A New Approach to Collaborative Filtering: Operator Estimation with Spectral Regularization

Francis Bach, Jacob Abernethy, Jean-Philippe Vert, Theodoros Evgeniou

A New Approach to Collaborative Filtering: Operator Estimation with Spectral Regularization

Jacob Abernethy JAKE@CS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA, USA

Francis Bach FRANCIS.BACH@MINES.ORG
INRIA - Willow project
Département d'Informatique, Ecole Normale Supérieure
45, rue d'Ulm
75230 Paris, France

Theodoros Evgeniou THEODOROS.EVGENIOU@INSEAD.EDU
Decision Sciences and Technology Management
INSEAD
Bd de Constance, 77300 Fontainebleau, France

Jean-Philippe Vert JEAN-PHILIPPE.VERT@ENSMP.FR
Centre for Computational Biology
Ecole des Mines de Paris
Fontainebleau, France

Abstract

We present a general approach for collaborative filtering (CF) using spectral regularization to learn linear operators from "users" to the "objects" they rate. Recent low-rank type matrix completion approaches to CF are shown to be special cases. However, unlike existing regularization based CF methods, our approach can be used to also incorporate information such as attributes of the users or the objects – a limitation of existing regularization based CF methods. We then provide novel representer theorems that we use to develop new estimation methods. We provide learning algorithms based on low-rank decompositions, and test them on a standard CF dataset. The experiments indicate the advantages of generalizing the existing regularization based CF methods to incorporate related information about users and objects. Finally, we show that certain multi-task learning methods can be also seen as special cases of our proposed approach.

1. Introduction

Collaborative filtering (CF) refers to the task of predicting preferences of a given "user" for some "objects" (e.g., books, music, products, people, etc...) based on his/her previously revealed preferences – typically in the form of purchases or ratings – as well as the revealed

preferences of other users. In a book recommender system, for example, one would like to suggest new books to someone based on what he and other users have recently purchased or rated. The ultimate goal of CF is to infer the preferences of users for new to them objects.

A number of CF methods have been developed in the past (Breese et al., 1998, Heckerman et al., 2000, Salakhutdinov et al., 2007). Recently there has been interest in CF using regularization based methods (Srebro and Jaakkola, 2003). This work adds to that literature by developing a novel general approach to developing regularization based CF methods.

Recent regularization based CF methods assume that the only data available are the revealed preferences, and no other information such as background information on the objects or users is given. In this case, one may formulate the problem as that of filling a matrix with users as rows, objects (e.g., books) as columns, and missing entries as currently unknown preferences. Assuming that the available information is in the form of ratings, the existing elements of the matrix are the existing ratings from some users to some objects – typically very few relative to the size of the matrix.

To make useful predictions within this setting, regularization based CF methods make certain assumptions about the *relatedness* of the objects and users. The most common assumption is that preferences can be decomposed into a small number of factors, both for users and objects, resulting in the search for a low-rank matrix which approximates the partially observed matrix of preferences (Srebro and Jaakkola, 2003). The rank constraint can be interpreted as a regularization on the hypothesis space. Since the rank constraint gives rise to a non-convex set of matrices, the associated optimization problem will be a difficult non-convex problem for which only heuristic algorithms exist (Srebro and Jaakkola, 2003). An alternative formulation, proposed by Srebro et al. (2005), suggests penalizing the predicted matrix by its *trace norm*, i.e., the sum of its singular values. An added benefit of the trace norm regularization is that, with a sufficiently large regularization parameter, the final solution will be low-rank (Fazel et al., 2001, Bach, 2007).

However, a key limitation of current regularization based CF methods is that they do not take advantage of information, such as attributes of users (e.g., gender, age) or objects (e.g., book's author, genre), which is often available. Intuitively, such information might be useful to guide the inference of preferences, in particular for users and objects with very few known ratings. For example, at the extreme, users and objects with no prior ratings can not be considered in the standard CF formulation, while their attributes alone could provide some basic preference inference.

The main contribution of this paper is to develop a general framework and specific algorithms for the more general CF setting where other information, such as attributes for users and/or objects, may be available. More precisely, we show that CF, typically seen as a problem of matrix completion, can be thought of more generally as estimating a linear operator from the space of users to the space of objects. Equivalently, this can be viewed as learning a bilinear form between users and objects. We then develop *spectral regularization* based methods to learn such linear operators. When dealing with operators, rather than matrices, one may also work with infinite dimensions, allowing to consider arbitrary feature spaces induced by some kernels. A key theoretical contribution of this paper is that we prove new representer theorems which allow us to develop new methods that learn finitely

many parameters also in the general case of infinite dimensional feature spaces that describe users or objects.

We also show that, with the appropriate choice of kernels for both users and objects, we may consider a number of existing machine learning methods as special cases of our general framework. In particular, we show that several CF methods, such as rank constraint, trace-norm regularization, and Frobenius norm regularization based ones, can all be cast as special cases of spectral regularization on operator spaces. Moreover, particular choices of kernels lead to specific subcases such as regular matrix completion and multitask learning. In the specific application of collaborative filtering with the presence of attributes, we show that our generalization of these subcases leads to better predictive performance.

The outline of the paper is as follows. In Section 2 we review the notion of a compact operator on Hilbert Space, and we show how to cast the collaborative filtering problem within this framework. We then introduce spectral regularization, and discuss how rank constraint, trace norm regularization, and Frobenius norm regularization are all special cases of spectral regularization. In Section 3 we show how our general framework encompasses many existing methods by proper choices of the loss function, the kernels, and the spectral regularizer. In Section 4 we provide three representer theorems for operator estimation with spectral regularization which allow for efficient learning algorithms. Finally, in Section 5 we present a number of algorithms, and describe several techniques to improve efficiency, and we test these algorithms in Section 6 on synthetic examples and a widely used movie database.

2. Learning compact operators with spectral regularization

In this section we propose a mathematical formulation for a general CF problem with spectral regularization.

2.1 A general CF formulation

We consider a general CF problem in which our goal is to model the preference of a user described by $\mathbf{x}$ for an item described by $\mathbf{y}$. We denote by $\mathbf{x}$ and $\mathbf{y}$ the data objects containing all relevant or available information; this could, for example, include a unique identifier i for the i-th user or object. Of course, the users and objects may additionally be characterized by attributes, in which case $\mathbf{x}$ or $\mathbf{y}$ may contain some representation of this extra information. Ultimately, we would like to consider such attribute information as encoded in some positive definite kernel between users, or equivalently between objects. This naturally leads us to model the users as elements in a Hilbert space $\mathcal{X}$, and the objects they rate as elements of another Hilbert space $\mathcal{Y}$.

We assume that our observation data is in the form of *ratings* from users to objects, a real-valued score representing the user's preference for the object. Alternatively, similar methods can be applied when the observations are binary, specifying for instance whether or not a user considered or selected an object.

Given a series of N observations $(\mathbf{x}_i, \mathbf{y}_i, t_i)_{i=1,\ldots,N}$ in $\mathcal{X} \times \mathcal{Y} \times \mathbb{R}$, where t_i represents the rating of user $\mathbf{x}_i$ for object $\mathbf{y}_i$, the generalized CF problem is then to infer a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that can then be used to predict the rating of any user $\mathbf{x} \in \mathcal{X}$ for any object $\mathbf{y} \in \mathcal{Y}$ by $f(\mathbf{x}, \mathbf{y})$. Note that in our notation, $\mathbf{x}_i$ and $\mathbf{y}_i$ represent the user and object corresponding to the i-th rating available. If several ratings of a user for different objects are available, as is commonly the case, several $\mathbf{x}_i$'s will be identical in $\mathcal{X}$ – a slight abuse of notation. We denote by $\mathcal{X}_N$ and $\mathcal{Y}_N$ the linear spans of $\{\mathbf{x}_i : i = 1, \ldots, N\}$ and $\{\mathbf{y}_i : i = 1, \ldots, N\}$ in $\mathcal{X}$ and $\mathcal{Y}$, with respective dimensions $m_{\mathcal{X}}$ and $m_{\mathcal{Y}}$.

For the function to be estimated we restrict ourselves to bilinear forms given by:

$$f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, F\mathbf{y} \rangle_{\mathcal{X}} \tag{1}$$

for some compact operator $F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X})$. For an introduction to relevant concepts in functional analysis, see Appendix A.

In the general case we consider below, one can also first map users $\mathbf{x}$ and objects $\mathbf{y}$ into possibly infinite dimensional feature spaces $\Phi_{\mathcal{X}}(\mathbf{x})$ and $\Psi_{\mathcal{Y}}(\mathbf{y})$ and use kernels. We refer the reader to Appendix A for basic definitions and properties related to compact operators that are useful below. The inference problem can now be stated as follows:

Given a training set of ratings, how may we estimate a "good" compact operator F to predict future ratings using (1)?

