REPRODUCING KERNEL HILBERT SPACE METHODS FOR INFORMATION THEORETIC LEARNING

By

LUIS GONZALO SÁNCHEZ GIRALDO

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2012

© 2012 Luis Gonzalo Sánchez Giraldo

To my present and future family.

ACKNOWLEDGMENTS

I want to express my most sincere gratitude to my Ph.D. advisor Dr. José Príncipe for his enthusiastic and supporting attitude towards my work. I also want to say thanks for his patience and understanding that reflects the wisdom of a true mentor. I am very indebted with Professor Murali Rao, for his invaluable help, friendship, and love for mathematics. I can say that both Dr. Príncipe and Dr. Rao are examples of men with contagious passion for what they do. A Ph.D. is a journey full of ups and downs and I strongly believe that their excitement for my work helped me through the hard times. I also want to extend my gratitude to Dr. Tan Wong and Dr. Anand Rangarajan for their valuable comments and sharp remarks. They really help shape my thinking and enhance the perspective I have of my work. During this years, I had the great opportunity to be part of an amazing group of talented students, but most importantly valuable persons. I really enjoyed the time at CNEL and all the things we did as the *family away from family* that we are. Especial thanks to Erion Hasanbelliu for his friendship and disinterested help and support. Sohan Seth for his friendship, help, and the interesting discussions, Alexander Singh gracias por su amistad y ayuda en el momento oportuno, Shalom Darmajan, Il “Memming” Park, Stefan Cracium, Austin Brockmeier, Lin Li, Hector Galloza, Goktug Cinar, Rakesh Chalasani, Abhishek Singh, and all the CNELers for the good times.

Finally, I want to thank my family for always being there for me, and to God for bringing Jihye Bae into my life. Jihye nae sarang, I cannot ask for more when I already have it all with you. This work is as yours as it is mine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

REPRODUCING KERNEL HILBERT SPACE METHODS FOR INFORMATION THEORETIC LEARNING

By

Luis Gonzalo Sánchez Giraldo

December 2012

Chair: José C. Príncipe
Major: Electrical and Computer Engineering

Information theory provides principled models for different machine learning problems, such as clustering, dimensionality reduction, classification, and many others. However, using the definitions of information theoretic quantities directly, posses a challenging estimation problem that often leads to strong simplifications, such as Gaussian models, or the use of plug in density estimators that restrict the kind of representations that we can use on the data. In this work, we adopt a data-driven perspective based on reproducing kernel Hilbert space methods that leads to successful application of information theoretic principles without resorting to estimation of the underlying probability distributions. The proposed methodology offers several advantages compared to other state of the art work such as entropic graphs because it can provide quantities that are more amenable for optimization (differentiability) as well as the representation flexibility that kernel methods provide. The work is divided into two main parts. First, we introduce an information theoretic objective function for unsupervised learning called the principle of relevant information. We employ an information theoretic reproducing kernel Hilbert space (RKHS) formulation, which can overcome some of the limitations of previous approaches based on Parzen's density estimation. Results are competitive with kernel-based feature extractors such as kernel PCA. Moreover, the proposed framework goes further on the relation between information theoretic learning, kernel methods and support vector algorithms.

For the second part, which is motivated by the results and insights obtained by the RKHS formulation of the principle of relevant information, we develop a framework for information theoretic learning based on infinitely divisible matrices. We formulate an entropy-like functional on positive definite matrices based on Renyi's definition and examine some key properties of this functional that lead to the concept of infinite divisibility. This formulation avoids the plug in estimation of density and allows to use the representation power that comes with the use of positive definite kernels. Learning from data comes from using this functional on positive definite matrices that correspond to Gram matrices constructed by pairwise evaluations of infinitely divisible kernels on data samples. We show how we can define analogues to quantities such as conditional entropy that can be employed to formulate different learning problems, and provide efficient ways to solve the optimization that arise from these formulations. Numerical results using the proposed quantities to test independence, metric learning, and image super-resolution show that the proposed framework can obtain state of the art performances.

CHAPTER 1
INTRODUCTION

1.1 The Problem of Learning

In a system, learning is the property to incorporate external information to improve its performance in a particular task. In this sense the system tries to infer general rules from a set of given examples. The rules can be in the form of a function that model the dependence relation between input output pairs or can consist of some observation about the structure of the spaces where the examples are represented. The information provided to the system mainly comes from a set of observed inputs $\{x_i\} \subset \mathcal{X}$, although other sources of information may be given depending upon the task and the context in which the system is put. Usually three scenarios can be described. The *supervised* setting, where the task correspond to finding a rule of association between pairs $\{(x_i, y_i)\}$ of observed inputs and corresponding targets given by an expert (external agent). Therefore, learning occurs when the system effectively predicts the correct output for an non previously seen input. A second scenario is called *reinforcement learning* in which the system interacts with the environment by performing actions that feedback to the system in the form of rewards or punishments. In this case the observed inputs are called the states. The goal of adaptation is to find a suitable set of actions that will maximize the reward over time and this sequence will depend on the observed state. The third setting, called *unsupervised* learning. In this case, we can loosely say that the only information available is through the observed inputs. The assumption is the presence of regularities among the observed examples since there is a process behind their generation. Therefore, the goal is to find such regularities and one motivation for doing so is that expressing the available information in terms of the underlying causes may simplify further stages of processing. This view assumes a generative model for the observed data. However, one cannot argue the actual causes will be unveiled by the learning process since learning must be always accompanied by assumptions that may

or may not necessarily agree with reality [1] . Therefore, we can think of exploiting the statistical regularities to encode the observed inputs into more compact representations and also reduce the effect of some external noise which in the absence of additional information is assumed to be unstructured, by exploiting the redundancy in the inputs. This is a feature extraction point of view of unsupervised learning.

From system design perspective, feature extraction has been considered an important stage for increasing accuracy and reducing over-fitting in commonly encountered subsequent tasks such as classification or regression or action selection. While features can be handcrafted based on the external knowledge about the domain of the application, learning the features offers advantages such as adaptability across different domains without much external intervention. However, to apply the *learning from examples* paradigm to the problem of finding a representation, we need to address two main questions:

- **How to assess the effectiveness of a representation without having to evaluate the results on the subsequent tasks?**
- **What are the desirable properties in the representation?**

Throughout literature, this questions have been approached from different perspectives leading to myriad of techniques. Yet surprisingly, is the fact that many of these techniques overlap either by using the same criteria to assess their effectiveness, or in the properties of the conveyed representations.

