces. We multiply (36) by *itself* and take the average — obtain (38) constraint. Substituting (36) to localized state (13) obtain

$$\langle \psi_{\mathbf{g}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{y}} \rangle^2 = \frac{\left| \sum\limits_{k=0}^{n-1} \sum\limits_{j,s=0}^{m-1} g_j G^{\mathbf{f};\,-1}_{js} u_{sk} y_k \right|^2}{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};\,-1}_{jj'} g_{j'} \sum\limits_{k,k'=0}^{n-1} y_k G^{\mathbf{x};\,-1}_{kk'} y_{k'}} \tag{37}$$

$$\langle f_j \,|\, f_{j'} \rangle = \sum_{k,k'=0}^{n-1} u_{jk} \langle x_k \,|\, x_{k'} \rangle u^*_{j'k'} \qquad\qquad j, j' = 0 \ldots m-1 \tag{38}$$

thus the optimization problem does not contain any "projective" factors $\langle f_j x_k \rangle$

$$S_{sk;s'k'} = \sum_{j,j'=0}^{m-1} \left\langle f_j x_k \,\middle|\, K^{(\mathbf{x})} K^{(\mathbf{f})} \,\middle|\, f_{j'} x_{k'} \right\rangle G^{\mathbf{f};\,-1}_{js} G^{\mathbf{f};\,-1}_{j's'} \tag{39}$$

$$\mathcal{F} = \sum_{s,s'=0}^{m-1} \sum_{k,k'=0}^{n-1} u_{sk} S_{sk;s'k'} u^*_{s'k'} = \sum_{l=1}^{M} \langle \psi_{\mathbf{f}^{(l)}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{x}^{(l)}} \rangle^2 \omega^{(l)} \underset{\mathcal{U}}{\longrightarrow} \max \tag{40}$$

This is *the equation*. The $\langle \psi_{\mathbf{f}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{x}} \rangle$ is interpreted as operator $\|\mathcal{U}\|$ relating the states from two *different* Hilbert space, a type of memoryless quantum channel, a map between two spaces of operators. Every admissible transformation u_{jk} must satisfy Gram matrix invariance condition (38). This condition can be satisfied only for $m \le n$ since $\langle f_j \,|\, f_{j'} \rangle$ has the rank m and the matrix in the right hand side has the rank not greater than n; in case $m > n$ one can consider (41) and obtain (43)

$$x_k = \sum_{j=0}^{m-1} u_{kj} f_j \tag{41}$$

$$\langle x_k \,|\, x_{k'} \rangle = \sum_{j,j'=0}^{m-1} u_{kj} \langle f_j \,|\, f_{j'} \rangle u^*_{k'j'} \qquad\qquad k,k' = 0 \ldots n-1 \tag{42}$$

$$\langle \psi_{\mathbf{g}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{y}} \rangle^2 = \frac{\left| \sum\limits_{j=0}^{m-1} \sum\limits_{k,q=0}^{n-1} y_k G^{\mathbf{x};\,-1}_{kq} u_{qj} g_j \right|^2}{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};\,-1}_{jj'} g_{j'} \sum\limits_{k,k'=0}^{n-1} y_k G^{\mathbf{x};\,-1}_{kk'} y_{k'}} \tag{43}$$

Thus it is sufficient just to swap $\mathbf{x}$ and $\mathbf{f}$ in numerical calculations. When working in orthogonal bases $\delta_{kk'} = \langle x_k \,|\, x_{k'} \rangle$ and $\delta_{jj'} = \langle f_j \,|\, f_{j'} \rangle$ the matrix elements of $S_{sk;s'k'}$ are (33). Also see Appendix B below for possible adjustment of probability normalizing.

Mapping an operator A between $\mathbf{x}$- and $\mathbf{f}$- spaces is the same transformation $A^{\mathbf{f}}_{jj'} = \sum_{k,k'=0}^{n-1} u_{jk} A^{\mathbf{x}}_{kk'} u^*_{j'k'}$ as for Gram matrix (38). The optimization problem (40) has the meaning of finding a quantum channel conveying the highest possible probability from $\mathbf{x}$–space to $\mathbf{f}$–space. A remarkable feature of this problem is that it does not contain any $\langle f_j x_k \rangle$ averages! All the $\mathbf{x} \to \mathbf{f}$ inference (communication between two ends of quantum channel) now contains *only* in operator $\|\mathcal{U}\|$ — a matrix u_{jk} of the dimension $m \times n$ to find from optimization problem (40). This is an important new result. In [13] coverage optimization problem was always formulated with some kind of $\mathbf{x} \to \mathbf{f}$ projection; if a model has $\langle f_j x_k \rangle$ terms – it is of "projective" type such as (14), (31) or (34) above. The (37) and (B1) probabilities do not have $\langle f_j x_k \rangle$ terms; operator $\|\mathcal{U}\|$ directly (36) relates $\mathbf{x}$- and $\mathbf{f}$- spaces subject to (38) scalar product invariance; it is the only link between *IN* and *OUT* spaces. Familiar least squares expansion (7) satisfies the required constraints (38)

$$\langle f_j \,|\, f_{j'} \rangle = \sum_{k,k'=0}^{n-1} \langle f_j x_k \rangle \, G^{\mathbf{x};\,-1}_{kk'} \, \langle x_{k'} f_{j'} \rangle \tag{44}$$

only when $\mathbf{f}$ is a subspace of $\mathbf{x}$; Proof: select some orthogonal bases such as $\delta_{kk'} = \langle x_k \,|\, x_{k'} \rangle$ and $\delta_{jj'} = \langle f_j \,|\, f_{j'} \rangle$, obtain $1 = \sum_{k=0}^{n-1} \langle f_j x_k \rangle^2$, i.e. only when $\mathbf{x} \to \mathbf{f}$ least squares mapping is exact. Note that one can always apply Appendix (A 5) method of singular values adjustment to obtain a partially unitary transform from the least squares or any other mapping that initially does not satisfy the partial unitarity constraints (38).

The optimization considered above has the objective function quadratic on partially unitary operator u_{jk}. There are other objective functions that are quadratic on partially unitary operator u_{jk} hence all the optimization above can be applied to them as well. With (36) definition one can consider it not as probability amplitude mapping $\psi_{\mathbf{x}} \to \psi_{\mathbf{f}}$, but as plain value mapping $\mathbf{x} \to \mathbf{f}$. This is essentially (37) without a denominator. Consider reproducing kernel $\sum_{j,j'=0}^{m-1} f_j G^{\mathbf{f};\,-1}_{jj'} g_{j'}$, it has a maximum at $\mathbf{f} = \mathbf{g}$, assume $\mathbf{g}$ is taken from (36), and sum it squared; obtain

$$\mathcal{F} = \sum_{s,s'=0}^{m-1} \sum_{k,k'=0}^{n-1} u_{sk} S_{sk;s'k'} u_{s'k'} = \sum_{l=1}^{M} \left(\sum_{j,j'=0}^{m-1} f_j^{(l)} G^{\mathbf{f};\,-1}_{jj'} \sum_{k=0}^{n-1} u_{j'k'} x_{k'}^{(l)} \right)^2 \omega^{(l)} \underset{u}{\longrightarrow} \max \quad (45)$$

$$S_{sk;s'k'} = \sum_{j,j'=0}^{m-1} \langle f_j x_k f_{j'} x_{k'} \rangle \, G^{\mathbf{f};\,-1}_{js} G^{\mathbf{f};\,-1}_{j's'} \quad (46)$$

This creates a different version of $S_{jk;j'k'}$, a fourth order moments–type (46) instead of previously used Christoffel functions product tensor $S_{sk;s'k'}$ from (39); an important feature of (46) is that an application of secondary sampling technique is not required for it's calculation.

In this setup the conditions on $\langle f_j f_{j'} \rangle$ and $\langle x_k x_{k'} \rangle$ are put into the constraints (38) and the $\langle f_j x_k f_{j'} x_{k'} \rangle$ is put into the objective function[4]. The mapping with this new $S_{sk;s'k'}$ maps the values, not the probabilities, but the values are considered to belong to some vector space. The squared term in (45) is just a scalar product of two vectors. With (B1) normalizing both vectors be of unit length and the maximal value of the objective function is $\langle 1 \rangle$. In (45) the vectors do not have this normalizing. One can also consider a "partially normalized" tensor, the one with only $K^{(\mathbf{f})}$ term in (33) assuming "average"–type normalizing for x_k is due to (38).

$$S_{sk;s'k'} = \sum_{j,j'=0}^{m-1} \left\langle f_j x_k \,\middle|\, K^{(\mathbf{f})} \,\middle|\, f_{j'} x_{k'} \right\rangle G^{\mathbf{f};\,-1}_{js} G^{\mathbf{f};\,-1}_{j's'} \quad (47)$$

[4] In (45) the scalar product of f_j and $\sum_{k=0}^{n-1} u_{jk} x_k$ is squared and then averaged over the sample. In finding the contributing subspace (20) it is averaged over the sample and then squared. This means the contributing subspace model assumes the factoring $\langle f_j x_k f_{j'} x_{k'} \rangle = \langle f_j x_k \rangle \langle f_{j'} x_{k'} \rangle$. It is similar to Lebesgue quadratures [18], where interchanging of averaging and taking square produces new result.

B. Optimization Problem

The problem of finding the *Knowledge Generalizing Operator* is now reduced to maximizing (40) coverage $\mathcal{F}$ (defined by the tensor $S_{sk;s'k'}$ of diverse possible forms) subject to (38) constraints; the meaning of the constraints is to preserve scalar product (Gram matrix). The result is u_{jk} matrix, $j = 0 \dots m-1; k = 0 \dots n-1$. This operator, given some input state (such as localized state $|\psi_{\mathbf{x}}\rangle$), uniquely (within a phase) finds the function in $\mathbf{f}$-space $|\mathcal{U}|\psi_{\mathbf{x}}\rangle$ (coefficients a_j) that predicts the probability (37) of outcome $|\psi_{\mathbf{f}}\rangle$:

$$P(\mathbf{f})\Big|_{\mathbf{x}} = \langle \psi_{\mathbf{f}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{x}} \rangle^2 = \frac{\left[\sum\limits_{j=0}^{m-1} a_j f_j \right]^2}{\sum\limits_{j,j'=0}^{m-1} f_j G^{\mathbf{f};-1}_{jj'} f_{j'}} \tag{48}$$

the $\mathbf{f}$ is equal to the value of the outcome we are interested to determine the probability of. Given $\mathbf{x}$ the probability of some outcome $\mathbf{f}$ is a squared linear function on f_j multiplied by Christoffel function.

If, however, not the probability but the value of the outcome is required — the easiest method to obtain it is to consider all possible $\mathbf{f}$ to find the maximum[5] of (48):

$$\mathbf{f} : \max_{\mathbf{f}} P(\mathbf{f})\Big|_{\mathbf{x}} = \max_{f_j} \frac{\left[\sum\limits_{j=0}^{m-1} a_j f_j \right]^2}{\sum\limits_{j,j'=0}^{m-1} f_j G^{\mathbf{f};-1}_{jj'} f_{j'}} \tag{49}$$

For 1D class label, where $f_j = f^j$, the problem is reduced to finding the roots of a polynomial. In general case the problem can be considered as generalized eigenvalue problem with the matrices $a_j a_{j'}$ (a dyadic product of two vectors) and $G^{\mathbf{f};-1}_{jj'}$ in the left- and right- hand sides. It has a single non-zero eigenvalue (51) (equals to the maximal probability), corresponding eigenvector (50) gives the most probable outcome $\mathbf{f}$. The maximal probability of the outcome corresponds to the value

$$f_j^{\max P} = \sum_{j'=0}^{m-1} G^{\mathbf{f}}_{jj'} a_{j'} \tag{50}$$

[5] The probability (48) is invariant with respect to $f_j \to const \cdot f_j$ for an arbitrary non–zero *const*. Actual values of f_j are determined using the requirement that the constant has always to be present in $\mathbf{x}$- and $\mathbf{f}$-bases. Since the value of f_j corresponding to this specific index $j : f_j = const$ is always known (a constant), the actual values of all f_j are obtained as f_j/C where $C = f_{j:f_j=const}$; see `com/polytechnik/kgo/KGOSolutionVectorXVectorF.java:evaluateAt(double[]xorig)`.

$$P(\mathbf{f}^{\max P})\Big|_{\mathbf{x}} = \sum_{j,j'=0}^{m-1} a_j G^{\mathbf{f}}_{jj'} a_{j'} \tag{51}$$

The $P(\mathbf{f}^{\max P})\Big|_{\mathbf{x}}$ is a certainty of the outcome, the maximal possible value of (48), a $[0:1]$ bounded function. A difficulty with this approach is that if $\mathbf{f}$ is constructed from a scalar function, such as $f_j = f^j$, this relation may not hold exactly in the result.