We estimate the operator F in (1) from the training data using a standard regularization and statistical machine learning approach. In particular, we propose to define the operator as the solution of an optimization problem over $\mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ whose objective function balances a data fitting term $R_N(F)$, which is small for operators that can correctly explain the training data, with a regularization term $\Omega(F)$. We now describe these two terms in more details.

2.2 Data fitting term

Given a loss function $\ell(t, t')$ that quantifies how good a prediction $t \in \mathbb{R}$ is if the true value is $t' \in \mathbb{R}$, we consider a fitting term equal to the empirical risk, i.e., the mean loss incurred on the training set:

$$R_N(F) = \frac{1}{N} \sum_{i=1}^{N} \ell\left(\langle \mathbf{x}_i, F\mathbf{y}_i \rangle_{\mathcal{X}}, t_i\right) . \tag{2}$$

The particular choice of the loss function should typically depend on the precise problem to be solved and on the nature of the variables t to be predicted. See more details in Section 3.

2.3 Regularization term

For the regularization term, we focus on a class of spectral functions defined as follows.

Definition 1 *A function $\Omega : \mathcal{B}_0(\mathcal{Y}, \mathcal{X}) \mapsto \mathbb{R} \cup \{+\infty\}$ is called a* spectral penalty function *if it can be written as:*

$$\Omega(F) = \sum_{i=1}^{d} s_i\left(\sigma_i(F)\right) , \tag{3}$$

where for any $i \geq 1, s_i : \mathbb{R}^+ \mapsto \mathbb{R}^+ \cup \{+\infty\}$ *is a non-decreasing penalty function satisfying* $s(0) = 0$*, and* $\sigma_i(F)$ *are the* d *singular values of* F *– d possibly infinite.*

Note that by the spectral theorem presented in Appendix A, any compact operator can be decomposed into singular vectors, with singular values being a sequence that tends to zero.

Spectral penalty functions include as special cases several functions often encountered in matrix completion problems:

- For a given integer r, taking $s_i = 0$ for $i = 1, \ldots, r$ and $s_{r+1}(u) = +\infty$ if $u > 0$, leads to the function:
$$\Omega(F) = \begin{cases} 0 & \text{if } \operatorname{rank}(F) \leq r\,, \\ +\infty & \text{otherwise.} \end{cases} \tag{4}$$
In other words, the set of operators F that satisfy $\Omega(F) < +\infty$ is the set of operators with rank smaller than r.

- Taking $s_i(u) = u$ for all i results in the trace norm penalty (see Appendix A):
$$\Omega(F) = \begin{cases} \| F \|_1 & \text{if } F \in \mathcal{B}_1(\mathcal{Y}, \mathcal{X})\,, \\ +\infty & \text{otherwise,} \end{cases} \tag{5}$$
where we note with $\mathcal{B}_1(\mathcal{Y}, \mathcal{X})$ the set of operators with finite trace norm.

- Taking $s_i(u) = u^2$ for all i results in the squared Hilbert-Schmidt norm penalty (also called squared Frobenius norm for matrices, see Appendix A):
$$\Omega(F) = \begin{cases} \| F \|_2^2 & \text{if } F \in \mathcal{B}_2(\mathcal{Y}, \mathcal{X})\,, \\ +\infty & \text{otherwise,} \end{cases} \tag{6}$$
where we note with $\mathcal{B}_2(\mathcal{Y}, \mathcal{X})$ the set of operators with finite squared Hilbert-Schmidt norm.

These particular functions can be combined together in different ways. For example, we may constrain the rank to be smaller than r while penalizing the trace norm of the matrix, which can be obtained by setting $s_i(u) = u$ for $i = 1, \ldots, r$ and $s_{r+1}(u) = +\infty$ if $u > 0$. Alternatively, if we want to penalize the Frobenius norm while constraining the rank, we set $s_i(u) = u^2$ for $i = 1, \ldots, r$ and $s_{r+1}(u) = +\infty$ if $u > 0$. We state these two choices of Ω explicitly since we use these in the experiments (see Section 6) or to design efficient algorithms (see Section 5).

$$\text{Trace+Rank Penalty:} \qquad \Omega(F) = \begin{cases} \| F \|_1 & \text{if } \operatorname{rank}(F) \leq r, \\ +\infty & \text{otherwise.} \end{cases} \tag{7}$$

$$\text{Frobenius+Rank Penalty:} \qquad \Omega(F) = \begin{cases} \| F \|_2^2 & \text{if } \operatorname{rank}(F) \leq r, \\ +\infty & \text{otherwise.} \end{cases} \tag{8}$$

2.4 Operator inference

With both a fitting term and a regularization term, we can now formally define our inference approach. It consists in finding an operator $\hat{F}$, if there exists one, that solves the following optimization problem:

$$\hat{F} \in \underset{F \in \mathcal{B}_0(\mathcal{Y},\mathcal{X})}{\arg\min} \left\{ R_N(F) + \lambda \Omega(F) \right\} , \tag{9}$$

where $\lambda \in \mathbb{R}$ is a parameter that controls the trade-off between fitting and regularization, and where $R_N(F)$ and $\Omega(F)$ are respectively defined in (2) and (3). We note that if the set $(F : \Omega(F) < +\infty)$ is not empty, then necessarily the solution $\hat{F}$ of this optimization problem must satisfy $\Omega(\hat{F}) < \infty$.

Before explaining in Sections 4 and 5 how problem (9) can be solved in practice in particular for Hilbert spaces of infinite dimensions, we provide in the next section several examples of algorithms that can be derived as particular cases of (9) and highlight their relationships to existing methods.

3. Examples and related approaches

The general formulation (9) can result in a variety of practical algorithms potentially useful in different contexts. In particular three elements can be tailored to one's particular needs: the loss function, the kernels, and the spectral penalty terms. We start this section by some generalities about the possible choices for these elements and their consequences, before highlighting some particular combinations of choices relevant for different applications.

1. **The loss function.** The choice of ℓ defines the empirical risk through (2). It is a classical component of many machine learning methods, and should typically depend on the type of data to be predicted (e.g., discrete or continuous) and of the final objective of the algorithm (e.g., classification, regression or ranking). The choice of ℓ also influences the algorithm, as discussed in Section 5. As a deeper discussion about the loss function is only tangential to the current work, we only consider the square loss here, knowing that other convex losses may be considered.

2. **The spectral penalty function.** The choice of $\Omega(F)$ defines the type of constraint we impose on the operator that we seek to learn. We gave in Section 2.3 several examples of such constraints, including the rank constraint (4), the trace norm constraint (5), the Hilbert-Schmidt norm constraint (6), or the trace norm constraint over low-rank operators (7). The choice of a particular penalty might be guided by some considerations about the problem to be solved, e.g., finding low-rank operators as a way to discover low-dimensional latent structures in the data. On the other hand, from an algorithmic perspective, the choice of the spectral penalty may affect the efficiency or feasibility of our learning algorithm. Certain penalty functions, such as the rank constraint for example, will lead to non-convex problems because the corresponding penalty function (4) is not convex itself. On the other hand the same rank constraint can vastly reduce the number of parameters to be learned. These algorithmic considerations are discussed in more details in Section 5.

3. **The kernels.** Our choice of kernels defines the inner products (i.e., embeddings) of the users and objects in their respective Hilbert spaces. We may use a variety of possible kernels depending on the problem to be solved and on the attributes available. Interestingly the choice of a particular kernel has no influence on the algorithm, as we show later (however, it does of course influence the running time of these algorithms). In the current work, we focus on two basic kernels (Dirac kernels and attribute kernels) and in Section 3.4 we discuss combining these.

   - The first kernel we consider is the *Dirac* kernel. When two users (resp. two objects) are compared, the Dirac kernel returns 1 if they are the same user (resp. object), and 0 otherwise. In other words the Dirac kernel amounts to representing the users (resp. the objects) by orthonormal vectors in $\mathcal{X}$ (resp. in $\mathcal{Y}$). This kernel can be used whether or not attributes are available for users and objects. We denote by $k_D^{\mathcal{X}}$ (resp. $k_D^{\mathcal{Y}}$) the Dirac kernel for the users (resp. objects).
   - The second kernel we consider is a kernel between *attributes*, when attributes are available to describe the users and/or objects. We call these "attributes kernels". These would typically be a kernel between vectors, such as the inner product or a Gaussian RBF kernel, when the descriptions of users and/or objects take the form of vectors of real-valued attributes. We denote by $k_A^{\mathcal{X}}$ (resp. $k_A^{\mathcal{Y}}$) the attributes kernel for the users (resp. objects).

Let us now illustrate how specific combinations of loss, spectral penalty and kernels can be relevant for various settings. In particular the choice of kernels leads to specific subcases, namely matrix completion, multi-task learning and pairwise learning. We then consider a new representation that allows interpolation between these subcases.