1.2 Statistical Learning Perspective

Let us begin with the supervised learning setting. Here, we are limited to a set of n observations $\{(x_i, y_i)\}_{i=1}^{n} \subset \mathcal{X} \times \mathcal{Y}$ that are assumed to be *i.i.d.* and sampled

[1] In the absence of assumptions there is no privileged or "best" feature representation, and that even the notion between patterns depends implicitly on assumptions that may or not be correct [22]

from an unknown joint distribution $P(x, y)$. The dependence between X and Y is modeled through a functional relationship $y = f(x)$ that tries to capture the unknown conditional distribution $P(y|x)$. The function $f(x)$ is chosen from a set of possible functions (hypothesis) $\mathcal{F} = \{f(x, \alpha);\ \alpha \in \mathbf{\Theta}\}$. The quality of a particular approximation $f(x, \alpha)$ is *ideally* measured in terms of an average loss defined as:

$$R(\alpha) = \int_{\mathcal{X},\mathcal{Y}} \ell\left(x, y, f(x, \alpha)\right) \mathrm{d}P(x, y). \tag{1–1}$$

The loss function ℓ measures the point-wise accuracy of the approximation and therefore influences the optimal choice. Typically, ℓ is a nonnegative function and $\ell\left(x, y, f(x, \alpha)\right) = 0$ if $y = f(x, \alpha)$. (1–1) also know as risk functional cannot be computed in practice since $P(x, y)$ is unknown. Instead, an empirical version is computed based on the sample of size n as

$$R_{\text{emp}}(\alpha) = \int_{\mathcal{X},\mathcal{Y}} \ell\left(x, y, f(x, \alpha)\right) \mathrm{d}P_n(x, y) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(x_i, y_i, f(x_i, \alpha)\right) \tag{1–2}$$

where P_n denotes the empirical distribution $P_n(x, y) = \frac{1}{n}$ if $(x, y) = (x_i, y_i)$ for $i = 1, \ldots, n$, and 0 otherwise. The empirical risk (1–2) is often accompanied by a regularization term that control overfitting when the function class $\mathcal{F}$ is too rich

$$R_{\text{reg}}(\alpha) = R_{\text{emp}}(\alpha) + \Omega\left(f(x, \alpha)\right). \tag{1–3}$$

The regularization term $\Omega\left(f(x, \alpha)\right)$ artificially shrinks the function class, which is justified theoretically by generalization bounds of the form $R(\alpha) \leq R_{\text{emp}}(\alpha) + C(\mathcal{F})$, where $C(\mathcal{F})$ is a term measuring the *capacity* of the function class. The above risk can be adapted to the case of unsupervised learning. In this setting the desired outputs are the inputs themselves and f captures the structure of the data. Therefore it is necessary to find ways to restrict the the possible choices for f to avoid trivial solutions. How to restrict the choices is an open question that explains why in unsupervised learning, there has been distinctions between clustering and projection methods. However, some of the tools

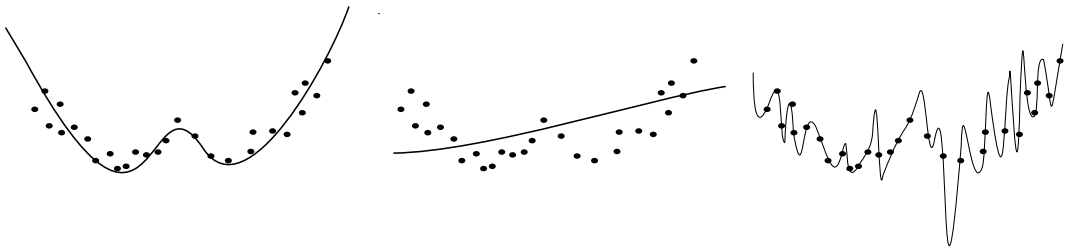


Figure 1-1. Different fittings to the data. The left figure shows a reasonable fit under smoothness assumptions of the functional dependency. The middle figure is an example of under fitting, which consider an overly simplistic model of the data and thus introduces large errors. The rightmost figure shows an example of how a rich class of functions can provide a candidate that can fit perfectly the data but maybe wrong in predicting the relation for unseen exemplars under the assumption of smoothness.

employed to analyze and solve the problems show that they are intimately related [45]. In the following, we motivate the idea of using information theory to address the problem of learning and highlight the relation with the regularized risk minimization setting that we have just presented.

1.3 Rate Distortion, the Information Bottleneck, and Learning

In the Shannon's view of information transfered through a channel, it was shown that the noise and bandwidth inherent to the channel determines the rate at which information can be transfered, and any attempt to transfer information above the limit imposed by the channel will incur in some error in the recovered message. This result is known as the channel capacity where the objective is to transmit information with zero error limit. Allowing some loss of information in the process yields the rate-distortion function. The information rate-distortion function $R(D)$ is defined as:

$$R(D) = \min_{q(\hat{x}|x)} I(X;\hat{X}) \text{ subject to } \mathrm{E}[d(X,\hat{X})] \leq D \tag{1–4}$$

where $d(\cdot,\cdot)$ is a distortion function that can be related to the loss incurred when the exact value of the input cannot be recovered from the output and $\mathrm{E}[d(X,\hat{X})] = \sum_{x,\hat{x}} p(x)q(\hat{x}|x)d(x,\hat{x})$. As we can see the average distortion is an analogue quantity

to the risk functional (1–1), and the mutual information $I(X;\hat{X})$ can be related to a regularization term that provides the least committed mapping from x to $\hat{x}$.

1.3.1 Example: Rate Distortion and PCA

The relation between rate distortion and PCA arises from the assumption of Gaussian source. Consider the Gaussian random vector $X \in \mathbb{R}^d$ with zero mean and covariance $\mathbf{\Sigma}$. We can find similarity transformation $\mathbf{U}$ to a Gaussian random vector $Z \in \mathbb{R}^d$ with zero mean and covariance $\mathbf{\Gamma}$, such that $\mathbf{U}^T\mathbf{\Sigma}\mathbf{U} = \mathbf{\Gamma}$. The transformation matrix $\mathbf{U}$ is unitary which implies $\det(\mathbf{U}) = 1$, and thus, $h(X) = h(Z)$ and the mapping is a bijection [2] . The following theorem from [16] will complete our result.

Theorem 1.3.1. (Rate distortion for a parallel Gaussian source) *Let $X_i \sim \mathcal{N}(0, \gamma_i^2)$, $i = 1, \ldots, d$, be independent Gaussian random variables, and the distortion measure $d(x, \hat{x}) = \sum_{i=1}^{d}(x_i - \hat{x}_i)^2$. Then the rate distortion function is given by*

$$R(D) = \sum_{i=1}^{d} \frac{1}{2} \log \frac{\gamma_i}{D_i} \tag{1–5}$$

where

$$D_i = \begin{cases} \lambda & \text{if } \lambda < \gamma_i^2 \\ \gamma_i^2 & \text{if } \lambda \geq \gamma_i^2, \end{cases} \tag{1–6}$$

and λ is chosen so that $\sum_{i=1}^{d} D_i = D$.

Applying the result to the random vector Z yields the conclusion that the rate for components with variance less than λ will be zero, and thus, no effort in representing such component should be made. This is the scheme by PCA in selecting the components where a variance threshold is set and only those components above the threshold are retained. On the other hand, λ plays the role of the observation noise

[2] Note, we are not using mutual information to relate X and Z, because in this case the mutual information is infinite

in the generative model for PCA, but the main difference is that $\hat{Z}$ is not assumed to be $\mathcal{N}(0, \mathbf{I})$.