Obtained probability formula (48) is of very general form: a linear function on f_j squared divided by a quadratic form on f_j. It can be obtained from many different considerations, the difference between models is in coefficients a_j. The simplest solution of this type is a "direct projection" solution of [13], where we take least squares expansion of $|f_j\rangle$ in $|x_k\rangle$ (7) and substitute obtained $\mathbf{f}_{LS}(\mathbf{x})$ as the localization point in (13) to obtain $\left|\psi_{\mathbf{f}_{LS}(\mathbf{x})}\right\rangle$. This is an example to obtain the probability of (48) form without quantum channel used. It should be also noted that squared linear function in numerator arises only for pure states. When working with states in the form of density matrix — the probability takes the form of two quadratic forms ratio.

The problem has remarkable invariance features. Consider (17) mappings $\psi_{\mathbf{x}^{(l)}} \to \psi_{\mathbf{f}^{(l)}}$, $l = 1 \dots M$ of n-dimensional vector $\psi_{\mathbf{x}^{(l)}}$ to m-dimensional vector $|\psi_{\mathbf{f}^{(l)}}\rangle$. The vectors are projected to each other with operator $\|\mathcal{U}\|$, projection absolute value is then squared and all summed (40) over the entire sample. The major difference from any observable value–mapping technique is that if we multiply all $\psi_{\mathbf{x}^{(l)}}$ and $\psi_{\mathbf{f}^{(l)}}$ by random phases $\exp(i\varphi^{(l)})$ the result will be identical! This is the same as in quantum mechanics: a wavefunction is defined within a phase, wavefunction absolute value squared defines the probability, but Schrödinger equation is written for the wavefunction. Similarly, the knowledge generalizing operator $\|\mathcal{U}\|$ is defined (for complex matrix) within a phase, for real matrix – within a ± 1 factor, but the probability (48) and coverage (40) are equal to operator $\|\mathcal{U}\|$ projections *squared*; individual $\psi_{\mathbf{x}^{(l)}}$ and $\psi_{\mathbf{f}^{(l)}}$ may have arbitrary phases.

Optimization problem (40) subject to (38) constraints is a variant of QCQP problem. It has the form: to find an operator $\|\mathcal{U}\|$ optimally transforming an *IN* state $|\psi_{\mathbf{x}}\rangle$ into an *OUT* state $|\psi_{\mathbf{f}}\rangle$ on (17) data, i.e. the ideology is similar to the one of S-Matrix. Currently we can solve this optimization problem only numerically. The problem is similar to an eigenvalue problem, see (A7). This is a new algebraic problem:

$$S\mathcal{U} = \lambda\mathcal{U} \tag{52}$$

where S is a Hermitian tensor, "eigenvector" $\|\mathcal{U}\|$ is a partially unitary $m \times n$ matrix, and "eigenvalues" λ is a Hermitian $m \times m$ matrix; functional (40) extremal value is equal to λ spur (the sum of diagonal elements (A9)). The mathematical structure of this eigenvalue–like problem, an "eigenoperator" problem, requires a separate study and we hope to obtain important new results soon. Currently — we have a fast, stable to degeneracy iteration algorithm to find a solution numerically, see Appendix A below.

Considered model assumes the dynamics is determined by a single unitary operator, possibly partially unitary. For a $\mathbf{x}$-localized pure state $\|\rho_{\mathbf{x}}\| = |\psi_{\mathbf{x}}\rangle \langle\psi_{\mathbf{x}}|$ a unitary operator $\|\mathcal{U}\|$ transforms the density matrix to

$$\|\widetilde{\rho}_{\mathbf{x}}\| = \|\mathcal{U}|\rho_{\mathbf{x}}|\mathcal{U}^{\dagger}\| \tag{53}$$

Whereas in quantum mechanics evolution operator $\|\mathcal{U}\|$ corresponds to the Hamiltonian of the system: $\mathcal{U} = \exp\left[-i\frac{t}{\hbar}H\right]$, in data analysis knowledge generalizing operator $\|\mathcal{U}\|$ is obtained from optimization problem (40) subject to (38) constraint. Quantum evolution of (53) form always transforms a pure state $\|\rho\| = |\psi\rangle \langle\psi|$ to the pure state $\|\widetilde{\rho}\| = |\mathcal{U}|\psi\rangle \left\langle\psi|\mathcal{U}^{\dagger}\right|$, and a mixed state $\|\rho\|$ to the mixed state $\|\widetilde{\rho}\|$. In data analysis there is a common situation when a pure state is transformed into a mixed state, Markov chain is an example. In this case a more general form of quantum evolution is required[19]:

$$\widetilde{\rho} = \sum_{s} B_s \rho B_s^{\dagger} \tag{54}$$

with Kraus operators B_s satisfying[6]

$$\sum_{s} B_s B_s^{\dagger} = \mathbb{1} \tag{55}$$

The data we use in this paper is of pure state to pure state mapping (17). For other type of input data unitary evolution (53) should be replaced by a more general form (54); one may think about it as a quantum system evolving with several Hamiltonians at once $B_s = \exp\left[-i\frac{t}{\hbar}H_s\right]$, not as about a system evolving with the Hamiltonian $H = \sum\limits_{s} H_s$. The approach is directly generalizable to e.g. probability distribution to probability distribution

[6] Similarly to (38) Kraus operators B_s can also be considered in a "partially unitary"–style with $b_{s;jk}$ matrices of the dimension $m \times n$ satisfying $\langle f_j \,|\, f_{j'}\rangle = \sum\limits_{s} \sum\limits_{k,k'=0}^{n-1} b_{s;jk} \langle x_k \,|\, x_{k'}\rangle b^{*}_{s;j'k'}$. The optimization problem (40) then becomes $\sum\limits_{s} \sum\limits_{l=1}^{M} \langle \psi_{\mathbf{f}^{(l)}} \,|\, B_s \,|\, \psi_{\mathbf{x}^{(l)}} \rangle^2 \omega^{(l)} \underset{B_s}{\longrightarrow} \max$.

mapping: in this case the observations are not localized states mapping $\psi_{\mathbf{x}^{(l)}} \to \psi_{\mathbf{f}^{(l)}}$, but corresponding density matrices mapping $\|\rho^{(l)}_{\mathbf{x}}\| \to \|\rho^{(l)}_{\mathbf{f}}\|$.

Initial $\mathbf{x}^{(l)} \to \mathbf{f}^{(l)}$ input data (1) was converted to pure state to pure state mapping $\psi_{\mathbf{x}^{(l)}} \to \psi_{\mathbf{f}^{(l)}}$ (17) to formulate optimization problem (40) subject to (38) constraints. It is essential from methodical point of view to discuss what input moments are required for this problem (to obtain the tensor $S_{jk;j'k'}$ (39)) and compare with other models. This is summarized in the table:

Model	Tensors Required to Calculate
Least Squares (7)	$\langle x_k x_{k'} \rangle$, $\langle x_k f_j \rangle$
Radon-Nikodym (11)	$\langle x_k x_{k'} \rangle$, $\langle x_k x_{k'} f_j \rangle$
$\mathbf{x}$ — $\mathbf{f}$ Christoffel function (12)	$\langle x_k x_{k'} \rangle$, $\langle f_j f_{j'} \rangle$, $\langle x_k f_j \rangle$
Pure Joint Distribution (14)	$\langle x_k x_{k'} \rangle$, $\langle f_j f_{j'} \rangle$, $\langle x_k f_j \rangle$, $\left\langle x_k f_j \,\middle|\, K^{(\mathbf{x})} K^{(\mathbf{f})} \,\middle|\, x_{k'} f_{j'} \right\rangle$
Partial Unitarity (KGO) (37)	$\langle x_k x_{k'} \rangle$, $\langle f_j f_{j'} \rangle$, $\left\langle x_k f_j \,\middle|\, K^{(\mathbf{x})} K^{(\mathbf{f})} \,\middle|\, x_{k'} f_{j'} \right\rangle$
Partial Unitarity (KGO) $K^{(\mathbf{f})}$ (47)	$\langle x_k x_{k'} \rangle$, $\langle f_j f_{j'} \rangle$, $\left\langle x_k f_j \,\middle|\, K^{(\mathbf{f})} \,\middle|\, x_{k'} f_{j'} \right\rangle$
Partial Unitarity (KGO) adj. (B1)	Beyond moments, no $\langle x_k f_j \rangle$ used.

The major difference — Knowledge Generalizing Operator (KGO) is the only model that does not require "projective" moments $\langle x_k f_j \rangle$; it requires Gram matrices (4) and (5) of *IN* and *OUT* bases and the moments of the Christoffel functions product (33). These moments can be obtained with an application of secondary sampling technique[15]: Gram matrices are built first; then, for every observation $l = 1 \ldots M$, Christoffel function is calculated and used as it were plain observed at observation l. These moments[7] of two Christoffel functions product are the input used to formulate the problem (40). For a Christoffel function in some multi-dimensional vector space $\boldsymbol{r}$ (e.g. $\mathbf{x}$ (10) or $\mathbf{f}$ (23)) with $\langle \cdot \rangle$ inner product and non-degenerated Gram matrix $G_{jj'} = \langle r_j \,|\, r_{j'} \rangle$ there is a $1/r^2$ asymptotic:

$$K(\boldsymbol{r}) = \frac{1}{\langle \boldsymbol{r} \,|\, G^{-1} \,|\, \boldsymbol{r} \rangle} = \frac{1}{\sum\limits_{j,j'} r_j G^{-1}_{jj'} r_{j'}} \tag{56}$$

$$K(\boldsymbol{r}) \sim 1/r^2 \qquad \text{for} \quad r \to \infty \tag{57}$$

The same $1/r^2$ long–range interaction presents in Coulomb's law or Newton's law of gravitation. With (57) asymptotic the Christoffel function can be viewed as a form of "long–range $1/r^2$

[7] The (B1) KGO model goes "beyond moments". Even with secondary sampling it is impossible to build from moments the (18) target functional with the probability (B1). Moreover, this problem is not a QCQP problem.

interaction", an anisotropic gravity–like law of data analysis. These non-local features, along with eigenproblem (A6) of the dimension Dn and SVD (A10) (or Gram matrix eigenproblem (A26)) that are required on every iteration, substantially slow down the algorithm when implemented without optimization. At this point, however, the goal is not to build a fast algorithm, but to understand all the benefits and drawbacks of ML knowledge representation in the form of partially unitary operator. Let us do a demonstration.

IV. A DEMONSTRATION OF KNOWLEDGE GENERALIZING OPERATOR APPLICATION

In this section we are going to present several demonstrations of $\mathbf{f}(\mathbf{x})$ calculation using (49). The $\mathbf{f}$ and $\mathbf{x}$ are treated as linear spaces, a basis for wavefunction, with partially unitary operator u_{jk} mapping (36). The result is invariant relatively $\mathbf{f} \to C \cdot \mathbf{f}$. To obtain actual value of $\mathbf{f}$ — it should be normalized to const. The constant has always to be present in both $\mathbf{f}$– and $\mathbf{x}$– bases. Thus

$$\mathbf{f}^{actual} = \frac{\mathbf{f}}{f_{j:f_j=const}} \tag{58}$$

In this equation the numerator is a linear function on $\mathbf{x}$ (36) and the denominator, the const–component of $\mathbf{f}$, possibly also is a linear function on $\mathbf{x}$. Thus the value obtained from partially unitary operator mapping is a ratio of two linear functions on $\mathbf{x}$. The least squares (7) always maps a constant to a constant, thus when u_{jk} is a least squares mapping the denominator in (58) is always a constant. In Radon–Nikodym mapping (11) the numerator is a quadratic form on $\mathbf{x}$ and the denominator is a positive quadratic form on $\mathbf{x}$; the denominator is never zero. In (58) the numerator and the denominator are both linear functions on $\mathbf{x}$ of most general form. The divergences coming from denominator's zeroes are important new features of the approach. In least squares – these zeroes are on the infinity. Denominator's zeroes may come either from deep internal properties of the model or from sub-optimal solution of the optimization problem (or badly chosen objective function).