3.1 Matrix completion

When the Dirac kernel is used for both users and objects, then we can organize the data $\{\mathbf{x}_i, i = 1, \ldots, n\}$ into $n_{\mathcal{X}}$ groups of identical data points and similarly $\{\mathbf{y}_i, i = 1, \ldots, n\}$ into $n_{\mathcal{Y}}$ groups. Since we use the Dirac kernel, we can represent each of these groups by the elements of the canonical basis $(\mathbf{u}_1, \ldots, \mathbf{u}_{n_{\mathcal{X}}})$ and $\left(\mathbf{v}_1, \ldots, \mathbf{v}_{n_{\mathcal{Y}}}\right)$ of $\mathbb{R}^{n_{\mathcal{X}}}$ and $\mathbb{R}^{n_{\mathcal{Y}}}$, respectively. A bilinear form using Dirac kernels only depends on the identities of the users and the objects, and we only predict the rating t_i based on the identities of the groups in both spaces. If we assume that each pair user/object is observed at most once, the data can be re-arranged into a $n_{\mathcal{X}} \times n_{\mathcal{Y}}$ incomplete matrix, the learning objective being to complete this matrix (indeed, in this context, it is not possible to generalize to never seen points in $\mathcal{X}$ and $\mathcal{Y}$).

In this case, our bilinear form framework exactly corresponds to completing the matrix, because the bilinear function of $\mathbf{x}$ and $\mathbf{y}$ is exactly equal to $\mathbf{u}_i^\top M \mathbf{v}_j$ where $\mathbf{x} = \mathbf{u}_i$ (i.e., $\mathbf{x}$ is the i-th person) and $\mathbf{y} = \mathbf{v}_j$. Thus, the (i, j)-th entry of the matrix M can be assimilated to the value of the bilinear form defined by the matrix M over the pair $(\mathbf{u}_i, \mathbf{v}_j)$. Moreover the spectral regularizer corresponds to the corresponding spectral function of the complete matrix $M \in \mathbb{R}^{n_{\mathcal{X}} \times n_{\mathcal{Y}}}$.

In this context, finding a low-rank approximation of the observed entries in a matrix is an appealing strategy, which corresponds to taking the rank penalty constraint (4) combined with, for example, the square loss error. This however leads to non-convex optimization problems with multiple local minima, for which only local search heuristics are known (Srebro and Jaakkola, 2003). To circumvent this issue convex spectral penalty functions can be considered. For example, in the case of binary preferences, combining the hinge loss function with the trace norm penalty (5) leads to the maximum margin matrix factorization (MMMF) approach proposed by Srebro et al. (2005), which can be rewritten as a semi-definite program. For the sake of efficiency, Rennie and Srebro (2005) proposed to add a constraint on the rank of the matrix, resulting in a non-convex problem that can nevertheless be handled efficiently by classical gradient descent techniques; in our setting, this corresponds to changing the trace norm penalty (5) by the penalty (7).

3.2 Multi-task learning

It may be the case that we have attributes only for objects $\mathbf{y}$ (we could do the same for attributes for users). In that case, for a finite number of users $\{\mathbf{x}_i : i = 1, \ldots, N\}$ organized in $n_{\mathcal{X}}$ groups, we aim to estimate a separate function on objects $f_i(\mathbf{y})$ for each of the $n_{\mathcal{X}}$ users i. Considering the estimation of each of these f_i's as a *learning task*, one can possibly learn all f_i's *simultaneously* using a *mutli-task learning* approach.

In order to adapt our general framework to this scenario, it is natural to consider the attribute kernel $k_A^{\mathcal{Y}}$ for the objects, whose attributes are available, and the Dirac kernel $k_D^{\mathcal{X}}$ for the users, for which no attributes are used. Again the choice of the loss function depends on the precise task to be solved, and the spectral penalty function can be tuned to enforce some sharing of information between different tasks.

In particular, taking the rank penalty function (4) enforces a decomposition of the tasks (learning each f_i) into a limited number of factors. This results in a method for multitask learning, based on a low-rank representation of the predictor functions f_i. The resulting problem, however, is not convex due to the use of the non-convex rank penalty function. A natural alternative is then to replace the rank constraint by the trace norm penalty function (5), resulting in a convex optimization problem when the loss function is convex. Recently, a similar approach was independently proposed by Amit et al. (2007) in the context of multiclass classification and by Argyriou et al. (2007) for multitask learning.

Alternatively, another strategy to enforce some constraints among the tasks is to constrain the variance of the different classifiers. Evgeniou et al. (2005) showed that this strategy can be formulated in the framework of support vector machines by considering a *multitask kernel* between different objects and tasks. This strategy is also a particular case of our general framework, where the spectral penalty function is taken to be the Hilbert-Schmidt norm (6), the kernel between objects is the attribute kernel, and the kernel between tasks is the Dirac kernel plus a strictly positive constant to be learned. Replacing the Hilbert-Schmidt norm by other penalties such as the trace norm penalty (5) would result in new algorithms for multi-task learning, that constrain both the variance of the classifiers f_i and their decomposition into a small number of factors.

3.3 Pairwise learning

When attributes are available for both users and objects, then it is possible to take the attributes kernels for both of them. Combining this choice with the Hilbert-Schmidt penalty (6) results in classical machine learning algorithms (e.g., SVM if the hinge loss is taken as a loss function) applied to the *tensor product* of $\mathcal{X}$ and $\mathcal{Y}$. This strategy is a classical approach to learn a function over pairs of points (see, e.g., Jacob and Vert, 2008). Replacing the Hilbert-Schmidt norm by another spectral penalty function, such as the trace norm, would result in new algorithms for learning low-rank functions over pairs.

3.4 Combining the attribute and Dirac kernels

As illustrated in the previous subsections, the setting of the application often determines the combination of kernels to be used for the users and the objects: typically, two Dirac kernels for the standard CF setting without attributes, one Dirac and one attributes kernel for multi-task problems, and two attributes kernels when attributes are available for both users and objects and one wishes to learn over pairs.

There are many situations, however, where the attributes available to describe the users and/or objects are certainly useful for the inference task, but on the other hand do not fully characterize the users and/or objects. For example, if we just know the age and gender of users, we would like to use this information to model their preferences, but would also like to allow different preferences for different users even when they share the same age and gender. In our setting, this means that we may want to use the attributes kernel in order to use the users/objects' attributes during inference, but also the Dirac kernel to model the fact that different users and/or objects remain different even when they share the same attributes.

This naturally leads us to consider the following convex combinations of Dirac and attributes kernels:

$$\begin{cases} k^{\mathcal{X}} = \eta k_{\mathrm{A}}^{\mathcal{X}} + (1-\eta) k_{\mathrm{D}}^{\mathcal{X}}, \\ k^{\mathcal{Y}} = \zeta k_{\mathrm{A}}^{\mathcal{Y}} + (1-\zeta) k_{\mathrm{D}}^{\mathcal{Y}}, \end{cases} \tag{10}$$

where $0 \leq \eta \leq 1$ and $0 \leq \zeta \leq 1$. These kernels interpolate between the Dirac kernels ($\eta = 0$ and $\zeta = 0$) and the attributes kernels ($\eta = 1$ and $\zeta = 1$). Combining this choice of kernels with, e.g., the trace norm penalty function (5), allows us to continuously interpolate between different settings corresponding to different "corners" in the (η, ζ) square: standard CF with matrix completion in $(0, 0)$, multi-task learning in $(0, 1)$ and $(1, 0)$, and learning over pairs in $(1, 1)$. The extra degree of freedom created when η and ζ are allowed to vary continuously between 0 and 1 provides a principled way to optimally balance the influence of the attributes in the function estimation process.

3.5 Generalization to new points

A drawback of collaborative filtering is the impossibility to generalize to unseen data points (i.e., a new movie or a new person in the context of movie recommendations). When

attributes are used, a prediction based on those can be made, and thus using attributes has an added benefit beyond better performance.

4. Representer theorems

We now discuss how the general optimization problem (9) can be solved in practice. A first difficulty with this problem is that the optimization space $\{F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X}) : \Omega(F) < \infty\}$ can be of infinite dimension. We note that this can occur even under a rank constraint, because the set $\{F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X}) : \mathrm{rank}(F) \leq R\}$ is not included into any finite-dimensional linear subspace if $\mathcal{X}$ and $\mathcal{Y}$ have infinite dimensions. In this section, we show that the optimization problem (9) can be rephrased as a finite-dimensional problem, and propose practical algorithm to solve it in Section 5. While the reformulation of the problem as a finite-dimensional problem is a simple instance of the representer theorem when the Hilbert-Schmidt norm is used as a penalty function (Section 4.1), we prove in Section 4.2 a generalized representer theorem that is valid with any spectral penalty function.