1.3.2 The Information Bottleneck

In rate distortion, the basic assumption is that the distortion measure is given; therefore, the results will depend on what distortion measure has been chosen. The information bottleneck [87] proposes an alternative way of formulating the precision-complexity trade-off without defining a distortion measure in advance. This is done by including a reference variable Y, called the *relevant variable*, into the problem. As in rate distortion, one wants to compress X into $\hat{X}$, such that, the mutual information between $\hat{X}$ and Y is maximized, that is,

$$\min_{q(\hat{x}|x)} I(X;\hat{X}) - \beta I(\hat{X};Y) \tag{1–7}$$

where $\beta > 0$ is the trade-off parameter. An equivalent formulation of (1–7) as a constrained optimization problem is based on the so called the relevance-compression function, which depends on the joint distribution $P(x, y)$, and is defined as follows:

$$\hat{R}(\hat{D}) = \min_{q(\hat{x}|x)} I(X;\hat{X}) \text{ subject to } I(\hat{X};Y) \geq \hat{D}, \tag{1–8}$$

that is, $\hat{R}(\hat{D})$ is the minimal achievable compression, for which the relevant information is above $\hat{D}$ [83]. It is important to make clear that $\hat{X}$ should be completely defined given X alone, this implies the following Markovian relation $Y \to X \to \hat{X}$, that combined with the data processing inequality shows that $I(\hat{X};Y) \leq I(X;Y)$.

In the above, we wanted to motivate the use of information theory to describe the learning problem. Nevertheless, the above formulation assume the availability of the distributions, which as we pointed out at the beginning of the discussion, is not the case. To make this formulation practical, we must have actual estimators of such operational quantities based on the observed data. Here, we thrive in the previous work on *information theoretic learning* (ITL) and its relation to kernel methods in machine learning.

This brings along the generality level at which information theory treats a problem with the powerful tools from machine learning to deal with practical problems. The goal of this thesis is to develop a framework for learning based information theoretic principles that effectively captures statistical regularities directly from data. To this end, we employ reproducing kernel Hilbert space methods and show how they can be employed to compute quantities with similar properties to entropy, mutual information and divergence. In this way, we can successfully integrate information theoretic principles to the problem of learning and obtain data driven representations that convey the necessary information to perform subsequent tasks, *i.e.* further stages of data processing.

First, we investigate an objective function called the *principle of relevant information* (PRI), which trades off between faithfulness of the representation as information preservation and parsimony of the representation. This brings together two core concepts of information theory, divergence and entropy. From the principle, it has been observed different levels of interaction of these two quantities relate to common tasks in unsupervised learning, ranging from clustering to manifold learning, and back to quantization. The estimators employed within the principle have led to establish connections between kernel methods and information theoretic objectives, and integrate powerful tools from optimization and machine learning into the algorithms that effectively manipulate information at its very essence.

In the second part, motivated by the idea of avoiding density estimation, we propose a matrix based formulation of entropy-like quantities that can be employed for information theoretic learning. These ideas lead to the concept infinitely divisible matrices and establish a theoretical framework that allows the use of reproducing kernel Hilbert space methods to pose learning problems as optimization of information theoretic quantities.

1.4 Contributions of the Present Work

We introduce an alternative formulation of the principle of relevant information based on the reproducing kernel Hilbert space understanding of ITL with the cross information potential. We showed how the information theoretic formulation of the principle can be cast as quasi-convex optimization problem and provide an efficient solution that was motivated by the sequential minimal optimization method developed for support vector machines. This first part serves as a motivation to investigate further on the RKHS formulations for ITL We develop a kernel based framework for information theoretic learning by studying the properties of entropy like functionals of positive definite matrices that avoids the process of estimating the underlying probability distribution as intermediate step. We arrive at the concept of infinitely divisible matrices as the bridge between two Hilbert spaces. One Hilbert space is related to the infinite divisible kernel provides the Gram matrix that can be employed to compute the entropy-like functionals on positive definite matrices. The second Hilbert space is the actual representation of the data, that comes from the relation between an infinite divisible matrices and conditionally negative definite matrices, that can be embedded in Hilbert spaces. We studied the integral operators involved in the limit cases of the proposed matrix functional using some of the analysis that has been done for positive definite kernels in the context of learning. This framework provides means of posing different learning problems (we mainly focus on supervised and unsupervised) based on information theoretic quantities, and brings to the table new application domains that where not addressed before, for instance applications that employ kernels for structured domains such as graphs, time series, could be included in the proposed framework.

CHAPTER 2
MATHEMATICAL PRELIMINARIES

In this chapter, we provide a brief account of the introductory concepts of the theory of reproducing kernel Hilbert spaces. We start with their formal definition and introduce the necessary and sufficient conditions for a function to be a reproducing kernel. Following, we present some situations where the theory of reproducing kernel that are relevant to learning. The contents of this chapter are based on the papers by Aronszajn [3], Parzen [58], the book by Schölkpf and Smola [76], and the book [17].

2.1 Reproducing Kernel Hilbert Spaces

Let $\mathcal{X}$ be a set and $\mathcal{F}$ be a vector space of functions from $\mathcal{X}$ to the a field $\mathbb{F}$; in particular, let $\mathbb{F} = \mathbb{R}$. We say that $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) on $\mathcal{X}$ over $\mathbb{R}$, if:

(i) $\mathcal{H}$ is a vector subspace of $\mathcal{F}$;

(ii) $\mathcal{H}$ is endowed with an inner product product, $\langle \cdot, \cdot \rangle$, and is complete in the metric induced by it;

(iii) for every $x \in \mathcal{X}$ and $f \in \mathcal{H}$, the linear evaluation functional $F_x : \mathcal{H} \mapsto \mathbb{R}$, defined as $F_x(f) = f(x)$, is bounded.

From Riez theorem [41], we know that for any bounded functional H on a Hilbert space $\mathcal{H}$, there exists a unique vector $h \in \mathcal{H}$ such that: $H(f) = \langle h, f \rangle$ for all $f \in \mathcal{H}$. In particular, for each evaluation functionals F_x there exist a corresponding vector $\kappa_x \in \mathcal{H}$. The bivariate function defined by

$$\kappa(x, y) = \kappa_x(y) \tag{2–1}$$

is called a *reproducing kernel* for $\mathcal{H}$; it is easy to verify that,

$$\kappa(x, y) = \langle \kappa_x, \kappa_y \rangle \tag{2–2}$$

and $\|F_x\|^2 = \|\kappa_x\|^2 = \langle \kappa_x, \kappa_x \rangle = \kappa(x, x)$. Let $\mathcal{H}$ be a RKHS on the set $\mathcal{X}$ with kernel κ. The linear span of $\{\kappa(\cdot, x) : x \in \mathcal{X}\}$ is dense in $\mathcal{H}$. This results from the fact that

any function f orthogonal to the the span of $\{\kappa(\cdot, x) : x \in \mathcal{X}\}$ must satisfy $\langle f, \kappa_x \rangle$ for all $x \in \mathcal{X}$, and thus $f(x) = 0$ for every $x \in \mathcal{X}$.