The objective function is determined by the tensor $S_{jk;j'k'}$. Whereas properly normalized probability (B1) lead to a non–QCQP problem, the original Christoffel (39), the adjusted number of degrees of freedom Christoffel (B7), $\mathbf{f}$–Christoffel (47), and plain $\langle f_j x_k f_{j'} x_{k'} \rangle$ (46) have $S_{jk;j'k'}$ tensor readily available and the optimization problem (A1) with the constraints

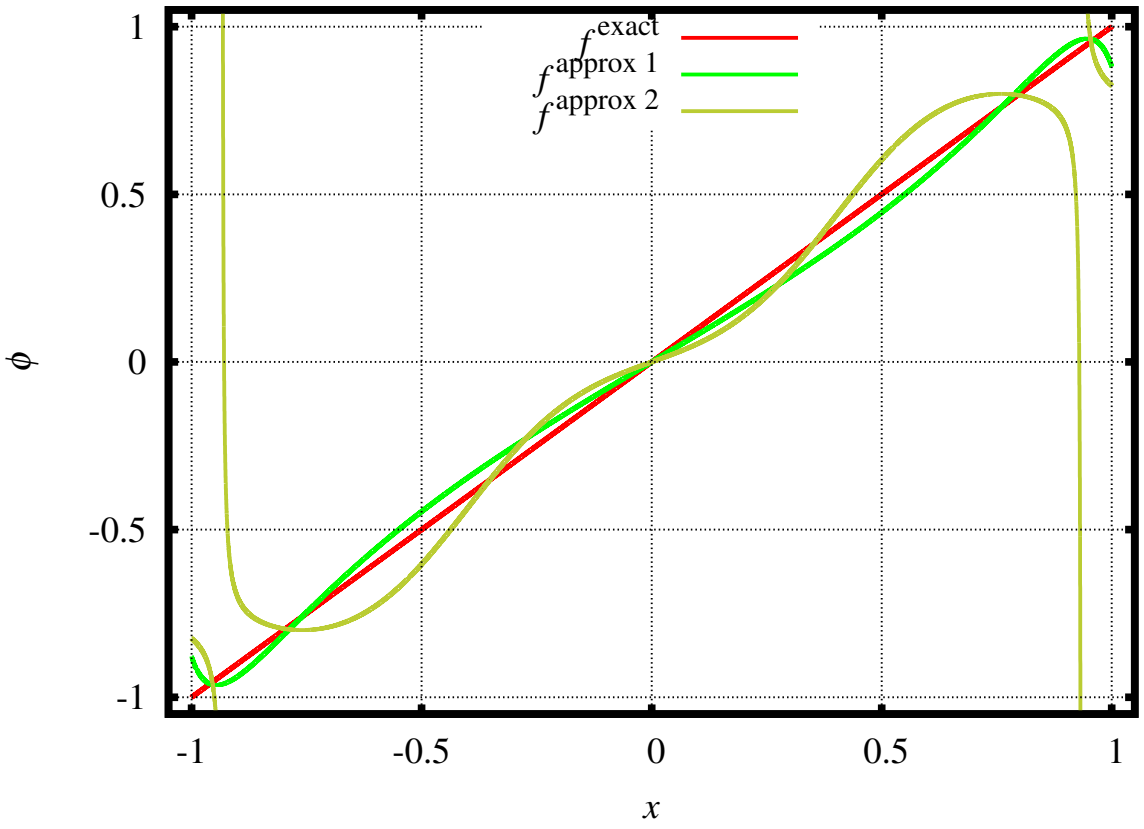

FIG. 3. For a data with known exact $f = x$ solution, when numerical method does not find it – it is possible to have zeroes in (58) and corresponding poles in the behavior.

(A2) can be formulated and solved numerically.

Among available $S_{jk;j'k'}$ versions the **f**–Christoffel (47) has the most "usual" properties. For example the (39) or (B7), when run with a data of exact $\mathbf{x} \to \mathbf{f}$ homomorphism can possibly give a higher $\mathcal{F}$ on non–exact mapping due to unusual localized states normalizing. For this reason all the demonstrations below will be performed with **f**–Christoffel $S_{jk;j'k'}$ (47).

Consider a trivial mapping with the measure $\langle g \rangle = \int_{-1}^{1} g(x)dx$ and the basis $\mathbf{x}$ constructed from 1D variable $x \in [-1:1]$ as $x_k = x^k$, $k = 0 \ldots 6; n = 7$, and $f_j = x_j$ for $j = 0 \ldots 4; m = 5$. The solution is trivial: take first m components of x_k and regularize; then use them for both: $\mathbf{x}$ and $\mathbf{f}$. However, when the numerical algorithm cannot find this exact solution we can observe a deviation from exact match. In Fig. 3 the exact solution along with two approximate solutions of different quality are presented. A not very accurate approximate numerical solution may give poles corresponding to the zeroes in (58) denominator (clearly

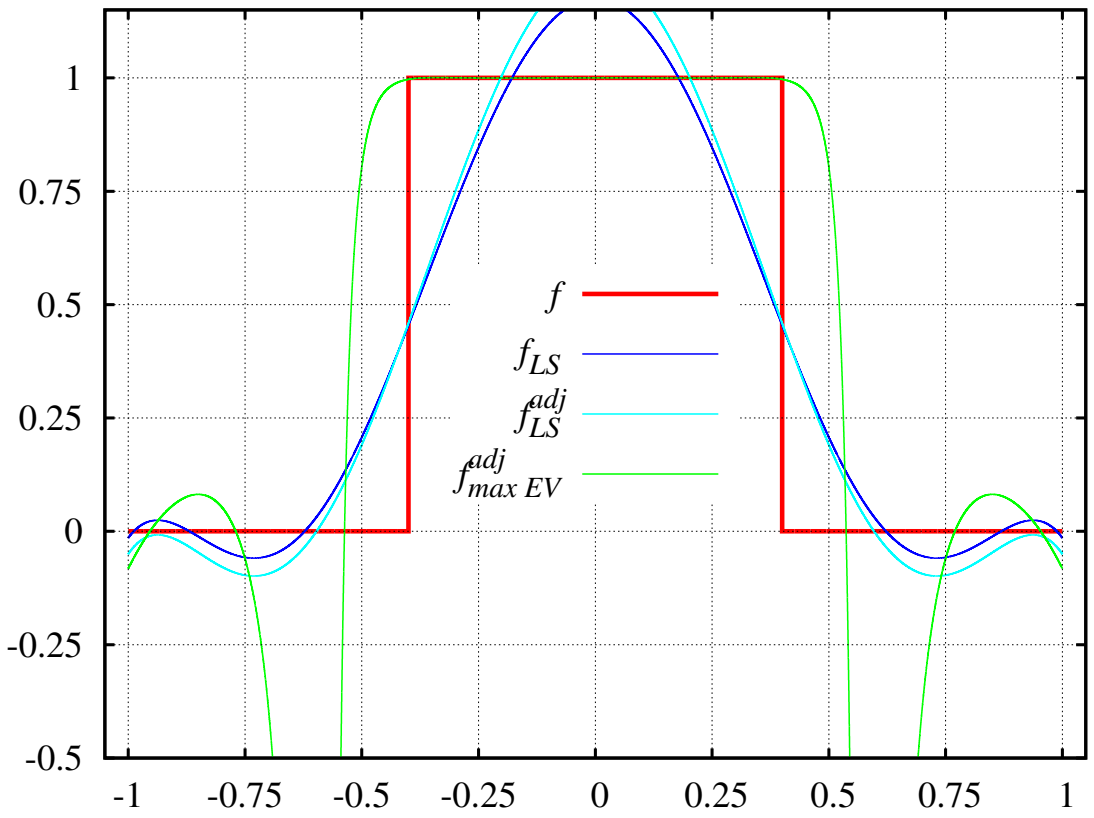

FIG. 4. A square wave step function (the same as in Fig. 2), with least squares (blue), least squares with (A11) SVD adjustment (light blue), and maximal eigenvalue with (A11) SVD adjustment (green).

observed for $f^{\text{approx}\,2}$ near interval edge).

In Fig. 4 a square wave step function (the same as in Fig. 2) is presented with the same measure and basis; $n = 7$. The f takes only two values since the only available m is $m = 2$. The exact solution was difficult to obtain numerically as the problem is substantially degenerated. We present three approximate solutions. The blue line is regular least squares (7). Light blue is the same least squares mapping (7) adjusted with (A11) to partial unitarity. Green — maximal eigenvalue (A3) solution adjusted to partial unitarity with (A11). One can see that partial unitarity adjustment makes little changes to least squares solution. For adjusted maximal eigenvalue solution the (58) denominator poles are close to the support of $\mathbf{x}$, this creates two artifacts in $\mathbf{f}$. Note almost exact $f = 1$ matching in the center.

Consider a 2D example. Let us take an image and consider it as a two–dimensional basis

mapping a pixel coordinate (x, y) to gray intensity f.

$$(x, y)^{(l)} \to f^{(l)} \qquad \text{weight } \omega^{(l)} = 1 \tag{59}$$

$$\mathbf{k} = (k_x, k_y) \tag{60}$$

$$x_{\mathbf{k}} = x^{k_x} y^{k_y} \qquad 0 \le k_x \le n_x - 1; \quad 0 \le k_y \le n_y - 1 \tag{61}$$

$$f_j = f^j \qquad j = 0 \ldots m - 1 \tag{62}$$

This forms a (1) basis[8] of $n = n_x n_y$ and m dimensions. Let us construct an operator u_{jk} mapping $\mathbf{x} \to \mathbf{f}$. A simple example is least squares (7), it creates a familiar image expansion similar to Fourier series. However, we are interested in operators u_{jk} satisfying all partial unitarity constraints (38). A simple variant of constraint–satisfying operator can be obtained from any u_{jk} operator applying Appendix A 5 algorithm. In Fig. 5 (top row) we present original image, least squares expansion and constraint–adjusted least squares for $n_x = n_y = 25$, $m = 5$. The constraint–adjusted least squares is very similar to the original least squares. The least squares operator maps pixel coordinates to gray intensity, not the localized states wavefunction. When an operator is optimized to map the wavefunctions this may cause poles in values, the zeroes of (58) denominator. It is trivially to construct a partially unitary operator u_{jk} preserving the constant: construct a partially unitary operator mapping $\mathbf{x}$-space *without const* to $\mathbf{f}$-space *without const* $\mathbf{x}\backslash C \to \mathbf{f}\backslash C$, then do a direct sum with $C \to C$ mapping. We do not perform such a transform specifically to observe the poles in (58). We present three pictures, corresponding to u_{jk} operators differently optimizing (40) with $\mathbf{f}$–Christoffel tensor (47). In Fig. 5 (middle row) we present the results corresponding to these three u_{jk}: optimizing (40) with simplified constraints (A5), the same one adjusted with (A11) to partial unitarity, and optimization result with Section A 4 algorithm (overall the best optimization algorithm we have so far). Left two pictures in the middle row — a simple solutions (based on trivial approach of maximal eigenvalue state), they have noticeable $1/n_{\{x,y\}}$ scale artifacts. The last one is very close to the global maximum of (40) and "mixes" the modes much stronger . The poles of (58) separate the regions and the structure of these "separators" can be a subject of our future research.

The developed approach works with probabilities, not with the values. For this reason it is of interest to present the probability (48) at given known outcome $\mathbf{f} = \mathbf{f}^{(l)}$. The result is

[8] For numerical stability it is better to use argument–scaled Chebyshev polynomials rather than monomials powers $x^{k_x} y^{k_y}$ and f^j.