4.1 The case of the Hilbert-Schmidt penalty function

In the particular case where the penalty function $\Omega(F)$ is the Hilbert-Schmidt norm (6), then the set $\{F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X}) : \Omega(F) < \infty\}$ is the set of Hilbert-Schmidt operators. As recalled in Appendix A. this set is a Hilbert space isometric through (1) to the reproducing kernel Hilbert space $\mathcal{H}_\otimes$ of the kernel:

$$k_\otimes\left(\left(\mathbf{x}, \mathbf{x}'\right), \left(\mathbf{y}, \mathbf{y}'\right)\right) = \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathcal{X}} \left\langle \mathbf{y}, \mathbf{y}' \right\rangle_{\mathcal{Y}},$$

and the isometry translates from F to f as:

$$\| f \|^2_{\mathcal{H}_\otimes} = \| F \|^2 = \Omega(F).$$

As a result, in that case the problem (9) is equivalent to:

$$\min_{f \in \mathcal{H}_\otimes} \left\{ R_N(f) + \lambda \| f \|^2_\otimes \right\}. \tag{11}$$

In that case the representer theorem for optimization of empirical risks penalized by the RKHS norm (Schölkopf et al., 2001) can be applied to show that the solution of (11) necessarily lives in the linear span of the training data. With our notations this translates into the following result:

Theorem 2 *If $\hat{F}$ is a solution of the problem:*

$$\min_{F \in \mathcal{B}_2(\mathcal{Y}, \mathcal{X})} \left\{ R_N(F) + \lambda \sum_{i=1}^{\infty} \sigma_i(F)^2 \right\}, \tag{12}$$

then it is necessarily in the linear span of $\{\mathbf{x}_i \otimes \mathbf{y}_i : i = 1, \ldots, N\}$, i.e., it can be written as:

$$\hat{F} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \otimes \mathbf{y}_i,$$

for some $\alpha \in \mathbb{R}^N$.

For the sake of completeness, and to highlight why this result is specific to the Hilbert-Schmidt penalty function (6), we rephrase here, with our notations, the main arguments in the proof of Schölkopf et al. (2001). Any operator F in $\mathcal{B}_2(\mathcal{Y}, \mathcal{X})$ can be decomposed as $F = F_S + F_\perp$, where F_S is the projection of F onto the linear span of $\{\mathbf{x}_i \otimes \mathbf{y}_i : i = 1, \ldots, N\}$. $F_\perp$ being orthogonal to each $\mathbf{x}_i \otimes \mathbf{y}_i$ in the training set, one easily gets $R_N(F) = R_N(F_S)$, while $\| F \|^2 = \| F_S \|^2 + \| F_\perp \|^2$ by the Pythagorean theorem. As a result a minimizer F of the objective function must be such that $F_\perp = 0$, i.e., must be in the linear span of the training tensor products.

4.2 A Representer Theorem for General Spectral Penalty Functions

Let us now move on to the more general situation (9) where a general spectral function $\Omega(F)$ is used as regularization. Theorem 2 is usually not valid in such a case. Its proof breaks down because it is not true that $\Omega(F) = \Omega(F_S) + \Omega(F_\perp)$ for general Ω, or even that $\Omega(F) \geq \Omega(F_S)$.

The following theorem, whose proof is postponed to Appendix B, can be seen as a generalized representer theorem. It shows that a solution of (9), if it exists, can be expanded over a finite basis of dimension $m_\mathcal{X} \times m_\mathcal{Y}$ (where $m_\mathcal{X}$ and $m_\mathcal{Y}$ are the underlying dimensions of the subspaces where the data lie), and that it can be found as the solution of a finite-dimensional optimization problem:

Theorem 3 *For any spectral penalty function $\Omega : \mathcal{B}_0(\mathcal{Y}, \mathcal{X}) \mapsto \mathbb{R} \cup \{+\infty\}$, let the optimization problem:*

$$\min_{F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X}),\ \Omega(F) < \infty} \{R_N(F) + \lambda \Omega(F)\} \,. \tag{13}$$

If the set of solutions is not empty, then there is a solution F in $\mathcal{X}_N \otimes \mathcal{Y}_N$, i.e., there exists $\alpha \in \mathbb{R}^{m_\mathcal{X} \times m_\mathcal{Y}}$ such that:

$$F = \sum_{i=1}^{m_\mathcal{X}} \sum_{j=1}^{m_\mathcal{Y}} \alpha_{ij} \mathbf{u}_i \otimes \mathbf{v}_j \,, \tag{14}$$

where $(\mathbf{u}_1, \ldots, \mathbf{u}_{m_\mathcal{X}})$ and $\left(\mathbf{v}_1, \ldots, \mathbf{v}_{m_\mathcal{Y}}\right)$ form orthonormal bases of $\mathcal{X}_N$ and $\mathcal{Y}_N$, respectively. Moreover, in that case the coefficients α can be found by solving the following finite-dimensional optimization problem:

$$\min_{\alpha \in \mathbb{R}^{m_\mathcal{X} \times m_\mathcal{Y}},\ \Omega(\alpha) < \infty} R_N\left(\operatorname{diag}\left(X \alpha Y^\top\right)\right) + \lambda \Omega(\alpha) \,, \tag{15}$$

where $\Omega(\alpha)$ refers to the spectral penalty function applied to the matrix α seen as an operator from $\mathbb{R}^{m_\mathcal{Y}}$ to $\mathbb{R}^{m_\mathcal{X}}$, and $X \in \mathbb{R}^{N \times m_\mathcal{X}}$ and $Y \in \mathbb{R}^{N \times m_\mathcal{X}}$ denote any matrices that satisfy $K = XX^\top$ and $G = YY^\top$ for the two $N \times N$ Gram matrices K and G defined by $K_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle_\mathcal{X}$ and $G_{ij} = \langle \mathbf{y}_i, \mathbf{y}_j \rangle_\mathcal{Y}$, for $0 \leq i, j \leq N$.

This theorem shows that, as soon as a spectral penalty function is used to control the complexity of the compact operators, a solution can be searched in the finite-dimensional space $\mathcal{X}_N \otimes \mathcal{Y}_N$, which in practice boils down to an optimization problem over the set of matrices of size $m_\mathcal{X} \times m_\mathcal{Y}$. The dimension of this space might however be prohibitively large for real-world application where, e.g., tens of thousands of users are confronted to

a database of thousands of objects. A convenient way to obtain an important decrease in complexity (at the expense of possibly losing convexity) is by constraining the rank of the operator through an adequate choice of a spectral penalty. Indeed, the set of non-zero singular components of F as an operator is equal to the set of non-zero singular values of α in (14) seen as a matrix. Consequently any constraint on the rank of F as an operator results in a constraint on α as a matrix, from which we deduce:

Corollary 4 *If, in Theorem 3, the spectral penalty function Ω is infinite on operators of rank larger than R (i.e., $\sigma_{R+1}(u) = +\infty$ for $u > 0$), then the matrix $\alpha \in \mathbb{R}^{m_\mathcal{X} \times m_\mathcal{Y}}$ in (14) has rank at most R.*

As a result, if a rank constraint $\operatorname{rank}(F) \leq r$ is added to the optimization problem then the representer theorem still holds but the dimension of the parameter α becomes $r \times (m_\mathcal{X} + m_\mathcal{Y})$ instead of $m_\mathcal{X} \times m_\mathcal{Y}$. We note that when the Hilbert-Schmidt norm is used, the rank constraint destroys the particular expansion available from Theorem 2.

5. Algorithms

In this section we explain how the optimization problem (15) can be solved in practice. We first consider a general formulation, then we specialize to the situation where many $\mathbf{x}$'s and many $\mathbf{y}$'s are identical; i.e., we are in a matrix completion setting where it may be advantageous to consider other formulations that take into account the group structure explicitly.

5.1 Convex dual of spectral regularization

For all $i = 1, \ldots, N$, we let denote $\psi_i(v_i) = \ell(v_i, t_i)$ the loss corresponding to predicting v_i for the i-th data point. For simplicity, we assume that each ψ_i is convex (this is usually met in practice). Following Bach et al. (2004b), we let $\psi_i^*(\alpha_i)$ denote its Fenchel conjugate defined as $\psi_i^*(\alpha_i) = \max_{v_i \in \mathbb{R}} \alpha_i v_i - \psi_i(v_i)$. Minimizers of the optimization problem defining the conjugate function are often referred to as Fenchel duals to α_i (Boyd and Vandenberghe, 2003). In particular, we have the following classical examples:

- *Least-squares regression*: we have $\psi_i(v_i) = \frac{1}{2}(t_i - v_i)^2$ and $\psi_i^*(\alpha_i) = \frac{1}{2}\alpha_i^2 + \alpha_i t_i$
- *logistic regression*: we have $\psi_i(v_i) = \log(1 + \exp(-y_i u_i))$, where $y_i \in \{-1, 1\}$, and $\psi_i^*(\alpha_i) = (1 + \alpha_i t_i)\log(1 + \alpha_i t_i) - \alpha_i t_i \log(-\alpha_i t_i)$ if $\alpha_i t_i \in (-1, 0)$, $+\infty$ otherwise.