Lemma 2.1.1. *Let $\{f_n\} \subseteq \mathcal{H}$. If $\lim_n \|f_n - f\| = 0$, then $f(x) = \lim_n f_n(x)$ for every $x \in \mathcal{X}$*

Proof 2.1.1. *This is a simple consequence of the reproducing property and Cauchy-Schwarz inequality.*

$$|f_n(x) - f(x)| = |\langle f_n - f, \kappa_x \rangle| \leq \|f_n - f\| \|\kappa_x\| \to 0$$

□

A consequence of the above lemma is:

Proposition 2.1.1. *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be RKHS on $\mathcal{X}$ with kernels κ_1 and κ_2, respectively. If $\kappa_1(x, y) = \kappa_2(x, y)$ for all $x, y \in \mathcal{X}$, then $\mathcal{H}_1 = \mathcal{H}_2$ and $\|f\|_1 = \|f\|_2$ for every f.*

Proof 2.1.2. *we can take $\kappa(x, y) = \kappa_1(x, y) = \kappa_2(x, y)$ and thus the $M_\ell = \text{span}\{\kappa_x \in M_\ell : x \in \mathcal{X}\}$ is dense in $\mathcal{H}_\ell$, and for any $f(x) = \sum_i \alpha_i \kappa_{x_i}(x)$ there is no regard about whether f belongs to either M_1 or M_2. Note that $\|f\|_1^2 = \sum_{i,j} \alpha_i alpha_j \kappa(x_i, x_j) = \|f\|_2^2$, and thus $\|f\|_1 = \|f\|_2$ for every $f \in M_1 = M_2$. If $f \in \mathcal{H}_1$, then there is a sequence of functions $\{f_n\} \subset M_1$ that converge to f in norm. Since $\{f_n\}$ is Cauchy in M_1 is also Cauchy in M_2, so by completeness of $\mathcal{H}_2$ there exist $g \in \mathcal{H}_2$ such that $f_n \to g$. Then, by Lemma 2.1.1, we have that $f(x) = \lim_n f_n(x) = g(x)$ for every $x \in \mathcal{X}$, thus every $f \in \mathcal{H}_1$ is also in $\mathcal{H}_2$ and vice versa, and $\mathcal{H}_1 = \mathcal{H}_2$. Finally, we can extend $\|f\|_1 = \|f\|_2$ to all $\mathcal{H}_1$ and $\mathcal{H}_2$.*

□

In other words, two different reproducing kernel Hilbert spaces do not have the same reproducing kernel. The following theorem shows an alternative way to express the reproducing kernel of a RKHS $\mathcal{H}$

Theorem 2.1.1. *Let $\mathcal{H}$ have reproducing kernel κ. If $\{e_\lambda : \lambda \in \Lambda\}$ is an orthonormal basis for $\mathcal{H}$; then*

$$\kappa(x, y) = \sum_{\lambda \in \Lambda} e_\lambda(x) e_\lambda(y). \tag{2–3}$$

where the series converges point-wise.

Proof 2.1.3. *Since* $\{e_\lambda : \lambda \in \Lambda\}$ *is an orthonormal basis*

$$\kappa_x = \sum_{\lambda\in\Lambda} \langle e_\lambda, \kappa_x\rangle e_\lambda = \sum_{\lambda\in\Lambda} e_\lambda(x) e_\lambda$$

where the series converges in norm (Parseval identity). Then. by Lemma 2.1.1

$$\kappa_x(y) = \sum_{\lambda\in\Lambda} e_\lambda(x) e_\lambda(y) = \kappa(x, y).$$

□

Now, we turn the focus on $\kappa(x, y)$ and explore necessary and sufficient conditions for this function to be a reproducing kernel. A matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called *positive definite* if it is symmetric and

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j A_{ij} \geq 0 \tag{2–4}$$

for any $\alpha_i \in \mathbb{R}$. A real function of two variables $\kappa(x, y)$ is called a *positive definite function* if for any finite subset $\{x_i\} \subseteq \mathcal{X}$ the matrix $K_{ij} = \kappa(x_i, x_j)$ is positive definite.

Proposition 2.1.2. *Let* $\mathcal{H}$ *be a RKHS on* $\mathcal{X}$ *with reproducing kernel* κ*. Then,* κ *is a positive definite function*

Proof 2.1.4. *For a fixed* $\{x_i\} \subseteq \mathcal{X}$*, we have*

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(x_i, x_j) = \left\langle \sum_{i=1}^{n} \alpha_i \kappa_{x_i}, \sum_{i=1}^{n} \alpha_i \kappa_{x_i} \right\rangle = \left\| \sum_{i=1}^{n} \alpha_i \kappa_{x_i} \right\|^2 \geq 0$$

□

To conclude this section, we show Moore's Theorem, which is the converse to the above result and gives us a characterization of a positive definite functions to be a sufficient condition for the function to be the reproducing kernel of some RKHS $\mathcal{H}$.

Theorem 2.1.2. *Let* $\mathcal{X}$ *be a set and* $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ *be a positive definite function. Then, there exist a reproducing kernel Hilbert space* $\mathcal{H}$ *of functions on* $\mathcal{X}$*, such that,* κ *is the reproducing kernel of* $\mathcal{H}$

Proof 2.1.5. *Consider the functions $\kappa_x(y) = \kappa(x, y)$ and the space W spanned by the set $\{\kappa_x : x \in \mathcal{X}\}$. The following bilinear map $B : W \times W \mapsto \mathbb{R}$*

$$B\left(\sum_i \alpha_i \kappa_{x_i}, \sum_j \beta_j \kappa_{x_j}\right) = \sum_{i,j} \alpha_i \beta_j \kappa(x_i, x_j),$$

where $\alpha_i, \beta_j \in \mathbb{R}$, is well defined on W. To support the above claim, notice that if $f(x) = \sum_i \alpha_i \kappa_{x_i}(x)$ is zero for all $x \in \mathcal{X}$, then by definition $B(f, \kappa_x) = 0$ for all x. Conversely, if $B(f, w) = 0$ for all $w \in W$, then by taking $w = \kappa_x$ we see that $f(x) = 0$. Thus B is well defined.

Since κ is positive definite $B(f, f) \geq 0$ and we see that $B(f, f) = 0$ if and only if $B(w, f) = 0$ for all $w \in W$, therefore $f(x) = 0$ for all $\mathcal{X}$. Now we have shown that W is a pre-Hilbert space with inner product B. Let $\mathcal{H}$ denote the completion of W, we need to show that every element of $\mathcal{H}$ is function on $\mathcal{X}$. Let $h \in \mathcal{H}$ be the limit point of a Cauchy sequence $\{f_n\} \subseteq W$. By Cauchy-Schwarz inequality $|f_n(x) - f_m(x)| = |B(f_n - f_m, \kappa_x)| \leq \|f_n - f_m\| \kappa(x, x)$. Hence, the point-wise limit $h(x) = \lim_n f_n(x)$ is well defined. TO conclude, let $\langle \cdot, cdot \rangle$ be the inner product on $\mathcal{H}$. Then, we have $\langle h, \kappa_x \rangle = \lim_n \langle f_n, \kappa_x \rangle = \lim_n B(f_n, \kappa_x) = h(x)$. Thus $\mathcal{H}$ is a reproducing kernel Hilbert space with reproducing kernel κ.