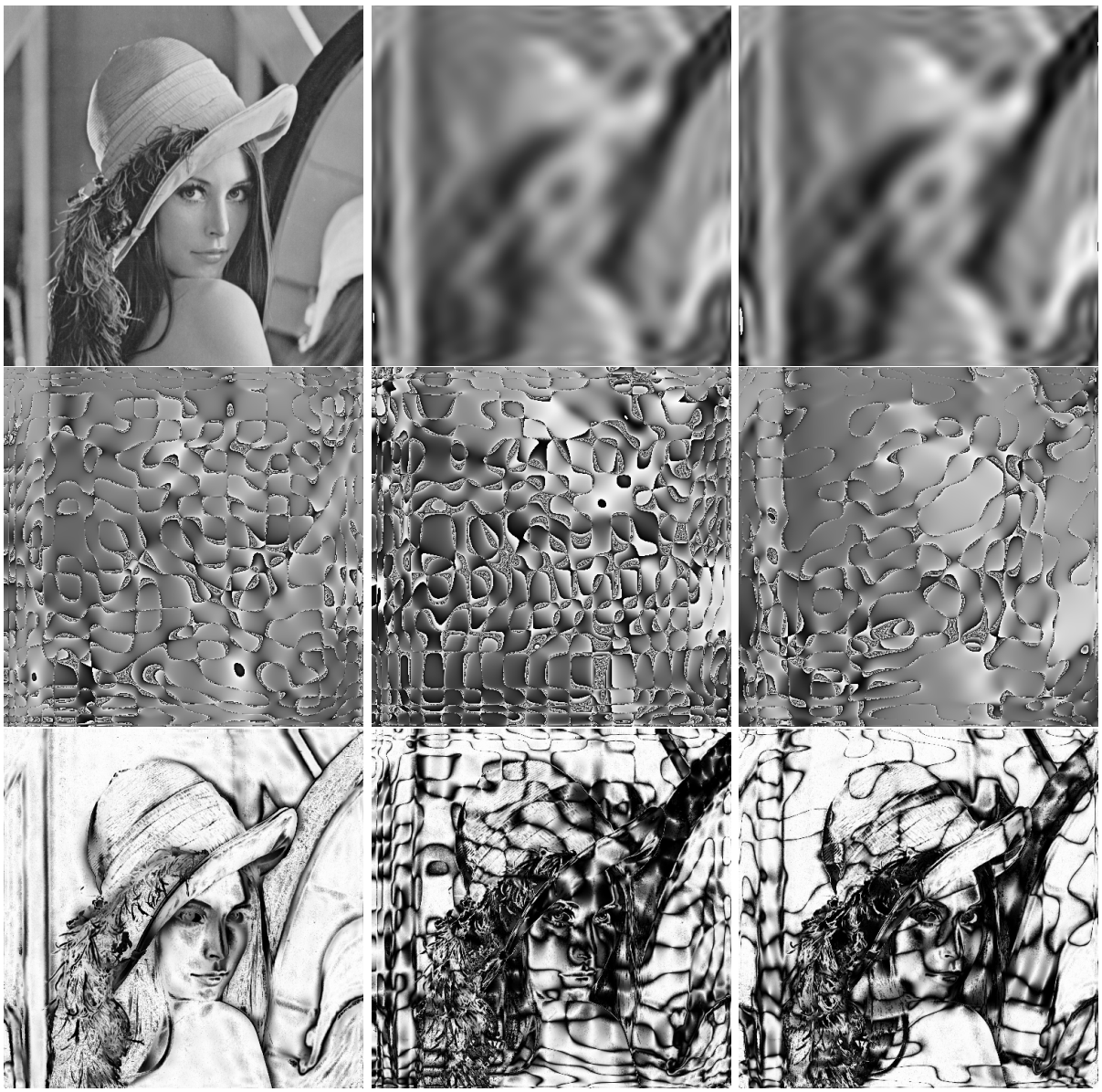

FIG. 5. A demonstration of image interpolation with $n_x = n_y = 25$, $m = 5$. Top row: original image, least squares (7) interpolated, and the same least squares adjusted with (A11) to partial unitarity constraints. Middle row: optimization (40) with simplified constraints (A5) (the state of maximal eigenvalue), the same one adjusted with (A11) to partial unitarity constraints, and optimization result with Section A 4 algorithm. Bottom row: The probability (48) is calculated at actual $\mathbf{f}$, white $P = 1$, black $P = 0$. It is calculated for: least squares (7) ("direct projection" model of [13]), the state of maximal eigenvalue (unadjusted), and Section A 4 algorithm.

presented in Fig. 5, the bottom row. The probability is scaled as white $P = 1$, black $P = 0$. It is presented in the bottom row for three algorithms: least squares (7), the state of maximal eigenvalue (unadjusted), and Section A 4 algorithm.

The method to overcome noticeable $1/n_{\{x,y\}}$ artifacts in Fig. 5 is to use properly normalized

states (B1). In most general form it can be considered as an *unconstrained* optimization problem. Given sampled data (1) find a linear transform u_{jk} (63), a general form matrix of the dimension $m \times n$, maximizing (64)

$$f_j = \sum_{k=0}^{n-1} u_{jk} x_k \qquad j = 0 \dots m-1 \tag{63}$$

$$\mathcal{F} = \sum_{l=1}^{M} \left\langle \psi_{\mathbf{f}^{(l)}} \,\middle|\, \psi_{u(\mathbf{x}^{(l)})} \right\rangle^2 \omega^{(l)} \underset{u}{\longrightarrow} \max \tag{64}$$

Here the $|\psi_{\mathbf{g}}\rangle$ is the state (13) localized at $\mathbf{f} = \mathbf{g}$, and $\left|\psi_{u(\mathbf{x})}\right\rangle$ is also $\mathbf{f}$–localized state (13) with the localization point $\mathbf{g}$ determined by (63) linear mapping. When expanded $\left\langle \psi_{\mathbf{g}} \,\middle|\, \psi_{u(\mathbf{y})} \right\rangle^2$ is (B1). The objective function (64) is the total probability transferred from $\mathbf{x}$–space to $\mathbf{f}$–space; this is an unconstrained problem. In this most general form the problem is not a QCQP problem and it is difficult to solve numerically; the difficulty is that with $\left|\psi_{u(\mathbf{x})}\right\rangle$ state the operator u_{jk} enters (through localization point $\mathbf{g}$) both the numerator and the denominator of (13), what makes the optimization problem (64) not a QCQP problem. The problem can be substantially simplified when the u_{jk} mapping is considered to be a partially unitary transform (38) to obtain a QCQP problem. The problem can be further approximated by splitting the solution into two steps: selecting the contributing subspace ϕ_k of the dimension m, then constructing a unitary (not partially unitary) mapping from the contributing subspace to f_j. A simple projective approach is presented above in Eq. (26) or, more generally, in the Appendix C below. A simple solution of this type is the "direct projection" model of [13] where the localization point is determined from plain least squares (7) to obtain the state $\left|\psi_{\mathbf{f}_{LS}(\mathbf{x})}\right\rangle$. The probability $\left\langle \psi_{\mathbf{f}^{(l)}} \,\middle|\, \psi_{\mathbf{f}_{LS}(\mathbf{x}^{(l)})} \right\rangle^2$ of the "direct projection" model is presented in Fig. 5 (leftmost in the bottom row).

These demonstrations make us to conclude that partial unitary mapping is a rich form of knowledge representation with a high generalizing power, however a more study is required.

V. CONCLUSION

The developed knowledge generalizing operator concept is similar to the S-Matrix approach since it is an operator optimally transforming an *IN* state $|\psi_{\mathbf{x}}\rangle$ into an *OUT* state $|\psi_{\mathbf{f}}\rangle$. As any wavefunction in ML is known within an arbitrary phase the equation for the operator must include only observable values. The problem we consider is to recover $\|\mathcal{U}\|$ from all it's

projections *squared*, from the probabilities (37). The condition of operator's optimality is (40) coverage maximization on (17) data; it is a new kind of algebraic problem (52) — the equation to determine the $\|\mathcal{U}\|$. The situation is the same as with the Schrödinger equation: the equation is written for ψ, but only ψ^2 is observable. This is the difference between our and all other ML knowledge representation techniques where knowledge representation characteristics are observable values. If a model relates an initial observable and the final observable then it is a "joint distribution model", it cannot predict something that has not been already observed in the training data. Knowledge generalizing operator relates the amplitude of the initial state to the amplitude of the final state. This is the very feature that creates generalization. The same is in quantum mechanics: ψ^2 vs ψ; whereas a mapping of ψ^2 is meaningless, the mapping of ψ determines the dynamics of a system.

Considered maximization problem (40) is a simple example of knowledge generalizing operator technique: for observations $l = 1 \ldots M$ convert $\mathbf{x}^{(l)} \to \mathbf{f}^{(l)}$ to $\psi_{\mathbf{x}^{(l)}} \to \psi_{\mathbf{f}^{(l)}}$, then reconstruct $\|\mathcal{U}\|$ from it's projections squared $\langle \psi_{\mathbf{f}^{(l)}} | \mathcal{U} | \psi_{\mathbf{x}^{(l)}} \rangle^2$. The problem can be generalized by considering, instead of l, $\mathbf{x}$, $\mathbf{f}$, and $\langle\cdot\rangle$, the structures generalizing the concepts of set, vector, and measure. In the most general form it can be formulated as: for $\psi \in S_x$ and $\varphi \in S_f$ recover partially unitary operator $\|\mathcal{U}\|$ from it's projections squared $\sum\limits_{l \in M} \omega^{(l)} \left\langle \varphi^{(l)} \middle| \mathcal{U} \middle| \psi^{(l)} \right\rangle^2 \underset{\mathcal{U}}{\longrightarrow} \max$. The problem can be further generalized by considering mixed states $\|\rho\| \in S_x$ and $\|\varrho\| \in S_f$ and recovering Kraus operators B_s (54) from projections squared: $\sum\limits_{l \in M} \omega^{(l)} \sum\limits_s \mathrm{Spur} \|\varrho^{(l)} | B_s | \rho^{(l)} | B_s^\dagger \| \underset{B_s}{\longrightarrow} \max$.

There is another interesting twist to the considered problem of finding a partially unitary matrix u_{jk} of the dimension $\dim(OUT) \times \dim(IN)$ mapping operators from $|IN\rangle$ to $|OUT\rangle$. Consider the problem: for $\dim(OUT) < \dim(IN)$ select $\dim(OUT)$ input attributes out of all $\dim(IN)$ available that maximize some correctness condition which is a function of all selected attributes. For all interesting correctness conditions this problem is typically a one of NP–complete type. There is a single correctness function (least squares) that can be trivially solved. Maximization of total matched probability (18) among all partially unitary operators u_{jk} also selects $\dim(OUT)$ inputs from all $\dim(IN)$ available. This is a new algebraic problem (52). Found mapping u_{jk} can be viewed as a solution to attributes selection problem with correctness conditions somewhere "in between" least squares and NP–complete, for example there is a simple subspace selection approach (C6) — then a problem of unitary mapping (not partially unitary) can be directly solved.

Appendix A: A Numerical Solution to Find the Knowledge Generalizing Operator

The problem we consider is a QCQP problem to maximize (A1) subject to (A2) constraint.

$$\mathcal{F} = \sum_{l=1}^{M} \langle \psi_{\mathbf{f}^{(l)}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{x}^{(l)}} \rangle^2 \, \omega^{(l)} = \sum_{j,j'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{jk} S_{jk;j'k'} u_{j'k'} \underset{u}{\longrightarrow} \max \tag{A1}$$

$$\langle f_j \,|\, f_{j'} \rangle = \sum_{k,k'=0}^{n-1} u_{jk} \, \langle x_k \,|\, x_{k'} \rangle \, u_{j'k'} \qquad\qquad j, j' = 0 \ldots D-1 \tag{A2}$$

Without loss of generality we put $\delta_{kk'} = \langle x_k \,|\, x_{k'} \rangle$ and $\delta_{jj'} = \langle f_j \,|\, f_{j'} \rangle$ as we can always choose an orthogonal basis by applying, for example, an orthogonalization of Gram–Schmidt type. Contrary to other methods (e.g. regular principal components) the result obtained with knowledge generalizing operator is invariant with respect to (8) transform of input data, thus it does not depend on initial regularization. The problem becomes:

$$\mathcal{F} = \sum_{j,j'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{jk} S_{jk;j'k'} u_{j'k'} \underset{u}{\longrightarrow} \max \tag{A3}$$

$$\delta_{jj'} = \sum_{k=0}^{n-1} u_{jk} u_{j'k} \qquad\qquad j, j' = 0 \ldots D-1 \tag{A4}$$

Consider the squared Frobenius norm of matrix u_{jk} to be a "simplified constraint":

$$\sum_{j=0}^{D-1} \sum_{k=0}^{n-1} u_{jk}^2 = D \tag{A5}$$

This is a "partial" constraint (it is the sum of all (A4) diagonal elements). For this "partial" constraint optimization problem (A3) can be readily converted to an eigenvalue problem that can be directly solved. The main idea is to adjust this "preliminary" solution to satisfy the full set of (A4) constraints and then calculate new values of Lagrange multipliers. Performing several iterations the process possibly converge to (A3) maximum with all the required constraints (A4) satisfied. In [13] a similar technique has been tried for a unitary operator (19). The (A4) corresponds to partially orthogonal operator (partially unitary real matrix): $D \leq n$.