We also assume that the spectral regularization is such that for all $i \in \mathbb{N}$, $s_i = s$, where s is a convex function such that $s(0) = 0$. In this situation, we have $\Omega(A) = \sum_{i \in \mathbb{N}} s(\sigma_i(A))$. We can also define a Fenchel conjugate for $\Omega(A)$, which is also a spectral function $\Omega^*(B) = \sum_{i \in \mathbb{N}} s^*(\sigma_i(B))$ (Lewis and Sendov, 2002).

Some special cases of interest for $s(\sigma)$ are:

- $s(\sigma) = |\sigma|$ leads to the trace norm and then $s^*(\tau) = 0$ if $|\tau|$ is less than 1, and $+\infty$ otherwise.

- $s(\sigma) = \frac{1}{2}\sigma^2$ leads to the Frobenius/Hilbert Schmidt norm and then $s^*(\tau) = \frac{1}{2}\tau^2$.
- $s(\sigma) = \varepsilon \log(1+e^{\sigma/\varepsilon}) + \varepsilon \log(1+e^{-\sigma/\varepsilon})$ is a smooth approximation of $|\sigma|$, which becomes tighter when ε is closer to zero. We have: $s^*(\tau) = \frac{1}{\varepsilon}(1+\tau)\log(1+\tau) + \frac{1}{\varepsilon}(1-\tau)\log(1-\tau)$. Moreover, $s'(\sigma) = \tau \Leftrightarrow (s^*)'(\tau) = \sigma = \frac{1}{\varepsilon}\log\frac{1+\tau}{1-\tau}$.

Once the representer theorem has been applied, our optimization problem can be rewritten in the *primal* form in (15):

$$\min_{\alpha \in \mathbb{R}^{m_x \times m_y}} \sum_{i=1}^{N} \psi_i((X\alpha Y^\top)_{ii}) + \lambda\Omega(\alpha). \tag{16}$$

We can now form the Lagrangian, associated with added constraints $v = \operatorname{diag}(X\alpha Y^\top)$ and corresponding Lagrange multiplier $\beta \in \mathbb{R}^N$:

$$\mathcal{L}(v, \alpha, \beta) = \sum_{i=1}^{N} \psi_i(v_i) - \sum_{i=1}^{N} \beta_i(v_i - (X\alpha Y^\top)_{ii}) + \lambda\Omega(\alpha),$$

and minimize with respect to v and W to obtain the *dual* problem, which is to maximize:

$$-\sum_{i=1}^{N} \psi_i^*(\beta_i) - \lambda\Omega^*\left(-\frac{1}{\lambda}X^\top \operatorname{Diag}(\beta) Y\right). \tag{17}$$

Once the optimal dual variable β is found (there are as many of those as there are observations), then we can go back to α (which may or may not be of smaller size), by Fenchel duality, i.e., α is among the Fenchel duals of $-\frac{1}{\lambda}X^\top \operatorname{Diag}(\beta)Y$. Note that when the function s is differentiable, then we have α in closed form from β and vice-versa.

Note that for optimization, we have two strategies: using the primal problem in Eq. (16) of dimension $m_{\mathcal{X}} m_{\mathcal{Y}} \leqslant n_{\mathcal{X}} n_{\mathcal{Y}}$ (the actual dimension of the underlying data) or using the dual problem in Eq. (17) of dimension N (the number of ratings). The choice between those two formulations is problem dependent.

5.2 Collaborative filtering

In the presence of (many) identical columns and row, which is standard in collaborative filtering situations, the kernel matrices K and L have some columns which are identical, and we can instead consider the kernel matrices (with their square-root decompositions) $\tilde{K} = \tilde{X}\tilde{X}^\top$ and $\tilde{L} = \tilde{Y}\tilde{Y}^\top$ as the kernel matrices for all distincts element of $\mathcal{X}$ and $\mathcal{Y}$ (let $n_{\mathcal{X}}$ and $n_{\mathcal{Y}}$ be their sizes). Then each observation $(\mathbf{x}_i, \mathbf{y}_i, t_i)$ corresponds to a pair of indices $(a(i), b(i))$ in $\{1, \dots, n_{\mathcal{X}}\} \times \{1, \dots, n_{\mathcal{Y}}\}$, and the primal/dual problems become:

$$\min_{\alpha \in \mathbb{R}^{m_x \times m_y}} \sum_{i=1}^{n} \psi_i(\delta_{a(i)}^\top \tilde{X}\alpha\tilde{Y}^\top \delta_{b(i)}) + \lambda\Omega(\alpha), \tag{18}$$

where δ_u is a vector with only zeroes except at position u. The dual function is

$$-\sum_{i=1}^{N} \psi_i^*(\beta_i) - \lambda\Omega^*\left(-\frac{1}{\lambda}\tilde{X}^\top \sum_{i=1}^{N} \beta_i \delta_{a(i)} \delta_{b(i)}^\top \tilde{Y}\right).$$

Similarly to usual kernel machines and the general case presented above, using the primal or the dual formulation for optimization depends on the number of available ratings N compared to the ranks $m_{\mathcal{X}}$ and $m_{\mathcal{Y}}$ of the kernel matrices $\tilde{K}$ and $\tilde{L}$. Indeed, the number of variables in the primal formulation is $m_{\mathcal{X}} m_{\mathcal{Y}}$, while in the dual formulation it is N.

5.3 Low rank constrained problem

We approximate the spectral norm by a infinitely differentiable spectral function. Since we consider in this paper infinitely differentiable loss functions, our problem is that of minimizing an infinitely differentiable convex function $G(W)$ over rectangular matrices of size $p \times q$ for certain integers p and q. Because of our spectral regularization, we expect to obtain (potentially approximately) low-rank matrices. In this context, it has proved advantageous to consider low-rank decompositions of the form $W = UV^\top$ where U and V have $m < \min\{p, q\}$ columns (Burer and Monteiro, 2005, Burer and Choi, 2006). Burer and Monteiro (2005) have shown that if $m = \min\{p, q\}$ then the non convex problem of minimizing $G(UV^\top)$ with respect to U and $V^\top$ has no local minima.

We now prove a stronger result in the context of twice differentiable functions, namely that if the global optimum of G has rank $r < \min\{p, q\}$, then the low-rank constrained problem with rank $r + 1$ has no local minimum and its global minimum corresponds to the global minimum of G. The following theorem makes this precise (see Appendix C for proof).

Proposition 5 *Let G be a twice differentiable convex function on matrices of size $p \times q$ with compact level sets. Let $m > 1$ and $(U, V) \in \mathbb{R}^{p \times m} \times \mathbb{R}^{q \times m}$ a local optimum of the function $H : \mathbb{R}^{p \times m} \times \mathbb{R}^{q \times m} \mapsto \mathbb{R}$ defined by $H(U, V) = G(UV^\top)$, i.e., U is such that $\nabla H(U, V) = 0$ and the Hessian of H at (U, V) is positive semi-definite. If U or V is rank deficient, then $N = UV^\top$ is a global minimum of G, i.e., $\nabla G(N) = 0$.*

The previous proposition shows that if we have a local minimum for the rank-m problem and if the solution is rank deficient, then we have a solution of the global optimization problem. This naturally leads to a sequence of reduced problems of increasing dimension m, smaller than $r + 1$, where r is the rank of the global optimum. However, the number of iterations of each of the local minimizations and the final rank m cannot be bound a priori in general.

Note that using a low-rank representation to solve the trace-norm regularized problem leads to a non convex minimization problem with no local minima, while simply using the low-rank representation *without* the trace norm penalty and potentially with a Frobenius norm penalty, may lead to local minima; i.e., we consider instead of Eq. (15) with the trace norm, the following formulation:

$$\min_{\alpha \in \mathbb{R}^{m_{\mathcal{X}} \times r}, \beta \in \mathbb{R}^{m_{\mathcal{Y}} \times r}, \Omega(\alpha) < \infty} R_N\left(diag\left(X \alpha \beta^\top Y^\top\right)\right) + \lambda \sum_{q=1}^{r} \|\alpha(:,k)\|^2 \|\beta(:,k)\|^2 , \qquad (19)$$

In the simulation section, we compare the two approaches on a synthetic example, and show that the convex formulation solved through a sequence of non convex formulations leads to better predictive performance.