□

Finally, combining Proposition 2.1.1 with Moore's Theorem shows there is a correspondence between RKHS's on the set $\mathcal{X}$ and positive definite functions on this set.

2.2 The Covariance Function

Consider a stochastic process $\{X(t) : t \in \mathcal{T}\}$, where $X(t)$ are real random variables defined on a probability space $(\Omega, \mathcal{B}, P)$ with bounded second order moments, that is,

$$\mathrm{E}\,|X(t)|^2 = \int_\Omega |X(t)|^2 \mathrm{d}P < \infty \tag{2–5}$$

Without loss of generality, we can consider random variables with zero mean, $\mathrm{E}[X(t)] = 0$ for all $t \in \mathcal{T}$; the covariance function is defined as,

$$R(s,t) = \mathrm{E}[X(s)X(t)] = \int_{\Omega} X(s)X(t)\mathrm{d}P \tag{2–6}$$

It is easy to verify that R is a positive definite function and therefore defines a reproducing kernel Hilbert space of functions on $\mathcal{T}$. A result originally due to Loéve, and presented by Parzen in [58] showed a congruence map between the RKHS induced by the function R and the L_2 space that correspond to the completion of the span of the set $\{X(t) : t \in \mathcal{T}\}$ denoted by $L_2(X(t) : t \in \mathcal{T})$.

Theorem 2.2.1. *Let $\{X(t) : t \in \mathcal{T}\}$ be a random process with covariance kernel R. Then $L_2(X(t) : t \in \mathcal{T})$ is congruent with the reproducing kernel Hilbert space $\mathcal{H}$ with reproducing kernel R. Furthermore, any linear map $\psi : \mathcal{H} \mapsto L_2(X(t) : t \in \mathcal{T})$ which has the property that for any $f \in \mathcal{H}$ and any $t \in \mathcal{T}$*

$$\mathrm{E}[\psi(f)X(t)] = f(t) \tag{2–7}$$

is the congruence from $\mathcal{H}$ onto $L_2(X(t) : t \in \mathcal{T})$, which maps $R(\cdot, t)$ into $X(t)$

2.3 RKHSs in Machine Learning

The study positive definite kernels in machine learning was initially motivated as a generalization of a well body of theory that has been developed for linear models and algorithms. In this context, a positive definite kernel κ is an implicit way to represent the set $\mathcal{X}$ objects of interest. We have already seen that there is a correspondence between a positive definite kernel κ and a reproducing kernel Hilbert space of functions $\mathcal{H}$ with reproducing kernel κ. Therefore, the kernel can be understood as an indirect way to compute inner products between elements of a Hilbert space that are the result of mapping the elements of $\mathcal{X}$ to $\mathcal{H}$. In other words, there exist a mapping $\phi : \mathcal{X} \mapsto \mathcal{H}$, such that

$$\kappa(x,y) = \langle \phi(x), \phi(y) \rangle. \tag{2–8}$$

The space $\mathcal{H}$ is know as the feature space and ϕ is called the feature map. One of the big appeals of this idea is that by performing linear operations in $\mathcal{H}$, we may be able to perform nonlinear manipulations in the input space $\mathcal{X}$, and most importantly that there is no need to perform any explicit computations in the feature space. Notice, this idea is completely different to the congruence map introduced in Theorem 2.2.1. An important result associated with the use of positive definite kernel in machine learning is the representer theorem [40] which is presented below.

Theorem 2.3.1. (Representer theorem)[76]. *Let $\Omega : [0, \infty) \mapsto \mathbb{R}$ be a strictly monotonic increasing function, $\mathcal{X}$ be a set, and $c : (\mathcal{X} \times \mathbb{R}^2)^n \mapsto \mathbb{R} \cup \infty$ be an arbitrary loss function. Then, each minimizer $f \in \mathcal{H}$ of the regularized risk functional*

$$c((x_1, y_1, f(x_1)), \dots, (x_n, y_n, f(x_n))) + \Omega\left(\|f\|_{\mathcal{H}}^2\right) \tag{2–9}$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{n} \alpha_i \kappa(x_i, x). \tag{2–10}$$

Proof 2.3.1. *Let $S = \text{span}\{\kappa(\cdot, x_i) : x_i \in \mathcal{X},\ i = 1, \dots, n\}$ denote the subspace of $\mathcal{H}$ spanned by the n training samples. Consider a solution $f \in \mathcal{H}$, this solution can be expressed as $f = f_S + f_{S^\perp}$ where $f_S \in S$ and $f_{S^\perp} \in S^\perp$. Therefore $f(x_i) = f_S(x) + f_{S^\perp}(x_i) = f_S(x) + 0$. Now for the second term of (2–9), we have that*

$$\Omega\left(\|f\|_{\mathcal{H}}^2\right) = \Omega\left(\|f_S\|_{\mathcal{H}}^2 + \|f_{S^\perp}\|_{\mathcal{H}}^2\right),$$

since Ω is strictly monotonic increasing we can see that the minimum will be achieved for $\|f_{S^\perp}\|_{\mathcal{H}}^2 = 0$ which implies that $f_{S^\perp} = 0$.

□

The representer theorem basically states that the solution to the minimization of the regularized risk functional can be expressed in term of the so called training sample $\{(x_i, y_i)\}_{i=1}^n$. This is important because it allows us to deal with problems that at first

glance appear to be infinite dimensional. Notice that the regularization does not prevent (2–9) of having local multiple minima, this property would require some extra conditions such as convexity.

CHAPTER 3
THE PRINCIPLE OF RELEVANT INFORMATION

3.1 Information Theoretic Learning in a Nutshell

An alternative paradigm for learning based on the ideas underlying information theory was proposed in [63][1] . Information theoretic learning (ITL) is a blend of ideas from information theory and the theory of adaptive systems to accomplish *information filtering* in a nonparametric fashion. ITL objective functions are build upon the concepts of entropy and divergence (extends to mutual information). Learning from data is made possible by using Parzen windows to compute plug in estimators of Renyi's entropy. For a random variable $X \in \mathcal{X} \subseteq \mathbb{R}^d$ and probability density function $f(x)$, Renyi defined a family of entropy functionals

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int_X f^{\alpha-1}(x) f(x) \mathrm{d}x \tag{3–1}$$

indexed by the parameter α. The case of $\alpha = 2$ is of particular interest because it provides an estimator of a rather simple form. Let $\{x_i\}_{i=1}^n \subset \mathcal{X}$ be an *i.i.d.* sample from the random variable X with density $f(x)$. For a Parzen window estimate $\hat{f}(x) = \frac{1}{n}\sum_{i=1}^n \kappa_\sigma(x, x_i)$, the plug in estimator of Renyi's entropy of order 2 is given by

$$\hat{H}_2(X) = -\log\left(\frac{1}{n^2}\sum_{i,j=1}^n \kappa_{\sqrt{2}\sigma}(x_i, x_j)\right) \tag{3–2}$$

where $\kappa_\sigma(x, y) = C_\sigma \exp -\frac{\|x-y\|^2}{2\sigma^2}$ is the Gaussian kernel with parameter σ. Following a similar line of thinking, ITL uses two definition of divergence that are compatible with

[1] A more complete account of ideas developments and algorithms can be found in [62].