Consider Lagrange multipliers $\lambda_{jj'}$, a matrix of $D \times D$ dimension, to approach optimization problem (A3) with the constraints (A4)

$$\sum_{j,j'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{jk} S_{jk;j'k'} u_{j'k'}^* + \sum_{j,j'=0}^{D-1} \lambda_{jj'} \left[\delta_{jj'} - \sum_{k'=0}^{n-1} u_{jk'} u_{j'k'}^* \right] \underset{u}{\longrightarrow} \max \tag{A6}$$

Despite the matrix u_{jk} being real we write it in a "complex" form to variate separately over u_{jk} and u^*_{jk}. The tensor $S_{jk;j'k'} = S^*_{j'k';jk}$ is Hermitian. The variations

$$0 = \sum_{j'=0}^{D-1} \sum_{k'=0}^{n-1} u_{j'k'} S_{j'k';jk} - \sum_{j'=0}^{D-1} \lambda_{j'j} u_{j'k} \tag{A7a}$$

$$0 = \sum_{j'=0}^{D-1} \sum_{k'=0}^{n-1} S_{jk;j'k'} u^*_{j'k'} - \sum_{j'=0}^{D-1} \lambda_{jj'} u^*_{j'k} \tag{A7b}$$

are consistent only when $\lambda_{jj'}$ is a Hermitian matrix

$$\lambda_{jj'} = \lambda^*_{j'j} \tag{A8}$$

From (A7) it follows that the functional (A3) extremal value is equal to the spur of Lagrange multipliers matrix $\lambda_{jj'}$:

$$\max \mathcal{F} = \sum_{j=0}^{D-1} \lambda_{jj} \tag{A9}$$

An iterative algorithm finding the maximum of (A3) subject to (A4) constraints is:

1. Take initial λ_{ij} and solve optimization problem (A6) with respect to u_{jk} subject to partial constraint (A5). Solution method — an eigenvalue problem of Dn dimension in a vector space formed by writing all u_{jk} matrix elements in a vector, row by row. The result: $p = 0 \ldots Dn-1$ eigenvalues $\mathcal{F}^{[p]}$ and corresponding matrices $u^{[p]}_{jk}$ reconstructed back from the eigenvectors, row by row.

2. To select the u_{jk} among all Dn eigenstates one need to try a number of them, selecting the ones providing a large value of the original functional. Taking only the state of the maximal eigenvalue typically gives a local maximum. Chosen u_{jk} is not partially unitary as the constraint (A5) is a subset of the full ones (A4). Expand u_{jk} in SVD:

$$u_{jk} = \sum_{j'=0}^{D-1} \sum_{k'=0}^{n-1} U_{jj'} \Sigma_{j'k'} V^\dagger_{k'k} \tag{A10}$$

and adjust all SVD numbers to ± 1. The $\Sigma_{jk} = \delta_{jk}$ is typically the best option as this is the minimal change (initial Σ_{jj} are positive). Obtained

$$\widetilde{u}_{jk} = \sum_{s=0}^{\min(D,n)-1} U_{js} V^\dagger_{sk} \tag{A11}$$

is a partially unitary matrix satisfying all the constraints (A4). This $\widetilde{u}_{jk}$ becomes the next iteration u_{jk} of the solution. Because of $u_{jk} \to \widetilde{u}_{jk}$ adjustment the value of $\mathcal{F}$ becomes less optimal. There are other methods to adjust the u_{jk} to satisfy the full set of (A4) constraints, for example an eigenvector expansion of the matrix $\sum_{k=0}^{n-1} u_{jk} u_{j'k}$ followed by eigenvalues adjustment[17], Gram–Schmidt orthogonalization, etc. However, the SVD expansion (A10) is special, see (A52) below.

3. Put this new u_{jk} to (A7a), then multiply it by u^*_{ik} and sum over $k = 0 \ldots n-1$. As the u_{jk} is partially unitary (A4) obtain new values for Lagrange multipliers $\widetilde{\lambda}_{ij}$ and take it's Hermitian part[9]:

$$\widetilde{\lambda}_{ij} = \sum_{j'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{j'k'} S_{j'k';jk} u^*_{ik} \tag{A12}$$

$$\lambda_{ij} = \frac{1}{2} \left[\widetilde{\lambda}_{ij} + \widetilde{\lambda}^*_{ji} \right] \qquad\qquad i,j = 0 \ldots D-1 \tag{A13}$$

This λ_{ij} is the next iteration of Lagrange multipliers. As iterations proceed – the $\widetilde{\lambda}_{ij}$ is expected to converge to a Hermitian matrix by itself, without (A13) required. For original (not yet full–constraint adjusted) u_{jk}, which is an eigenvector of $S_{j'k';jk}$, the $\widetilde{\lambda}_{ij}$ is Hermitian. The anti–Hermitian part of $\widetilde{\lambda}_{ij}$ cancels in the quadratic form (A6). One can possibly obtain a Hermitian λ_{ij} right away with multiplication of (A7a) by itself (instead of u_{jk} for (A12)); the Hermitian λ_{ij} is then obtained from λ^2_{ij} as all the eigenvalues of λ_{ij} are all positive; the result is very similar to (A13), a drawback for this new λ_{ij} — the (A9) now holds only approximately for current iteration of u_{jk}, see `com/polytechnik/kgo/KGOIterationalLambda2.java`.

4. Put this new λ_{ij} to (A6) and repeat iteration process until converged. On the first iteration take initial values of Lagrange multipliers $\lambda_{ij} = 0$.

For a simpler scalar QCQP optimization problem of [15], "*Appendix F: Directional Information:* $I \underset{\psi}{\longrightarrow} \max$ *Subject To the Constraint* $\langle \psi \,|\, C \,|\, \psi \rangle = 0$", where we considered a single quadratic

[9] The equation for Lagrange multipliers (A12) produces an arbitrary matrix $\widetilde{\lambda}_{ij}$; a variation of the constraints produces Hermitian matrix λ_{ij}. Lagrange multipliers in (A7) should be set to make the first variation at given u_{jk} as close to zero as possible; least squares expansion of the first variation ($D \times n$ matrix) in Lagrange multipliers ($D \times D$ matrix) gives (A12). For an arbitrary matrix $\mathcal{A}$ it's best approximation by a Hermitian matrix $\mathcal{B}$ is the Hermitian part $\mathcal{B} = \mathcal{A}_H = \frac{1}{2}(\mathcal{A} + \mathcal{A}^\dagger)$. This follows immediately from the Frobenius norm triangle inequality by splitting the matrix into Hermitian $\mathcal{A}_H$ and anti–Hermitian $\mathcal{A}_{AH}$ parts: $\|\mathcal{B} - \mathcal{A}\|_F = \|\mathcal{B} - \mathcal{A}_H - \mathcal{A}_{AH}\|_F \le \|\mathcal{B} - \mathcal{A}_H\|_F + \|\mathcal{A}_{AH}\|_F$.

constraint, similar iterative algorithm converges fast but may fail when optimization and constraint matrices have a number of eigenvectors in common. The optimization problem (A3) subject to (A4) constraints is a problem of (52) type, it has a more complex internal structure than the problem considered in [15].

The described Lagrange multipliers algorithm is based on eigenvalue problem solution: (A6) with partial constraint (A5) as normalizing: $D = \langle \psi^2 \rangle$. It is much less sensitive to degeneracy than Newtonian type iterations, where even a single degenerate degree of freedom makes linear system (with Hessian matrix) iteration to fail. A question arise when the described above iterative algorithm fails. Currently — we do not have the exact answer; the condition of iterative algorithm convergence requires a separate study. The algorithm does not converge well for partially unitary operators with $D < n$, but given large enough iterations number it produces a good enough solution. The reason for a slow convergence is that with (A12) λ_{ij} the Hessian matrix is degenerated at the adjusted u_{jk} (A11) — at this u_{jk} not only first but also second variation of the objective function is zero; this is a constraint qualification problem. The algorithm does not diverge, it provides a sequence of close to optimal solutions. See `com/polytechnik/kgo/KGOIterationalSimpleOptimizationU.java` for a numerical implementation. We also tried to find an algorithm of contraction mapping type, but this requires more study. The convergence can be greatly improved using linear constraints, see Appendix A 4 below where the constraints (A4) were replaces by the closeness of u_{jk} to current iteration value (A40). In many situation, however, an approximate solution is sufficient.

1. On Constrained Optimization In The Singular Values Basis

Before we go further let us discuss the roles of (A10) singular values and their relation to the calculation of Lagrange multipliers. If we write optimization problem (A6) in SVD basis (A10) the u_{jk} is represented as a product of three matrices. The constraints (A4) require all singular values $\Sigma_{jj} = \pm 1$. We denote this diagonal matrix as vector Σ_s. The objective function (A1) is then $\mathcal{F} = \sum_{s,s'=0}^{D-1} \Sigma_s \widetilde{S}_{ss'} \Sigma_{s'}$. Obtain constrained optimization problem with D Lagrange multipliers $\widetilde{\lambda}_s$

$$\sum_{s,s'=0}^{D-1} \Sigma_s \widetilde{S}_{ss'} \Sigma_{s'} + \sum_{s=0}^{D-1} \widetilde{\lambda}_s \left[1 - \Sigma_s^2\right] \to \max \tag{A14}$$

$$\widetilde{S}_{ss'} = \sum_{j,j'=0}^{D-1} \sum_{k,k'=0}^{n-1} U_{js} V^{\dagger}_{sk} S_{jk;j'k'} U_{j's'} V^{\dagger}_{s'k'} \tag{A15}$$

from which we immediately obtain the values

$$\widetilde{\lambda}_s = \frac{1}{\Sigma_s} \sum_{s'=0}^{D-1} \widetilde{S}_{ss'} \Sigma_{s'} \tag{A16}$$

for all adjusted $\Sigma_s = 1$

$$\widetilde{\lambda}_s = \sum_{s'=0}^{D-1} \widetilde{S}_{ss'} \tag{A17}$$

Comparing (A14) with (A6) obtain λ_{ij} in original basis

$$\widetilde{\lambda}_s = \sum_{i,j=0}^{D-1} \lambda_{ij} U_{is} U_{js} \tag{A18}$$

$$\lambda_{ij} = \sum_{s=0}^{D-1} U_{is} U_{js} \widetilde{\lambda}_s \tag{A19}$$

Whereas the original functional (A6) has D^2 Lagrange multipliers λ_{ij}, the (A14) has only D — a constraint for every singular value of the matrix u_{jk}; it is clear why: since the partial constraint (A5) is always satisfies from the eigenproblem it is sufficient to set $D-1$ diagonal elements of (A4) to 1, then all off–diagonal elements are immediately zero.

2. On Iteration Step Without Using The SVD

In the algorithm above we extensively used SVD expansion (A10) for iterations. Let us consider how to avoid using the SVD by replacing it with an eigenvalue problem of the dimension $D \times D$ for the purpose of both: computational complexity and better understanding of the algorithm. Obtained partial constraint (A5) solution matrix u_{jk} is non–orthogonal, the Gram matrix is:

$$G^u_{jj'} = \sum_{k=0}^{n-1} u_{jk} u_{j'k} \tag{A20}$$

We need to "adjust" u_{jk} to satisfy the full set of (A4) constraints. Consider the eigenstates of the Gram matrix

$$\left| G^u | u^{[i]} \right\rangle = \lambda^{[i]}_G \left| u^{[i]} \right\rangle \tag{A21}$$

The eigenvalues of this problem are equal to the singular values (A10) squared $\lambda_G^{[i]} = \Sigma_{ii}^2$. Setting all $\lambda_G^{[i]} = 1$ (eigenvalues adjustment technique [17]) produces a new basis in which (A4) constraints are satisfied in full. The result is identical to the transform (A11) of setting all $\Sigma_{jj} = 1$ but it is obtained without solving a SVD problem, the eigenvalue $D \times D$ problem (A21) is used instead, see `com/polytechnik/kgo/KGOEVSelection.java:getEVAdjustedTo1()` for an implementation.