5.4 Kernel learning for spectral functions

In our collaborative filtering context, there are two potentially useful sources of kernel learning: learning the attribute kernels, or learning the weights η and ζ between Dirac kernels and attribute kernels. (XXX THIS SENTENCE IS STRANGE We follow the approach of Bach et al. (2004a), which penalizes by sums of block norms We assume that we have M kernel matrices $K_1, \ldots, K_M$ for $\mathcal{X}$ and M kernel matrices $G_1, \ldots, G_M$ for $\mathcal{Y}$, together with their square roots $X_1, \ldots, X_M$ and $Y_1, \ldots, Y_M$. We look for predictor functions which are sums of the M possible atomic predictor functions, and we penalize by the sum of spectral functions, to obtain the following optimization problem:

$$\min_{\forall k, \alpha_k \in \mathbb{R}^{m_x^k \times m_y^k}} \sum_{i=1}^{n} \psi_i \left(\sum_{k=1}^{M} (X_k \alpha_k Y_k^\top)_{ii} \right) + \lambda \sum_{k=1}^{M} \Omega(\alpha_k).$$

We form the Lagrangian:

$$\mathcal{L}(v, \alpha_1, \ldots, \alpha_M, \beta) = \sum_{i=1}^{n} \psi_i(v_i) - \sum_{i=1}^{N} \beta_i (v_i - \sum_{k=1}^{M} (X \alpha_k Y^\top)_{ii}) + \lambda \sum_{k=1}^{M} \Omega(\alpha_k),$$

and minimize w.r.t. v and $\alpha_1, \ldots, \alpha_M$ to obtain the dual problem, which is to maximize

$$- \sum_{u} \psi_i^*(\beta_i) - \sum_{k} \lambda \Omega^* \left(-\frac{1}{\lambda} X_k^\top \operatorname{Diag}(\beta) Y_k \right). \tag{20}$$

In the case of the trace norm, we obtain support kernels (Bach et al., 2004a). In the dual formulation, there is only one α to optimize, and thus it is preferable to use the dual formulation rather than the primal formulation. We show in Section 6 how the framework can be used to automatically combine the four corners presented in Section 3.4.

6. Experiments

In this Section we present several experimental findings for the algorithms and methods discussed above. Much of the present work was motivated by the problem of Collaborative Filtering and we therefore focus primarily within this domain. As discussed in Section 3, by using operator estimation and spectral regularization as a framework for CF, we may utilize potentially more information to predict preferences. Our primary goal now is to show that, as one would hope, such capabilities do improve prediction accuracy.

6.1 Synthetic examples

We have first run a number of experiments on a synthetic dataset. We generated data as follows: (1) generate i.i.d. multivariate features for x, (2) generate i.i.d. multivariate features for y, (3) sample z from a random bilinear form in x and y plus some noise, (4) discard half of the features. Since we discard some of the features, the label cannot be entirely predicted by the known features. On the other hand, since we keep some of them, knowing and using these attributes should work better than not using them. In other

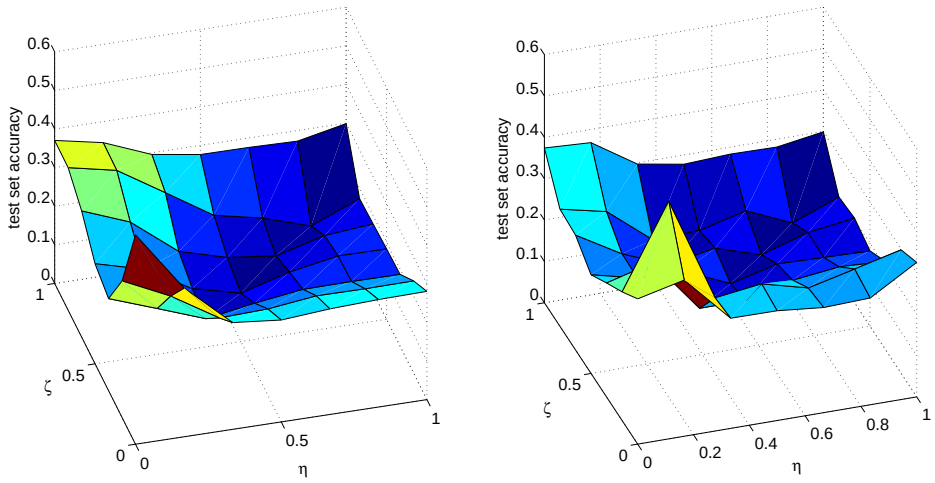

Figure 1: Comparing trace norm (left) and rank constraints (right) for various values η and ζ using the simulated dataset: plot of the test set accuracy (after all hyperparameters are optimized upon). The minimal value achieved by the trace norm is 0.1222 and the one achieved by the rank constraint is 0.1540.

words, we expect that setting η and ζ to be values other than 0 or 1 should provide better performance. All test set accuracies are measured as the root mean squared error averaged over 10-fold cross validations.

6.1.1 Tracenorm Versus Low-rank

In Figure 1, we compare the tracenorm formulation (no local minima, optimized trough low rank decompositions) to the low rank constraints, showing that the trace norm indeed performs better (better MSE for the trace norm than for the low-rank+Frobenius penalty). Moreover, best predictive performance is achieved in both cases in the middle of the square and not at any of the four corners.

6.1.2 Kernel Learning

In Figure 2, we show the test set accuracy as a function of the regularization parameter, when we use the kernels corresponding to the four corners as the four basis kernels. We can see that we recover similar performance than by searching over all η and ζ's. The same algorithm could also be used to learn kernels on the attributes.

6.2 Performance on MovieLens Data

The main "real-world" dataset we worked with is the well-known MovieLens 100k dataset from the GroupLens Research Group at the University of Minnesota. This dataset consists

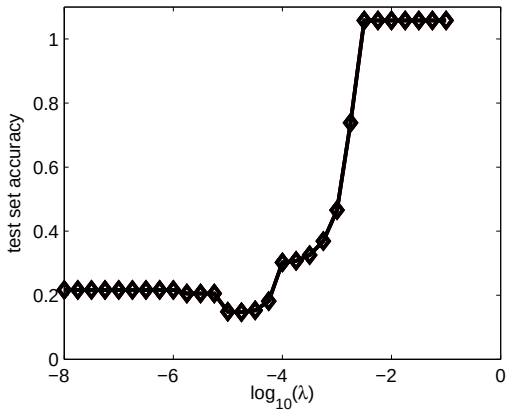

Figure 2: Learning the kernel: test set accuracy vs. regularization parameter. Minimum value is 0.14.

of ratings of 1682 movies by 943 users. Each user provided a rating, in the form of a score from $\{1, 2, 3, 4, 5\}$, for a small subset of the movies. Each user rated at least 20 movies, and the total number of ratings available is exactly 100,000, averaging about 105 per user. This dataset was rather appropriate as it included attribute information for both the movies and the users. Each movie was labeled with at least one among 19 genres (e.g., action or adventure), while the users' attributes included age, gender, and an occupation among a list of 21 occupations (e.g., administrator or artist). We converted the users age attribute to a set of binary features that describes to which of 5 age categories the user belongs.

In Figure 3 we see predictive accuracy in RMSE on the MovieLens dataset, obtained by 10-fold cross-validation. The heat plot provides some insight on the relative value, for both movies and users, of the given attribute kernels versus the simple identity kernels. The corners have higher values than some of the values inside the square, showing that the best balance between attribute and Dirac kernels is achieved for $\eta, \zeta \in (0, 1)$.

7. Conclusions

We presented a method for solving a generalized matrix completion problem where we have attributes describing the matrix dimensions. The problem is formalized as the problem of inferring a linear compact operator between two general Hilbert spaces, which generalizes the classical finite-dimensional matrix completion problem. We introduced the notion of spectral regularization for operators, which generalized various spectral penalizations for matrices, and proved a general representer theorem for this setting. Various approaches, such as standard low rank matrix completion, are special cases of our method. It is particularly relevant for CF applications where attributes are available for users and/or objects, and preliminary experiments confirm the benefits of our method.

An interesting direction of future research is to explore further the multi-task learning algorithm we obtained with low-rank constraint, and to study the possibility to derive

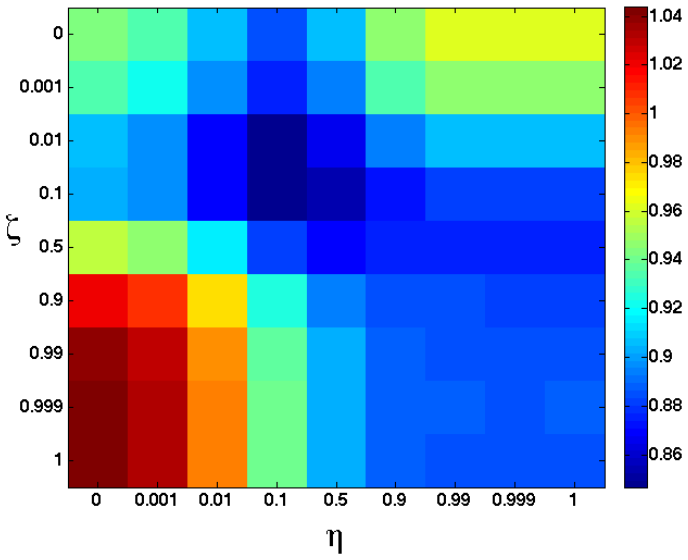

Figure 3: A heat plot of performance for a range of kernel parameter choices, η, ζ, using the MovieLens dataset.

on-line implementations that may better fit the need for large-scale applications where training data are continuously increasing. On the theoretical side, a better understanding of the effects of norm and rank regularizations and their interaction would be helpful.