Renyi's entropy of order 2: a divergence measure based on the Euclidean norm,

$$\begin{aligned} D_{\mathrm{ED}}(X;Y) &= \int_{\mathcal{X}} \left(f(x) - g(x)\right)^2 \mathrm{d}x \\ &= \int_{\mathcal{X}} f^2(x)\mathrm{d}x + \int_{\mathcal{X}} g^2(x)\mathrm{d}x - 2\int_{\mathcal{X}} f(x)g(x)\mathrm{d}x; \end{aligned} \tag{3–3}$$

and a measure based on the Cauchy-Schwarzs inequality,

$$\begin{aligned} D_{\mathrm{CS}}(X;Y) &= -\log \frac{\left(\int_{\mathcal{X}} f(x)g(x)\mathrm{d}x\right)^2}{\int_{\mathcal{X}} f^2(x)\mathrm{d}x \int_{\mathcal{X}} f^2(x)\mathrm{d}x} \\ &= \log \int_{\mathcal{X}} f^2(x)\mathrm{d}x + \log \int_{\mathcal{X}} g^2(x)\mathrm{d}x - 2\log \int_{\mathcal{X}} f(x)g(x)\mathrm{d}x. \end{aligned} \tag{3–4}$$

Note that both divergences share similar terms, but most interestingly is that these terms correspond to variations of the same building block. The cross-information potential

$$V(X,Y) = \int_{\mathcal{X}} f(x)g(x)\mathrm{d}x \tag{3–5}$$

which correspond to an inner product between probability density functions and thus acts as a measure of similarity. It turns out that the estimator of (3–5) also takes a simple form when estimated from data. Let $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^m$ be *i.i.d.* samples from $f(x)$ and $g(x)$, respectively; The plug in estimator of the cross-information potential is given by:

$$\hat{V}(X,Y) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} \kappa_{\sqrt{2}\sigma}(x_i, y_j) \tag{3–6}$$

3.2 The Principle of Relevant Information

All the methods we reviewed for unsupervised learning C refer to information preservation, but their approach at quantifying information is somehow loose given that information theory provides operational quantities that appropriately accomplish such objective. Nevertheless, estimating information theoretic quantities from data is a difficult problem leading to simplifications and assumptions that may lead to poor

results or negligible improvement if data is far from being in agreement with the imposed restrictions. In this respect, **information theoretic learning** (ITL) [62] develops a framework that deals with estimators that have the information theoretic flavor yet are simple to compute from data. Here, we present a principle motivated by the goals in unsupervised learning, which effectively manipulates information by setting the problem within th ITL framework.

Structure can be associated to the statistical regularities present on the outcomes of a process. Therefore, the entropy related to these outcomes can be attributed in part to the underlying structure, and the rest to particularities of each outcome, *i.e.* details or even random perturbations. Hence, we can think of the minimization of entropy as a means of finding such regularities. Suppose we are given a random variable S with PDF g, for which we want to find a description in terms of a PDF f with reduced entropy, that is, a variable X that captures the underlying structure of S. The **principle of relevant information** (PRI) formulates the above problem as a trade-off between the entropy $H_2(f)$ of X and its descriptive power about the observed random variable S in terms of their relative entropy $D_{\mathrm{CS}}(f\|g)$. For a fixed PDF $g \in \mathcal{F}$ the objective is given by:

$$J(f) = H_2(f) + 2\lambda D_{\mathrm{CS}}(f\|g), \tag{3–7}$$

where λ is the trade-off parameter. The minimization of J within a set of admissible PDFs $\mathcal{F}$ should lead to a function $f \in \mathcal{F}$ that has minimum entropy, but at the same time, the information gain from observing the sample which is represented by g is also maximized. Nevertheless, as it is often the case, it is necessary to choose a way to compute g from the sample and suitable space $\mathcal{F}$ to search for f. The only available information about g is encoded in a sample $S = \{x_i\}_{i=1}^{N}$, and some assumptions about the function class $\mathcal{F}$ must be made in order to obtain a tractable solution.

3.2.1 PRI as a Self-organization Mechanism

A suitable way to frame the above search as an optimization problem was proposed in [68]. This solution combines Parzen density estimation with a self organization of a sample to match the desired density f that minimizes (D–8). The optimization problem becomes:

$$\arg \min_{X \in (\mathbb{R}^d)^M} \left[\hat{H}_2(X) + \lambda \hat{D}_{\mathrm{CS}}(X \| S) \right] \tag{3–8}$$

where $S \in (\mathbb{R}^d)^N$ is a set of d-dimensional points with cardinality N, and X a set of d-dimensional points with cardinality M. For the Parzen window with Gaussian kernel G_σ we can evaluate the cost as:

$$-(1-\lambda) \log \left(\frac{1}{M^2} \sum_{i,j=1}^{M} G_{\sigma\sqrt{2}}(x_i, x_j) \right) - \lambda \log \frac{\left(\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_i, s_j) \right)^2}{\left(\frac{1}{N^2} \sum_{i,j=1}^{N} G_{\sigma\sqrt{2}}(s_i, s_j) \right)} \tag{3–9}$$

where $x_i \in X$ and $s_i \in S$. We can further simplify the cost by dropping the denominator on the right term of (D–11) since it is constant. The self organization principle moves each particle x_i according to the forces exerted by the samples S and X. The entropy minimization creates attractive forces among the information particles x_i's interacting with the forces induced by the field created by the sample S, which restricts the motion of each x_i. The particles will move around the space until the whole system reaches a point of equilibrium (Local minimum of $J(X)$). Computing the partial derivatives of (D–11) with respect to each point in X yields a fixed point update described in Algorithm 1.

3.2.2 On the Influence of λ

Notice that the trade-off parameter λ can be varied between 0 and ∞ determining the strengths of each term in J and thus the equilibrium points. Consequently, we should expect different behaviors of the system according to λ. First, we can look at the extreme points in the range $[0, \infty)$. For $\lambda \to 0$ all the points in X will collapse to one

Algorithm 1 PRI algorithm with self organization

1: Initialization elements:

- S set of points for the original pdf $\hat{g}$.
- $X^{(0)} = S$ initial guess for $\hat{f}$.