Optimization problem is question is invariant relatively a unitary transform (the A_{sj} is a unitary matrix)

$$v_{sk} = \sum_{j=0}^{D-1} A_{sj} u_{jk} \tag{A22}$$

The tensor $S_{jk;j'k'}$ transforms with A_{sj} as (A23), Gram matrix (A20) corresponds to the tensor $S_{jk;j'k'} = G^u_{jj'}\delta_{kk'}$.

$$S_{sk;s'k'} = \sum_{jj'=0}^{D-1} A_{sj} S_{jk;j'k'} A_{s'j'} \tag{A23}$$

$$\mathcal{F} = \sum_{j,j'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{jk} S_{jk;j'k'} u_{j'k'} = \sum_{s,s'=0}^{D-1} \sum_{k,k'=0}^{n-1} v_{sk} S_{sk;s'k'} v_{s'k'} \tag{A24}$$

The constraints for new variables v_{sk} have the same form (A4)

$$\delta_{ss'} = \sum_{k,k'=0}^{n-1} v_{sk} v_{s'k} \qquad\qquad s, s' = 0 \ldots D-1 \tag{A25}$$

Let us transform the input to the basis of Gram matrix eigenvectors. Solve generalized eigenproblem (A21) to find the eigenvalues $\lambda_G^{[s]}$ and the eigenvectors $v_j^{[s]}$ of the Gram matrix $G^u_{jj'}$

$$\sum_{j'=0}^{D-1} G^u_{jj'} v_{j'}^{[s]} = \lambda_G^{[s]} v_j^{[s]} \tag{A26}$$

Were it all $\lambda_G^{[s]} = 1$ — the eigenstates of the Gram matrix would form the sought partially unitary operator, but this is typically not. Take Gram matrix eigenvectors as a new basis, the unitary transform matrix is $A_{sj} = v_j^{[s]}$, and write optimization problem (A24) in this new basis v_{sk} (A22) with the tensor $S_{sk;s'k'}$ transformed from the $S_{jk;j'k'}$ according to (A23). If all scaling coefficients $\mu_s = 1$ — this would be exactly the original problem since it is invariant

relatively unitary transforms of the basis, but if we put the factors μ_s (A27) — this makes the solution to satisfy (A25); non–unitary scaling factors μ_s adjust the solution to satisfy the full set of the constraints.

$$\mu_s = \pm \frac{1}{\sqrt{\lambda_G^{[s]}}} \tag{A27}$$

$$\sum_{s,s'=0}^{D-1} \sum_{k,k'=0}^{n-1} \mu_s v_{sk} S_{sk;s'k'} v_{s'k'} \mu_{s'} \underset{v}{\longrightarrow} \max \tag{A28}$$

This scaling adjustment performed in Gram matrix basis is an alternative to SVD adjustment (A11). One need to convert the problem from original basis to the basis of Gram matrix eigenvectors, then scale them by the (A27) factors. The $\mu_s v_{sk}$ satisfies partial orthogonality constraints. We can write optimization problem in this new basis, and perform the iterative algorithm of Appendix A above, then "chaining" unitary transforms as iterations proceed, the result will be identical as the problem is invariant relatively these transforms, but the idea of solution adjustment in the from of pure scaling opens a number of new ways to improve the algorithm, see `com/polytechnik/kgo/KGOIterationalMultipleTransforms.java` for a numerical implementation.

3. On Operator–Dependent Solution Adjustment

In the previous section we considered solution adjustment procedure applied to some initial "partial" solution. This adjustment is a non–unitary basis transform. A question arise about a generalization: applying some other non–unitary transform before the adjustment. Optimization problem in question is to maximize (A29) subject to (A30) constraints:

$$\frac{\mathcal{F}}{D} = \frac{\sum_{j,j'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{jk} S_{jk;j'k'} u_{j'k'}}{\sum_{j=0}^{D-1} \sum_{k=0}^{n-1} u_{jk}^2} \underset{u}{\longrightarrow} \max \tag{A29}$$

$$\delta_{jj'} = \sum_{k=0}^{n-1} u_{jk} u_{j'k} \qquad j, j' = 0 \ldots D-1 \tag{A30}$$

Consider a Hermitian operator $\mathcal{J}$ with matrix elements $\mathcal{J}_{jj'}$, this can be e.g. Lagrange multipliers matrix (A13), unit matrix, etc. A generalized eigenvalue problem with $\mathcal{J}_{jj'}$ and

$G^u_{jj'}$ (A20) matrices is formulated as

$$\sum_{j'=0}^{D-1} \mathcal{J}_{jj'} v^{[s]}_{j'} = \lambda^{[s]}_{\mathcal{J}} \sum_{j'=0}^{D-1} G^u_{jj'} v^{[s]}_{j'} \tag{A31}$$

Because of the Gram matrix $G^u_{jj'}$ in the right hand side obtained solution

$$v_{sk} = \sum_{j=0}^{D-1} v^{[s]}_j u_{jk} \tag{A32}$$

satisfies (A30) constraints $\delta_{ss'} = \sum_{k,k'=0}^{n-1} v_{sk} v_{s'k}$. The transform $v^{[s]}_j$ is non–unitary

$$\delta_{ss'} = \sum_{j,j'=0}^{D-1} v^{[s]}_j G^u_{jj'} v^{[s']}_{j'} \tag{A33a}$$

$$G^{u;-1}_{jj'} = \sum_{s=0}^{D-1} v^{[s]}_j v^{[s]}_{j'} \tag{A33b}$$

Condition (A33a) creates the basis (A32) satisfying partial orthogonality constraints. Let us write the optimization problem (A29) in this new basis v_{sk}. Using

$$u_{jk} = \sum_{j',s=0}^{D-1} G^u_{jj'} v^{[s]}_{j'} v_{sk} \tag{A34}$$

obtain the original problem (A29) with the tensor $S_{sk;s'k'}$ instead of $S_{jk;j'k'}$

$$S_{sk;s'k'} = \sum_{j,j',i,i'=0}^{D-1} v^{[s]}_i G^u_{ij} S_{jk;j'k'} G^u_{i'j'} v^{[s']}_{i'} \tag{A35}$$

This is a generalization of (A23) to non–unitary transforms. This is exactly the original problem (without an adjustment), but written in the v_{sk} basis.

It can be noticed that adjustment procedure of previous section is actually a non–unitary transform with the inverse square root of the Gram matrix $G^{u;-1/2}_{jj'}$ (A27); there are 2^{D-1} distinct combinations of signs but we take all equal to 1. The adjustment is equivalent to multiplying (A34) by $G^{u;-1/2}_{jj'}$ to obtain the "adjusted" tensor

$$S^{adj}_{sk;s'k'} = \sum_{j,j',i,i'=0}^{D-1} v^{[s]}_i G^{u;1/2}_{ij} S_{jk;j'k'} G^{u;1/2}_{i'j'} v^{[s']}_{i'} \tag{A36}$$

This way the adjustment is "transferred" from the state u_{jk} to operator $S_{sk;s'k'}$. Equivalent u_{jk} adjustment corresponds to $u_{jk} = \sum_{j',s=0}^{D-1} G^{u;1/2}_{jj'} v^{[s]}_{j'} v_{sk}$. The (A36) is an important option to transfer an adjustment from a state to tensor, this allows to combine the adjustment with optimization algorithm. Considered in Section A 2 above adjustment procedure corresponds to $\mathcal{J}_{jj'}$ being a unit matrix.

4. On Optimization Algorithm With Linear Constraint Iteration

In previous sections we considered optimization algorithm with quadratic constraints of (A4) form. In numerical implementation (A6) these constraints lead to a poor convergence since at the point the constraints are applied the Hessian matrix is degenerated. Consider a linear type of constraints.

Extend u_{jk} with one more degree of freedom χ to form a "vector" of the dimension $D \times n + 1$.

$$\boldsymbol{z} = \begin{pmatrix} u_{jk} \\ \chi \end{pmatrix} \tag{A37}$$

Then the quadratic from

$$\mathcal{F} = \frac{\boldsymbol{z}^T \mathcal{S} \boldsymbol{z}}{\boldsymbol{z}^T \boldsymbol{z}} = \frac{\chi^2 S_0 + 2\chi \sum\limits_{j=0}^{D-1} \sum\limits_{k=0}^{n-1} b_{jk} u_{jk} + \sum\limits_{j,j'=0}^{D-1} \sum\limits_{k,k'=0}^{n-1} u_{jk} S_{jk;j'k'} u_{j'k'}}{\chi^2 + \sum\limits_{j=0}^{D-1} \sum\limits_{k=0}^{n-1} u_{jk}^2} \tag{A38}$$

has the matrix $\mathcal{S}$

$$\mathcal{S} = \left(\begin{array}{c|c} S_{jk;j'k'} & b_{j'k'} \\ \hline b_{jk} & S_0 \end{array} \right) \tag{A39}$$

The idea is to consider the b_{jk} and S_0 as some kind of "Lagrange Multipliers" to set the variation of (A38) to zero at the "adjusted" u_{jk}, denote it as iteration value u_{jk}^{IT}. Consider the constraints

$$u_{jk} - u_{jk}^{IT} = 0 \tag{A40}$$

this is the closeness of u_{jk} to current iteration value u_{jk}^{IT} (adjusted value satisfying all the required constraints (A4)). A one more degree of freedom χ was introduced to preserve the form of the Rayleigh quotient for the optimization problem (A38). Variating it over u_{jk} and χ obtain (A43) and (A44) respectively; in these formulas $u_{jk} = u_{jk}^{IT}$, F_0^{IT} is a known constant, and B_0 and S_0 are unknown constants.

$$F_0^{IT} = \sum_{j,j'=0}^{D-1} \sum_{k,k'=0}^{n-1} u_{jk} S_{jk;j'k'} u_{j'k'} \tag{A41}$$

$$B_0 = \sum_{j=0}^{D-1} \sum_{k=0}^{n-1} b_{jk} u_{jk} \tag{A42}$$

$$0 = \sum_{j'=0}^{D-1} \sum_{k'=0}^{n-1} S_{jk;j'k'} u_{j'k'} + \chi b_{jk} - \frac{\chi^2 S_0 + 2\chi B_0 + F_0^{IT}}{D + \chi^2} u_{jk} \tag{A43}$$

$$0 = \chi S_0 + B_0 - \frac{\chi^2 S_0 + 2\chi B_0 + F_0^{IT}}{D + \chi^2} \chi \tag{A44}$$

$$0 = F_0^{IT} + \chi B_0 - \left(\chi^2 S_0 + 2\chi B_0 + F_0^{IT}\right) \frac{D}{D + \chi^2} \tag{A45}$$

Multiply (A43) by u_{jk} and sum it over j and k, obtain (A45). For a given χ the (A44) and (A45) can be considered as a linear system for B_0 and S_0. Obtained 2×2 linear system is degenerated and has multiple solutions:

$$(D - \chi^2) B_0 + (D\chi) S_0 = F_0^{IT} \chi \tag{A46}$$

The specific set (χ, S_0, B_0) should be selected for best convergence. The selection

$$\chi = 1 \tag{A47a}$$

$$S_0 = F_0^{IT} \tag{A47b}$$

$$B_0 = -S_0 \tag{A47c}$$

is the first one to try.

1. Take the values of b_{jk} and S_0 to construct (A39).

2. Solve (A38) and select the most appropriate vector $\mathbf{z}$. The result of this step — the "adjusted" u_{jk}^{IT} satisfying all the required constraints (A4).

3. Take this new u_{jk}^{IT}, and select some value of χ, for example (A47), calculate "Lagrange Multipliers" b_{jk} (A43) and S_0 (A46) to construct (A39) matrix. If one uses χ value from (A38) maximization problem — iterations typically stick to some local maximum. If one uses a fixed value for χ, such as (A47a) — a convergence is observed; not very fast, but better than in the Appendix A above. Repeat iteration process. On the first iteration take $b_{jk} = S_0 = 0$.

This "Linear constraints" algorithm is implemented in the `com/polytechnik/kgo/KGOIterationalLinearConstraintsE.java`. An attempt to use D extra degrees of freedom instead of a single one was much less successful `com/polytechnik/kgo/KGOIterationalLinearConstraintsExtraDegreesOfFreedom.java`.