Appendix A. Compact operators on Hilbert spaces

In this appendix we recall basic definitions and properties of Hilbert space operators. We refer the interested reader to general books (Brezis, 1980) for more details.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two Hilbert spaces, with respective inner products denoted by $\langle \mathbf{x}, \mathbf{x}' \rangle_{\mathcal{X}}$ and $\langle \mathbf{y}, \mathbf{y}' \rangle_{\mathcal{Y}}$ for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$. We denote by $\mathcal{B}(\mathcal{Y}, \mathcal{X})$ the set of bounded operators from $\mathcal{X}$ to $\mathcal{Y}$, i.e., of continuous linear mappings from $\mathcal{Y}$ to $\mathcal{X}$. For any two elements $(\mathbf{x}, \mathbf{y})$ in $\mathcal{X} \times \mathcal{Y}$, we denote by $\mathbf{x} \otimes \mathbf{y}$ their *tensor product*, i.e., the linear operator from $\mathcal{Y}$ to $\mathcal{X}$ defined by:

$$\forall \mathbf{h} \in \mathcal{Y}, \quad (\mathbf{x} \otimes \mathbf{y})\, \mathbf{h} = \langle \mathbf{y}, \mathbf{h} \rangle_{\mathcal{Y}}\, \mathbf{x}\,. \tag{21}$$

We denote by $\mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ the set of *compact* linear operators from $\mathcal{Y}$ to $\mathcal{X}$, i.e., the set of linear operators that map the unit ball of $\mathcal{Y}$ to a relatively compact set of $\mathcal{X}$.

When $\mathcal{X}$ and $\mathcal{Y}$ have finite dimensions, then $\mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ is simply the set of linear mappings from $\mathcal{Y}$ to $\mathcal{X}$, which can be represented by the set of matrices of dimensions $\dim(\mathcal{X}) \times \dim(\mathcal{Y})$. In that case the tensor product $x \otimes y$ is represented by the matrix $xy^\top$, where $y^\top$ denotes the transpose of y.

For general Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$, any linear operator $F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ admits a *spectral decomposition*:

$$F = \sum_{i=1}^{\infty} \sigma_i \mathbf{u}_i \otimes \mathbf{v}_i \,. \tag{22}$$

Here the the *singular values* $(\sigma_i)_{i\in\mathbb{N}}$ form a sequence of non-negative real numbers such that $\lim_{i\to\infty} \sigma_i = 0$, and $(\mathbf{u}_i)_{i\in\mathbb{N}}$ and $(\mathbf{v}_i)_{i\in\mathbb{N}}$ form orthonormal families in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Although the vectors $(\mathbf{u}_i)_{i\in\mathbb{N}}$ and $(\mathbf{v}_i)_{i\in\mathbb{N}}$ in (22) are not uniquely defined for a given operator F, the set of singular values is uniquely defined. By convention we denote by $\sigma_1(F), \sigma_2(F), \ldots$, the successive singular values of F ranked by decreasing order. The *rank* of F is the number $\mathrm{rank}(F) \in \mathbb{N} \cup \{+\infty\}$ of strictly positive singular values.

We now describe three subclasses of compact operators of particular relevance in the rest of this paper.

- The set of operators with finite rank is denoted $\mathcal{B}_F(\mathcal{Y}, \mathcal{X})$.
- The operators $F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ that satisfy:

  $$\sum_{i=1}^{\infty} \sigma_i(F)^2 < \infty$$

  are called *Hilbert-Schmidt* operators. They form a Hilbert space, denoted $\mathcal{B}_2(\mathcal{Y}, \mathcal{X})$, with inner product $\langle \cdot, \cdot \rangle_{\mathcal{X}\otimes\mathcal{Y}}$ between basic tensor products given by:

  $$\left\langle \mathbf{x} \otimes \mathbf{y}, \mathbf{x}' \otimes \mathbf{y}' \right\rangle_{\mathcal{X}\otimes\mathcal{Y}} = \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathcal{X}} \left\langle \mathbf{y}, \mathbf{y}' \right\rangle_{\mathcal{Y}} \,. \tag{23}$$

  In particular the Hilbert-Schmidt norm of an operator in $\mathcal{B}_2(\mathcal{Y}, \mathcal{X})$ is given by:

  $$\| F \|_2 = \left(\sum_{i=1}^{\infty} \sigma_i(F)^2 \right)^{\frac{1}{2}} \,.$$

  Another useful characterization of Hilbert-Schmidt operators is the following. Each linear operator $F : \mathcal{Y} \to \mathcal{X}$ uniquely defines a bilinear function $f_H : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ by

  $$f(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, F\mathbf{y} \rangle_{\mathcal{X}} \,.$$

  The set of functions f_F associated to the Hilbert-Schmidt operators forms itself a Hilbert space of functions $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which is the reproducing kernel Hilbert space of the product kernel defined for $((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) \in (\mathcal{X} \times \mathcal{Y})^2$ by

  $$k_{\otimes}\left((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')\right) = \left\langle \mathbf{x}, \mathbf{x}' \right\rangle_{\mathcal{X}} \left\langle \mathbf{y}, \mathbf{y}' \right\rangle_{\mathcal{Y}} \,.$$

- The operators $F \in \mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ that satisfy:

  $$\sum_{i=1}^{\infty} \sigma_i(F) < \infty$$

are called *trace-class* operators. The set of trace-class operators is denoted $\mathcal{B}_1(\mathcal{Y},\mathcal{X})$. The *trace norm* of an operator $F \in \mathcal{B}_1(\mathcal{Y},\mathcal{X})$ is given by:

$$\| F \|_1 = \sum_{i=1}^{\infty} \sigma_i(F) .$$

Obviously the following ordering exists among these various classes of operators:

$$\mathcal{B}_F(\mathcal{Y},\mathcal{X}) \subset \mathcal{B}_1(\mathcal{Y},\mathcal{X}) \subset \mathcal{B}_2(\mathcal{Y},\mathcal{X}) \subset \mathcal{B}_0(\mathcal{Y},\mathcal{X}) \subset \mathcal{B}(\mathcal{Y},\mathcal{X}) ,$$

and all inclusions are equalities if $\mathcal{X}$ and $\mathcal{Y}$ have finite dimensions.

Appendix B. Proof of Theorem 3

We start with a general result about the decrease of singular values for compact operators composed with projection:

Lemma 6 *Let $\mathcal{G}$ and $\mathcal{H}$ be two Hilbert spaces, H a compact linear subspace of $\mathcal{H}$, and Π_H denote the orthogonal projection onto H. Then for any compact operator $F : \mathcal{G} \mapsto \mathcal{H}$ it holds that:*

$$\forall i \geq 1 , \quad \sigma_i(\Pi_H F) \leq \sigma_i(F) .$$

Proof We use the classical characterization of the i-th singular value:

$$\sigma_i(F) = \max_{V \in \mathcal{V}_i(\mathcal{G})} \min_{\mathbf{x} \in V, \| \mathbf{x} \|_{\mathcal{G}} = 1} \| F\mathbf{x} \|_{\mathcal{H}} ,$$

where $\mathcal{V}_i(\mathcal{G})$ denotes the set of all linear subspaces of $\mathcal{G}$ of dimension i. Now, observing that for any $\mathbf{x}$ we have $\| \Pi_H F\mathbf{x} \|_{\mathcal{H}} \leq \| F\mathbf{x} \|_{\mathcal{H}}$ proves the Lemma. ■