2: Compute the information potential of $X^{(t)}$

$$\hat{V}(X^{(t)}) = \frac{1}{M^2} \sum_{i,j=1}^{M} G_{\sigma\sqrt{2}}(x_i^{(t)}, x_j^{(t)})$$

3: Compute the cross-information potential between $X^{(t)}$ and S

$$\hat{V}(X^{(t)}, S) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_i^{(t)}, s_j)$$

4: Update the elements of X with the following fixed point rule:

$$x_i^{(t+1)} = c \left(\frac{1-\lambda}{\lambda} \right) \frac{\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_i^{(t)}, x_j^{(t)}) x_j^{(t)}}{\sum_{j=1}^{M} G_{\sigma\sqrt{2}}(x_i^{(t)}, s_j)} + \frac{\sum_{j=1}^{M} G_{\sigma\sqrt{2}}(x_i^{(t)}, s_j) s_j}{\sum_{j=1}^{M} G_{\sigma\sqrt{2}}(x_i^{(t)}, s_j)}$$
$$- c \left(\frac{1-\lambda}{\lambda} \right) \frac{\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_i^{(t)}, x_j^{(t)})}{\sum_{j=1}^{M} G_{\sigma\sqrt{2}}(x_i^{(t)}, s_j)} x_i^{(t)}$$

where $c = \frac{M\hat{V}(X^{(t)},S)}{N\hat{V}(X^{(t)})}$

5: Iterate the steps 2 through 4 until convergence.

single point, which in the limit case becomes independent of the target sample S. This does not appear to be a very interesting case. The other extreme case is when X is initialized by the locations provided by the sample S and $\lambda \to \infty$. Here the locations of X will not move away from the locations of S, the system is already in equilibrium. Interesting cases arise when $\lambda \geq 1$. For instance, it has been shown that the case $\lambda = 1$ correspond to the Gaussian mean shift algorithm [67]. Other values of lambda can provide other descriptions of the statistical structure of the data. We will illustrate this phenomenon with a simple example, figure 3-1 shows a 2-dimensional sample S of points distributed in a non-linear arrangement. Algorithm 1 is run for kernel size $\sigma = 0.5$ and different values of λ. Figure 3-2 shows the behavior of the entropy term $\hat{H}_2(X)$ and

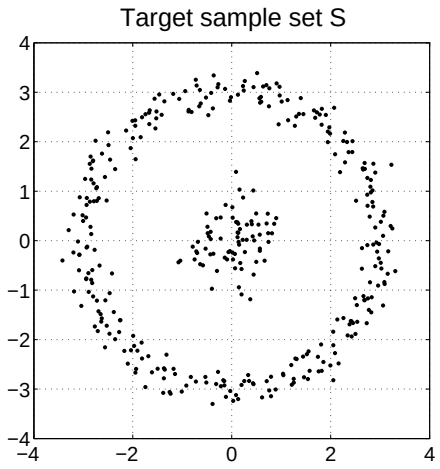


Figure 3-1. Data set utilized to illustrate the behavior of the self organization obtained by the principle of relevant information

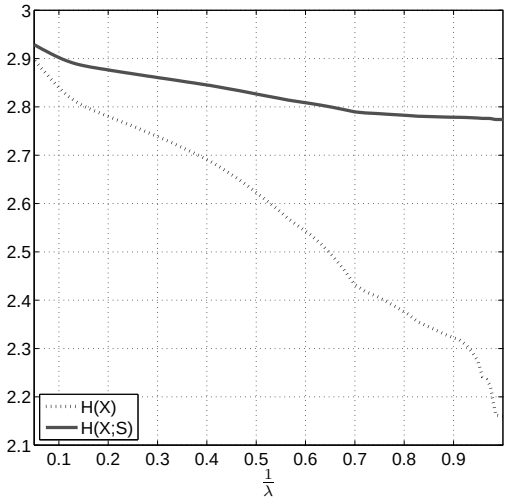


Figure 3-2. Entropy X and cross-entropy with respect to S after equilibrium for different values of λ. Notice that the values are plotted against $\frac{1}{\lambda}$ for better visualization.

the cross-entropy term $\hat{H}_2(X;S)$ obtained from the numerator of the rightmost term of (D–11) for different values of the trade-off parameter on the dataset. We can observe how this terms go through a series of transitions as λ increases. Figure 3-3 shows the different arrangements of the points given by the different values of the trade off parameter. We can see that the first transition occurs when λ starts moving away from 1. In the case of $\lambda = 1$ the points locate on the modes of the Parzen density estimator

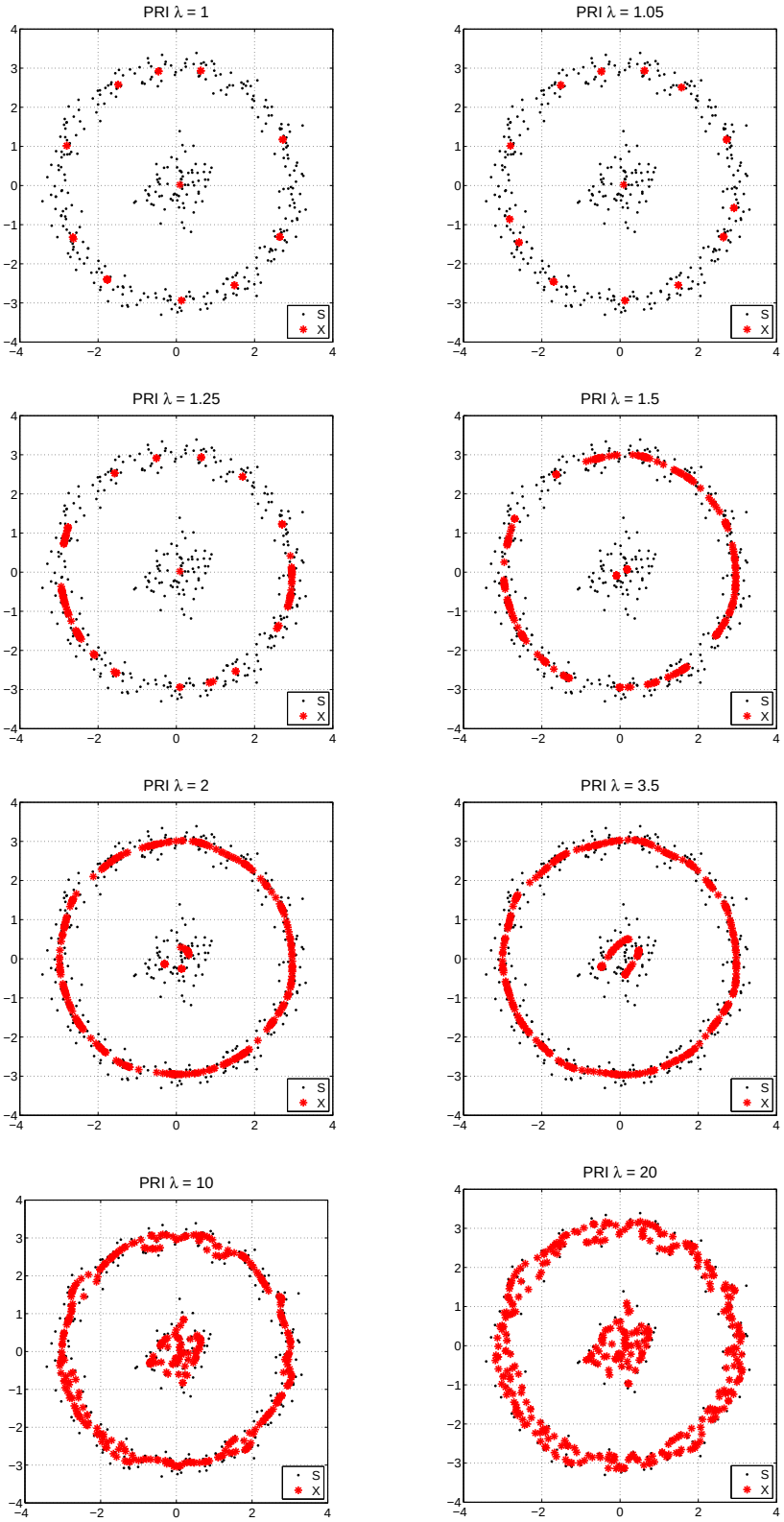


Figure 3-3. Resulting sample X after self-organization for different values of λ.

of the PDF according to S. As λ increases points in X start to spread out, however they do not move freely in all directions, and accommodate on rather regular patterns. For example, for $\lambda = 2$ the points appear to form regular patterns reminiscent of principal curves. As we go for larger values of λ, we see that regularity becomes less and less evident, and points in X will spread similarly to the points in the sample S.