5. An Algorithm to Find an Approximate Solution to the Knowledge Generalizing Operator

Consider the same problem (A3) subject to (A4) constraint. The bases are considered already orthogonalized: $\delta_{jj'} = \langle f_j \,|\, f_{j'} \rangle$ and $\delta_{kk'} = \langle x_k \,|\, x_{k'} \rangle$. Assume we found optimization problem (A3) solution with "partial" constraints (A5), this is (A6) with $\lambda_{ij} = 0$. Put it to (A2) and expand u_{jk} in SVD:

$$u_{jk} = \sum_{j'=0}^{D-1} \sum_{k'=0}^{n-1} U_{jj'} \Sigma_{j'k'} V^{\dagger}_{k'k} \tag{A48}$$

$$\langle f_j \,|\, f_{j'} \rangle = \sum_{k,k'=0}^{n-1} u_{jk} \langle x_k \,|\, x_{k'} \rangle u^{*}_{j'k'} \qquad\qquad j, j' = 0 \ldots D-1 \tag{A49}$$

Write (A49) for orthogonal bases $\mathfrak{x}_k$ and $\mathfrak{f}_j$

$$\mathfrak{x}_k = \sum_{k'=0}^{n-1} V^{\dagger}_{kk'} x_{k'} \tag{A50}$$

$$\mathfrak{f}_j = \sum_{j'=0}^{D-1} U^{\dagger}_{jj'} f_{j'} \tag{A51}$$

$$\langle \mathfrak{f}_j \,|\, \mathfrak{f}_{j'} \rangle = \sum_{k,k'=0}^{n-1} \Sigma_{jk} \langle \mathfrak{x}_k \,|\, \mathfrak{x}_{k'} \rangle \Sigma_{j'k'} \qquad\qquad j, j' = 0 \ldots D-1 \tag{A52}$$

The (A52) is (A49) written in $\mathfrak{x}_k$ and $\mathfrak{f}_j$ orthogonal bases. Since $\delta_{jj'} = \langle \mathfrak{f}_j \,|\, \mathfrak{f}_{j'} \rangle$ and $\delta_{kk'} = \langle \mathfrak{x}_k \,|\, \mathfrak{x}_{k'} \rangle$ the (A52) is satisfied only when all singular values of u_{jk} are ± 1. Actually we made a single iteration of the algorithm above, this Σ_{jk}-adjusted solution is an approximate solution one should try first. Since Σ_{jk} is diagonal, in the $(\mathfrak{f}_j, \mathfrak{x}_k)$ basis we have a one–to–one relation

$$\mathfrak{f}_j = \mathfrak{x}_j \Sigma_{jj} \tag{A53}$$

This is *not* a least squares type of relation, for example the result is invariant relatively the transform $\Sigma_{jk} \to -\Sigma_{jk}$. If $(\mathfrak{f}_j, \mathfrak{x}_k)$ basis satisfies (A52) then all singular values are ± 1 (the condition of partial unitarity) and $(\mathfrak{f}_j, \mathfrak{x}_k)$ relation is plain $\mathfrak{f}_j = \pm \mathfrak{x}_j$. The probability (37) in this basis is

$$\langle \psi_{\mathfrak{f}} \,|\, \mathcal{U} \,|\, \psi_{\mathfrak{x}} \rangle^2 = \frac{\left| \sum\limits_{j=0}^{D-1} \mathfrak{f}_j \mathfrak{x}_j \Sigma_{jj} \right|^2}{\sum\limits_{j=0}^{D-1} \mathfrak{f}_j^2 \sum\limits_{k=0}^{n-1} \mathfrak{x}_k^2} \tag{A54}$$

Partial unitarity "adjusted" case corresponds to $\Sigma_{jj} = \pm 1$.

Consider the meaning of a state with an arbitrary $\Sigma_{jj'}$. The (A52) is actually the constraint (A4) but with the positive diagonal matrix $\Sigma^2_{jj'}$, not $\delta_{jj'}$. What does it mean if we put this u_{jk} "partial constraint (A5)" solution to probability (37) without any adjustment? This breaks the preservation of probability, the probability (A54) is no longer $[0:1]$ bounded, it is now $0 \leq P(\mathbf{f})\Big|_{\mathbf{x}} \leq \max\limits_{j} \Sigma^2_{jj}$; the range $[0:1]$ holds only "on average", for the entire sample. However, this does not change the calculation of outcome value (49). One can also modify (A54) to have the probability $[0:1]$ bounded, the maximal value is 1, it corresponds to $\mathfrak{f}_j = \mathfrak{x}_j \Sigma_{jj}$.

$$P(\mathfrak{f})\Big|_{\mathfrak{x}} \approx \frac{\left| \sum\limits_{j=0}^{D-1} \mathfrak{f}_j \mathfrak{x}_j \Sigma_{jj} \right|^2}{\sum\limits_{j=0}^{D-1} \mathfrak{f}_j^2 \sum\limits_{j=0}^{D-1} \mathfrak{x}_j^2 \Sigma^2_{jj}} \tag{A55}$$

But this is only for evaluation, this is not the function used in optimization problem, optimization problem with the probability (B1) is much more difficult. There is a trivial option to use the probability (B2) for optimization and (A55) for evaluation. The Σ^2_{jj}, $j = 0 \ldots D-1$, factor (whether the singular values are adjusted or not) in the denominator prevents a decrease of probability when polluting the $\mathbf{x}$–space with a large number of completely random components (B6); the value of $\mathfrak{f}$ (49) does not depend on this $\mathfrak{x}$–depended factor, maximal value of probability corresponds to $\mathfrak{f}_j = \mathfrak{x}_j \Sigma_{jj}$; the probability is invariant with respect to $\mathfrak{f}_j \to C\mathfrak{f}_j$, normalize it to const to obtain actual values. This partial constraint solution of (A3) subject to (A5) is an approximate solution one may try. Whereas a quantum channel that preserves probability "on average" does not have a physical meaning, in data analysis it is an approximation with a clear meaning: it emphasizes (A52) internal relations with high probability, the Σ_{jj} factor in (A55). Mathematically this means that in (52) we allow operators $\mathcal{U}$ that preserve Gram matrix spur: $D = \sum\limits_{j=0}^{D-1} \Sigma^2_{jj}$, not the Gram matrix itself (A49) as previously considered; the solution can be found from eigenproblem (A6) in original basis with $\lambda_{ij} = 0$.

Conceptually, this technique consists in taking any approximate u_{jk}, such as least squares (7) or any other matrix, not necessary (A6) solution, Gram matrix spur preservation is not required, expanding u_{jk} in SVD (A48), then set Σ_{jj} to 1 or -1. There are 2^{m-1} distinct combinations, typically the minimal change adjustment — all $\Sigma_{jj} = 1$ gives the best result as

the initial Σ_{jj} are positive. Obtained new u_{jk} matrix with singular values equal to ± 1 satisfies all the required constraints (A49). Alternatively one can solve the eigenproblem (A21) and adjust all the eigenvectors by the factors $\pm 1/\sqrt{\lambda^{[j]}}$ (A27) to obtain the same solution without using the SVD (it is equivalent to multiplication (A36) of unadjusted u_{jk} by inverse square root of corresponding Gram matrix).

Appendix B: On Adjusted Normalizing Of Probability

The probability (37) has a normalizing factor as a product of two Christoffel functions: on $\mathbf{x}$ and on $\mathbf{f}$ (33); these two Christoffel functions have n and m degrees of freedom respectively. In some situations it is convenient to construct a normalizing factor where both $\mathbf{x}$- and $\mathbf{f}$-factors have the same number of degrees of freedom: m.

One can consider the probability adjusted to only "important" $\mathbf{x}$–components, this is $\left\langle \psi_{\mathbf{g}} \,\middle|\, \psi_{u(\mathbf{y})} \right\rangle^2$ from (64) expanded:

$$\langle \psi_{\mathbf{g}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{y}} \rangle^2 = \frac{\left| \sum\limits_{k=0}^{n-1} \sum\limits_{j,s=0}^{m-1} g_j G^{\mathbf{f};\,-1}_{js} u_{sk} y_k \right|^2}{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};\,-1}_{jj'} g_{j'} \sum\limits_{j,j'=0}^{m-1} \sum\limits_{k,k'=0}^{n-1} y_k u_{jk} G^{\mathbf{f};\,-1}_{jj'} u_{j'k'} y_{k'}} \tag{B1}$$

Whereas this formula for $\left\langle \psi_{\mathbf{g}} \,\middle|\, \psi_{u(\mathbf{y})} \right\rangle^2$ has a more suitable normalizing than (37), it has u_{jk} in the denominator and the problem can no longer[10] be reduced to the one of form (40) that requires only the moments of Christoffel functions product (33). For probability evaluation, not for optimization, this can be done straightforward (A55). A quantum channel u_{jk} optimizing (18) with the probability (B1) is an interesting direction of future research, this new problem is no longer a QCQP problem — it is a problem to maximize the sum of M ratios of two quadratic forms on u_{jk} subject to (38) constraint or, more generally, an unconstrained optimization of (64). The one in the numerator is a dyadic product squared, the one in the denominator is non–negative, it cancels with the numerator when it's value is close to zero.

To adjust the number of degrees of freedom one can use a much simpler alternative approach. All we need is to calculate a Christoffel function in $\mathbf{x}$–space to normalize the

[10] This difficulty does not arise with $\mathbf{x}$- and $\mathbf{f}$- being the same space. For example for a unitary $\mathcal{U}$ the denominator does not depend on $\mathcal{U}$.

probability. A trivial approach is to use the contributing subspace $\left|\phi^{[i]}\right\rangle$, e.g. from (20). Despite the moments $\langle f_j x_k\rangle$ have been used to build the contributing subspace $\left|\phi^{[i]}\right\rangle$, this does not create any difficulty as we use these projections only to construct a Christoffel function with matched number of degrees of freedom. The (B5) is invariant with respect to $\langle f_j x_k\rangle \to -\langle f_j x_k\rangle$ and tends to a constant when any $\langle f_j x_k\rangle \to \infty$ (factors in the denominator and inverse $G^{fxa}_{jj'}$ matrix).

$$\langle \psi_{\mathbf{g}} \,|\, \mathcal{U} \,|\, \psi_{\mathbf{y}} \rangle^2 = \frac{\left| \sum\limits_{k=0}^{n-1} \sum\limits_{j,s=0}^{m-1} g_j G^{\mathbf{f};\,-1}_{js} u_{sk} y_k \right|^2}{\sum\limits_{j,j'=0}^{m-1} g_j G^{\mathbf{f};\,-1}_{jj'} g_{j'} \sum\limits_{i=0}^{m-1} {\phi^{[i]}}^2(\mathbf{y})} \tag{B2}$$

$$G^{fxa}_{jj'} = \sum_{k,k'=0}^{n-1} \langle f_j x_k\rangle \, G^{\mathbf{x};\,-1}_{kk'} \, \langle x_{k'} f_{j'}\rangle \tag{B3}$$

$$G^{C}_{qq'} = \sum_{s,s'=0}^{n-1} \sum_{j,j'=0}^{m-1} G^{\mathbf{x};\,-1}_{qs} \, \langle x_s f_j\rangle \, G^{fxa;\,-1}_{jj'} \, \langle f_{j'} x_{s'}\rangle \, G^{\mathbf{x};\,-1}_{s'q'} \tag{B4}$$

$$K^{adj}(\mathbf{x}) = \frac{1}{\sum\limits_{i=0}^{m-1} {\phi^{[i]}}^2(\mathbf{x})} = \frac{1}{\sum\limits_{q,q'=0}^{n-1} x_q G^{C}_{qq'} x_{q'}} \tag{B5}$$

The value of $K^{adj}(\mathbf{x})$ is never zero on training sample since contributing subspace always has a constant among the components. The probability (B2) uses Christoffel function with adjusted number of degrees of freedom $K^{adj}(\mathbf{x})$ (B5) instead of the original $K(\mathbf{x})$ (10) for the probability (37). The difference between two these Christoffel functions is in extra terms in the denominator sum. Since the entire $\mathbf{x}$–space can be represented as the direct sum of $\left|\phi^{[i]}\right\rangle$ and $\left|\phi^{O;[i]}\right\rangle$, a subspace of $\mathbf{x}$ orthogonal to $\left|\phi^{[i]}\right\rangle$, the $K(\mathbf{x})$ (10) is:

$$K(\mathbf{x}) = \frac{1}{\sum\limits_{i=0}^{m-1} {\phi^{[i]}}^2(\mathbf{x}) + \sum\limits_{i=m}^{n-1} {\phi^{O;[i]}}^2(\mathbf{x})} = \frac{1}{\sum\limits_{k,k'=0}^{n-1} x_k G^{\mathbf{x};\,-1}_{kk'} x_{k'}} \tag{B6}$$

Thus we always have $K^{adj}(\mathbf{x}) \geq K(\mathbf{x})$. The moments of two Christoffel functions product are

$$\left\langle x_k f_j \,\middle|\, K^{adj(\mathbf{x})} K^{(\mathbf{f})} \,\middle|\, x_{k'} f_{j'} \right\rangle = \sum_{l=0}^{M} \omega^{(l)} \frac{x^{(l)}_k x^{(l)}_{k'}}{\sum\limits_{q,q'=0}^{n-1} x^{(l)}_q G^{C}_{qq'} x^{(l)}_{q'}} \cdot \frac{f^{(l)}_j f^{(l)}_{j'}}{\sum\limits_{s,s'=0}^{m-1} f^{(l)}_s G^{\mathbf{f};\,-1}_{ss'} f^{(l)}_{s'}} \tag{B7}$$

This tensor has the same dimensions as (33), the difference only in normalizing — it uses $G^{C}_{qq'}$ from (B4) instead of $G^{\mathbf{x};\,-1}_{qq'}$ in (33). Despite it now depends on $\langle f_j x_k\rangle$ moments — they are used only to construct Christoffel function for normalizing, this does not change the essence of the solution due to the invariance properties of the Christoffel function.