Given a training set of patterns $(\mathbf{x}_i, \mathbf{y}_i)_{i=1,\ldots,N} \in \mathcal{X} \times \mathcal{Y}$, remember that we denote by $\mathcal{X}_N$ and $\mathcal{Y}_N$ the linear subspaces of $\mathcal{X}$ and $\mathcal{Y}$ spanned by the training patterns $\{\mathbf{x}_i : i = 1, \ldots, N\}$ and $\{\mathbf{y}_i : i = 1, \ldots, N\}$, respectively. For any operator $F \in \mathcal{B}_0(\mathcal{Y},\mathcal{X})$, let us now consider the operator $G = \Pi_{\mathcal{X}_N} F \Pi_{\mathcal{Y}_N}$. By construction, F and G agree on the training patterns, in the sense that for $i = 1, \ldots, N$:

$$\langle \mathbf{x}_i, G\mathbf{y}_i \rangle_{\mathcal{X}} = \langle \mathbf{x}_i, \Pi_{\mathcal{X}_N} F \Pi_{\mathcal{Y}_N} \mathbf{y}_i \rangle_{\mathcal{X}} = \langle \Pi_{\mathcal{X}_N} \mathbf{x}_i, F \Pi_{\mathcal{Y}_N} \mathbf{y}_i \rangle_{\mathcal{X}} = \langle \mathbf{x}_i, F\mathbf{y}_i \rangle_{\mathcal{X}} .$$

Therefore F and G have the same empirical risk:

$$R_N(F) = R_N(G) . \tag{24}$$

Now, by denoting F^* the adjoint operator, we can use Lemma 6 and the fact that the singular values of an operator and its adjoint are the same to obtain, for any $i \geq 1$:

$$\begin{aligned} \sigma_i(G) &= \sigma_i(\Pi_{\mathcal{X}_N} F \Pi_{\mathcal{Y}_N}) \\ &\leq \sigma_i(F \Pi_{\mathcal{Y}_N}) \\ &= \sigma_i(\Pi_{\mathcal{Y}_N} F^*) \\ &\leq \sigma_i(F^*) \\ &= \sigma_i(F). \end{aligned}$$

This implies that the spectral penalty term satisfies $\Omega(G) \leq \Omega(F)$. Combined with (24) this shows that if F is a solution to (13), then $G = \Pi_{\mathcal{X}_N} F \Pi_{\mathcal{Y}_N}$ is also a solution. Observing that $G \in \mathcal{X}_N \otimes \mathcal{Y}_N$ concludes the proof of the first part of Theorem 3, resulting in (14).

We have now reduced the optimization problem in $\mathcal{B}_0(\mathcal{Y}, \mathcal{X})$ to a finite-dimensional optimization over the matrix α of size $m_\mathcal{X} \times m_\mathcal{Y}$. Let us now rephrase the optimization problem in this finite-dimensional space.

Let us first consider the spectral penalty term $\Omega(F)$. Given the decomposition (14), the non-zero singular values of F as an operator are exactly the non-zero singular values of α as a matrix, as soon as $(\mathbf{u}_1, \ldots, \mathbf{u}_{m_\mathcal{X}})$ and $\left(\mathbf{v}_1, \ldots, \mathbf{v}_{m_\mathcal{Y}}\right)$ form *orthonormal* bases of $\mathcal{X}_N$ and $\mathcal{Y}_N$, respectively. In order to be able to express the empirical risk $R_N(F)$ we must however consider a decomposition of F over the training patterns, as:

$$F = \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_{ij} \mathbf{x}_i \otimes \mathbf{y}_j \,. \tag{25}$$

In order to express the singular values from this expression let us introduce the *Gram matrices* K and G of the training patterns, i.e., the $N \times N$ matrices defined for $i, j = 1, \ldots, N$ by:

$$K_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle_\mathcal{X} \,, \quad G_{ij} = \langle \mathbf{y}_i, \mathbf{y}_j \rangle_\mathcal{Y} \,.$$

We note that by definition the ranks of K and G are respectively $m_\mathcal{X}$ and $m_\mathcal{Y}$. Let us now factorize these two matrices as $K = XX^\top$ and $G = YY^\top$, where $X \in \mathbb{R}^{N \times m_\mathcal{X}}$ and $Y \in \mathbb{R}^{N \times m_\mathcal{Y}}$ are any square roots, e.g., obtained by kernel PCA or Cholesky decomposition (Fine and Scheinberg, 2001, Bach and Jordan, 2005). The matrices X and Y provide a representation of the pattern in two orthonormal bases which we denote by $(\mathbf{u}_1, \ldots, \mathbf{u}_{m_\mathcal{X}})$ and $\left(\mathbf{v}_1, \ldots, \mathbf{v}_{m_\mathcal{Y}}\right)$. In particular we have, for any $i, j \in 1, \ldots, N$:

$$\mathbf{x}_i \otimes \mathbf{y}_j = \sum_{l=1}^{m_\mathcal{X}} \sum_{m=1}^{m_\mathcal{Y}} X_{il} Y_{jm} \mathbf{u}_l \otimes \mathbf{v}_m \,,$$

from which we deduce:

$$F = \sum_{l=1}^{m_\mathcal{X}} \sum_{m=1}^{m_\mathcal{Y}} \left(\sum_{i=1}^{N} \sum_{j=1}^{N} X_{il} Y_{jm} \gamma_{ij} \right) \mathbf{u}_l \otimes \mathbf{v}_m \,.$$

Comparing this expression to (14) we deduce that:

$$\alpha = X^\top \gamma Y \,.$$

The empirical error $R_N(F)$ is a function of $f(\mathbf{x}_l, \mathbf{y}_l)$ for $l = 1, \ldots, N$. From (25) we see that:

$$f(\mathbf{x}_l, \mathbf{y}_l) = \sum_{i=1}^{N} \sum_{j=1}^{N} \gamma_{ij} K_{il} G_{lj} \,,$$

and therefore the vector of predictions $F_N = (f(\mathbf{x}_l, \mathbf{y}_l) \,:\, l = 1, \ldots, N) \in \mathbb{R}^N$ can be rewritten as:

$$F_N = \operatorname{diag}(K \gamma G) = \operatorname{diag}\left(X \alpha Y^\top\right) \,.$$

We can now replace the empirical risk $R_N(F_N)$ by $R_N\left(\operatorname{diag}\left(X\alpha Y^\top\right)\right)$ and the penalty $\Omega(F)$ by $\Omega(\alpha)$ to deduce the optimization problem (15) from (13), which concludes the proof of Theorem 3.

Appendix C. Proof of Proposition 5

Since the function has compact level sets, we may assume that we are restricted to an open bounded subset of $\mathbb{R}^{p\times q}$ where the second and first derivatives are uniformly bounded. We let denote $C > 0$ a common upper bound of all derivatives. The gradient of the function H is equal to $\nabla H = \binom{\nabla G^\top \ U}{\nabla G \ \ V}$, while the Hessian of H is the following quadratic form

$$\nabla^2 H((dU, dV), (dU, dV)) = 2 \operatorname{tr} dV^\top \nabla G dU + \nabla^2 G(U dV^\top + dU V^\top, U dV^\top + dU V^\top).$$

Without loss of generality, we may assume that the last columns of U and V are equal to zero (this can be done by rotation of U or V). The zero gradient assumption implies that $\nabla G^\top U = 0$ and $\nabla G V = 0$. While if we take dU and dV with the first $m-1$ columns equal to zero, and last columns equal to arbitrary u and v, then the second term in the Hessian is equal to zero. The positivity of the first term implies that for all u and v, $v^\top \nabla G u = 0$, i.e., the gradient of G at $N = UV^\top$ is equal to zero, and thus we get a stationary point and thus a global minimum of G.

References

Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the Twenty-fourth International Conference on Machine Learning*, 2007.

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Adv. Neural. Inform. Process Syst. 19*, pages 41–48, Cambridge, MA, 2007. MIT Press.

F. R. Bach. Consistency of trace norm minimization. Technical Report 00179522, HAL, 2007.

F. R. Bach and M. I. Jordan. Predictive low-rank decomposition for kernel methods. In *ICML*, 2005.

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proc. ICML*, 2004a.

F. R. Bach, R. Thibaux, and M. I. Jordan. Computing regularization paths for learning multiple kernels. In *Advances in Neural Information Processing Systems 17*, 2004b.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 2003.

John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, W.I., 1998. Morgan Kaufman.

H. Brezis. *Analyse Fonctionnelle*. Masson, 1980.

S. A. Burer and C. Choi. Computational enhancements in low-rank semidefinite programming. *Optimization Methods and Software*, 21:493–512, 2006.

S. A. Burer and R. D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103:427–444, 2005.

T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, 2005.

M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proc. American Control Conference*, volume 6, 2001.

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *J. Mach. Learn. Res.*, 2:243–264, 2001.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

L. Jacob and J.-P. Vert. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics*, 2008. To appear.

A. S. Lewis and H. S. Sendov. Twice differentiable spectral functions. *SIAM J. Mat. Anal. App.*, 23(2):368–386, 2002.

J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719, New York, NY, USA, 2005. ACM Press.

R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 791–798, New York, NY, USA, 2007. ACM.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, pages 416–426, 2001.

N. Srebro and T. Jaakkola. Weighted low-rank approximations. In T. Fawcett and N. Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Adv. Neural. Inform. Process Syst. 17*, pages 1329–1336, Cambridge, MA, 2005. MIT Press.