Appendix C: On Contributing Subspace Selection

Considered above optimization problem finds partially unitary operator u_{jk} that does both: selects the contributing subspace (m vectors of the dimension n) and optimizes the objective function. Besides computational difficulties this also creates a problem with normalizing since properly normalized objective function (64) has operator u_{jk} both in the numerator and in the denominator (B1), thus some surrogate normalizing (39), (B7), (46), or (47) was used instead. It is a very attractive option to split the problem into two:

- Find the contributing subspace $\phi^{[j]}$ of the dimension m.
- Find a unitary (not partially unitary!) operator $\mathcal{U}$ mapping from $\phi^{[j]}$ space to f_j space.

A simple "projective" example with contributing subspace was considered in Eq. (27) above. The matrix $K^{(\mathbf{f}\to\mathbf{x})}_{kk'}$ from (25) has the rank at most m and the best what can be obtained in the projective paradigm is a solution[13] of "direct projection" type where the least squares expansion $\mathbf{f}_{LS}(\mathbf{x})$ of $|f_j\rangle$ in $|x_k\rangle$ (7) is used as the localization point in (13) to obtain the state $\left|\psi_{\mathbf{f}_{LS}(\mathbf{x})}\right\rangle$ to be used in calculation of probabilities.

Properly normalized objective function (64) maximizes the probability transferred from $\mathbf{x}$ to $\mathbf{f}$. Consider a much simpler problem: find a subspace of $\mathbf{x}$ contributing to the coverage of $\mathbf{f}$. The $\mathbf{f}$–coverage is determined by $\mathbf{f}$–Christoffel function $K^{(\mathbf{f})}(\mathbf{g})$ from (23). Consider it's values in a $\psi(\mathbf{x})$ state

$$\text{Coverage}_{\psi} = \frac{\left\langle \psi \,\middle|\, K^{(\mathbf{f})} \,\middle|\, \psi \right\rangle}{\langle \psi \,|\, \psi \rangle} \tag{C1}$$

Previously we considered a similar problem where the Christoffel function K and ψ both were functions on $\mathbf{x}$, see [18], *Appendix B: On The Christoffel Function Spectrum*. Now the Christoffel function is a function on $\mathbf{f}$, and ψ is a function on $\mathbf{x}$. The (C1) can be similarly expanded in spectrum of $\mathbf{f}$–Christoffel function matrix

$$\left\langle x_k \,\middle|\, K^{(\mathbf{f})} \,\middle|\, x_{k'} \right\rangle = \sum_{l=1}^{M} \frac{x^{(l)}_k x^{(l)}_{k'}}{\sum\limits_{j,j'=0}^{m-1} f^{(l)}_j G^{\mathbf{f};-1}_{jj'} f^{(l)}_{j'}} \omega^{(l)} \tag{C2}$$

It is different from (24) with x–moments instead of f–moments. Consider generalized eigenvalue problem

$$\phi^{[i]} = \sum_{k=0}^{n-1} \alpha^{\phi;[i]}_k x_k \qquad\qquad i = 0 \ldots n-1; \tag{C3}$$

$$\delta_{ii'} = \left\langle \phi^{[i]} \,\middle|\, \phi^{[i']} \right\rangle = \sum_{k,k'=0}^{n-1} \alpha_k^{\phi;[i]} \langle x_k x_{k'} \rangle \, \alpha_{k'}^{\phi;[i']} \tag{C4}$$

$$\lambda^{[i]}\delta_{ii'} = \left\langle \phi^{[i]} \,\middle|\, K^{(\mathbf{f})} \,\middle|\, \phi^{[i']} \right\rangle = \sum_{k,k'=0}^{n-1} \alpha_k^{\phi;[i]} \left\langle x_k \,\middle|\, K^{(\mathbf{f})} \,\middle|\, x_{k'} \right\rangle \alpha_{k'}^{\phi;[i']} \tag{C5}$$

$$\sum_{k'=0}^{n-1} \left\langle x_k \,\middle|\, K^{(\mathbf{f})} \,\middle|\, x_{k'} \right\rangle \alpha_{k'}^{\phi;[i]} = \lambda^{[i]} \sum_{k'=0}^{n-1} \langle x_k x_{k'} \rangle \, \alpha_{k'}^{\phi;[i]} \tag{C6}$$

Because $\mathbf{x}$– and $\mathbf{f}$– bases are different the condition[18] $\langle 1 \rangle = \sum_{i=0}^{n-1} \lambda^{[i]}$ no longer holds, it is typically $\langle 1 \rangle \leq \sum_{i=0}^{n-1} \lambda^{[i]}$ since $m \leq n$; moreover the sum of m maximal eigenvalues can possibly exceed the total weight $\langle 1 \rangle \lesseqgtr \sum_{i=0}^{m-1} \lambda^{[i]}$. From Christoffel function invariance it immediately follows that the sum of m maximal eigenvalues is equal to $\langle 1 \rangle$ if $\mathbf{f}$ and $\mathbf{x}$ belong to the same space.

The m eigenstates of (C6) corresponding to m maximal eigenvalues $\lambda^{[i]}$, $i = 0 \ldots n-1$ form the m states contributing most to the coverage. This is an alternative option for the contributing subspace. The problem is now reduced to finding a unitary (not partially unitary) operator $\mathcal{U}$ of the dimension $m \times m$ mapping from ϕ_k to f_j, where $|\phi\rangle = \sum_{k=0}^{m-1} \phi_k \left|\phi^{[k]}\right\rangle$,

$$f_j = \sum_{k=0}^{m-1} u_{jk} \phi_k \tag{C7}$$

In this form the optimization problem is greatly simplified and the $\mathbf{x}$–normalizing in (B1) becomes u_{jk} independent:

$$\langle \psi_{\mathbf{g}} \,|\, \mathcal{U} \,|\, \psi_\phi \rangle^2 = \frac{\left| \sum_{k=0}^{m-1} \sum_{j,s=0}^{m-1} g_j G_{js}^{\mathbf{f};\,-1} u_{sk} \phi_k \right|^2}{\sum_{j,j'=0}^{m-1} g_j G_{jj'}^{\mathbf{f};\,-1} g_{j'} \sum_{j,j'=0}^{m-1} \phi_j G_{jj'}^{\phi;\,-1} \phi_{j'}} \tag{C8}$$

This probability is exactly the same as the one we considered above, but with the ϕ_k used as the input instead of the x_k; we also have $n = m$ thus the operator u_{jk} is unitary!

Appendix D: Software description

- Install java 19 or later.
- Download the latest version of the source code `code_polynomials_quadratures.zip` from [20] or from an alternative location.

- Decompress and recompile the program. Run a selftest.

```
unzip code_polynomials_quadratures.zip
javac -g com/polytechnik/*/*java
java com/polytechnik/kgo/TestKGO
```

- Run the program with bundled deterministic data file, test trivial mapping.

```
java com/polytechnik/kgo/KGO --data_cols=9:0,6:0,4:8:1 \
  --SKtype=FXFX_F_CHRISTOFFEL \
  --approximation=MAXEV_EVADJ \
  --data_file_to_build_model_from=dataexamples/runge_function.csv \
  --output_files_prefix=/tmp/out_
```

- There are a number of `--approximation=` available options. There is no perfect implementation yet available.

An effective algorithm to the problem will be found later in [21, 22], see `com/polytechnik/kgo/KGO IterationalSubspaceLinearConstraints.java` for an implementation. This algorithm, instead of usual iteration internal state in the form of a pair: approximation, Lagrange multipliers: (u_{jk}, λ_{ij}), uses iteration internal state in the form of a triple: approximation, Lagrange multipliers, homogeneous linear constraints $(u_{jk}, \lambda_{ij}, C_{d;jk})$, it is the linear constraints that provide a good convergence. The dimension of eigenvalue problem to solve on each iteration is $Dn - (D-1)(D+2)/2$ instead of Dn of this paper.

---

[1] V. G. Malyshkin, R. Bakhramov, and A. E. Gorodetsky, A Massive Local Rules Search Approach to the Classification Problem, arXiv preprint arXiv:cs/0609007 10.48550/arXiv.cs/0609007 (2001).

[2] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain., Psychological review **65**, 386 (1958).

[3] V. Vapnik and A. Y. Chervonenkis, The method of ordered risk minimization, I, Avtomatika i Telemekhanika **8**, 21 (1974).

[4] V. Vapnik and A. Y. Chervonenkis, On the method of ordered risk minimization, II, Avtomatika i Telemekhanika **9**, 29 (1974).

[5] P. Hájek and T. Havránek, On generation of inductive hypotheses, International Journal of Man-Machine Studies **9**, 415 (1977).

[6] V. Vapnik, *The nature of statistical learning theory* (Springer science & business media, 2013).

[7] I. H. Witten and E. Frank, Data mining: practical machine learning tools and techniques with Java implementations, Acm Sigmod Record **31**, 76 (2002).

[8] L. A. Zadeh, Fuzzy sets, Information and control **8**, 338 (1965).

[9] P. Hájek, Fuzzy logic and arithmetical hierarchy, Fuzzy sets and Systems **73**, 359 (1995).

[10] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence **35**, 1798 (2013).

[11] V. G. Malyshkin, Norm-Free Radon-Nikodym Approach to Machine Learning, ArXiv e-prints 10.48550/arXiv.1512.03219 (2015), `http://arxiv.org/abs/1512.03219`, arXiv:1512.03219 [cs.LG].

[12] A. V. Bobyl, V. V. Davydov, and V. G. Malyshkin, On The Radon–Nikodym Machine Learning Parallelization, in *The 4th International Conference on Future Networks and Distributed Systems (ICFNDS)* (2020) pp. 1–5.

[13] V. G. Malyshkin, On The Radon-Nikodym Spectral Approach With Optimal Clustering, arXiv preprint arXiv:1906.00460 10.48550/arXiv.1906.00460 (2019).

[14] S. Marx, E. Pauwels, T. Weisser, D. Henrion, and J.-B. Lasserre, Tractable semi-algebraic approximation using Christoffel-Darboux kernel, arXiv preprint arXiv:1904.01833 10.48550/arXiv.1904.01833 (2019).

[15] V. G. Malyshkin, Market Dynamics: On Directional Information Derived From (Time, Execution Price, Shares Traded) Transaction Sequences, arXiv preprint arXiv:1903.11530 10.48550/arXiv.1903.11530 (2019).

[16] N. Z. Shor, *Nondifferentiable optimization and polynomial problems*, Vol. 24 (Springer Science & Business Media, 2013).

[17] G. S. Malyshkin, The comparative efficiency of classical and fast projection algorithms in the resolution of weak hydroacoustic signals (Сравнительная эффективность классических и быстрых проекционных алгоритмов при разрешении слабых гидроакустических сигналов), Acoustical Physics **63**, 216 (2017), doi:10.1134/S1063771017020099 (eng) ; doi:10.7868/S0320791917020095

(pyc).

[18] V. G. Malyshkin, On Lebesgue Integral Quadrature, arXiv preprint arXiv:1807.06007 10.48550/arXiv.1807.06007 (2018).

[19] K. Kraus, *States, Effects, and Operations: Fundamental Notions of Quantum Theory*, Lecture Notes in Physics, Vol. 190 (Springer-Verlag, 1983) Lectures in Mathematical Physics at the University of Texas at Austin.

[20] V. G. Malyshkin, The code for polynomials calculation (2014), `http://www.ioffe.ru/LNEPS/malyshkin/code.html` and an alternative location.

[21] M. G. Belov and V. G. Malyshkin, Partially unitary learning, Phys. Rev. E **110**, 055306 (2024).

[22] M. G. Belov, V. V. Dubov, A. V. Filimonov, and V. G. Malyshkin, Quantum channel learning, Phys. Rev. E **111**, 015302 (2025).