3.2.3 A Note on Information Theoretic Vector Quantization

Above, we argued that if we use the sample S as initial guess for X and we let $\lambda \to \infty$, we end up with $X = S$ as the nearest equilibrium for the system related to the PRI cost. Basically, this occurs because we only put emphasis on the C-S divergence term in (D–8), and thus, a minimum of $J(X)$ is achieved if we have a set of points that generate the same Parzen density estimate with the one based on S. Indeed, the term controlling the uncertainty of X is neglected. This apparently non-interesting case can be given a positive twist if we impose an entropy constraint in an indirect fashion. The entropy minimization creates a bottleneck on the amount of information the output contains. Quantization is an implicit way of limiting the amount of information that can be possibly conveyed about the input variable by limiting the representation alphabet. If we let the number of points X be less than the number of points in S, with proper initialization, we can obtain a vector quantization algorithm since points in X will distribute evenly on according to the Cauchy-Schwarz divergence with respect to S to faithfully represent the set of points S. An important point here is that the capacity of the representation is not limited by an entropic term, but by the number of points employed in X. The quantization optimization problem is given by:

$$\arg \min_{X \in (\mathbb{R}^d)^M} \left[\hat{H}_2(X) + \lambda \hat{D}_{\mathrm{CS}}(X \| S) \right], \tag{3–10}$$

where M is the number of possible states after quantization. Therefore, the self-organization mechanism provided by the PRI can be applied in the training of a quantizer. Figure 3-4 displays the vector quantization solution obtained by the PRI fixed point algorithm with

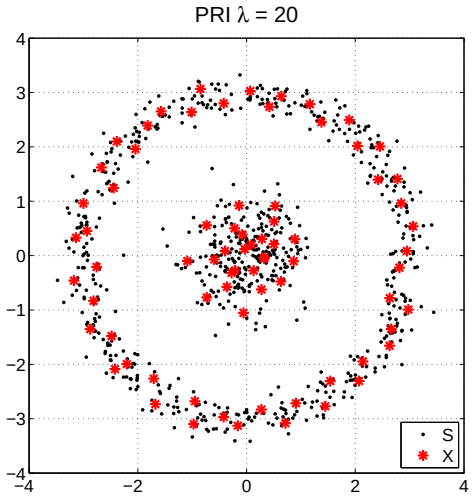


Figure 3-4. Resulting vector quantization using the PRI.

linear kernel annealing and uniform grid initialization for X. Notice that in this case the fidelity measure of the vector quantization is based on the distribution, which has a clear effect in how the vectors in X will distribute according to S. Since all points in X are equally weighted, the PRI will place more points on regions where the PDF estimated from S has larger values. This may or may not be an appealing property, but it is highly problem dependent. For example, the application shown in [68] for the face contours benefits from the above property since more vectors are located in regions that contain more detailed features since they are more densely sampled. However, for other compression scenarios the actual details are very rare events in the distribution and thus overlooked by the PRI.

3.2.4 Practical Issues and Open Questions

The use of marginal distributions to optimize the objective function through self organization arises practical issues and theoretical questions that eventually will lead to a better understanding of the principle. Among the practical issues that were not previously addressed we can list:

- How can we reduce the computational complexity of the optimization?

- How well does the self organization principle operate in high dimensional spaces? What is the influence of the Parzen density estimation underlying the solution?
- How can we apply the principle to other domains besides $\mathbb{R}^d$?
- Is there a way to make the encoding explicit?

In the attempt to solve these issues, we also encounter a number of questions from the theoretical point of view:

- How does the PRI avoid the computation of mutual information, which seems to be a better-suited goal?
- Is there a more general formulation of the PRI than the one based of Parzen?
- Is there a geometric understanding of the PRI?
- What are the approximation guarantees for the finite sample regime?

In the following sections, we start to address some of the above issues to shape the developments presented though the rest of this thesis.

3.3 Alternative Solutions to the Principle of Relevant Information

3.3.1 The PRI as a Weighting Problem

For the set $\mathcal{F}$ of probability density functions that are square integrable in $\mathcal{X} \subseteq \mathbb{R}^n$, we can define the cross-information potential $\mathcal{V}$ (CIP) as a bilinear form that maps densities $f_i, f_j \in \mathcal{F}$ to the real numbers trough the integral,

$$\mathcal{V}(f_i, f_j) = \int_{\mathcal{X}} f_i(x) f_j(x) \mathrm{d}x. \tag{3–11}$$

It is easy to see that for a basis of uniformly bounded, square integrable probability density functions, $\mathcal{V}$ is a positive semidefinite function on the $\mathrm{span}\{\mathcal{F}\}$. Now, consider the set $\mathcal{G} = \{g = \sum_{i=1}^{m} \alpha_i \kappa_\sigma(x_i, \cdot) | x_i \in \mathbb{R}^n,\ \sum_{i=1}^{m} \alpha_i = 1, \text{ and } \alpha_i \geq 0\}$, where κ_σ is a "Parzen" type of kernel which is also square integrable, that is, κ_σ is symmetric, nonnegative, has bounded integral (can be normalized), belongs to L^2, and shift invariant with σ as the scale parameter. Clearly, for any $g \in \mathcal{G}$ we have $\|g\|_2 \leq$

$\|\kappa_\sigma(x,\cdot)\|_2$. Thence ,$\mathcal{G}$ is bounded. However, if the $\mathcal{X}$ is non-compact our search space is also non-compact.

The objective function for the principle of relevant information (D–8) can be written in terms of IP function. Using the Parzen based estimation, we restrict the search problem to $\mathcal{G} \subset \mathcal{F}$. In this case, we have that equation (D–8) can be rewritten as:

$$J(f) = -\log \mathcal{V}(f,f) - \lambda \log \frac{[\mathcal{V}(f,g)]^2}{\mathcal{V}(f,f)\mathcal{V}(g,g)} \tag{3–12}$$

straightforward manipulation of the terms yields an equivalent problem:

$$\arg\min_{f\in\mathcal{G}} \left[-(1-\lambda)\log \mathcal{V}(f,f) - 2\lambda \log \mathcal{V}(f,g)\right] \tag{3–13}$$

Two important aspects of the above objective are: the choice of the kernel, shape and size σ, determines different scales for the analysis; the trade-off parameter λ defines a set of regimes for the possible solutions to the problem. As we previously mentioned, the only available information is contained in the sample $S = \{x_i\}_{i=1}^N$. An approximation of the target density g is then given by its weighted Parzen window estimator $\hat{g}(x) = \sum_{i=1}^N \alpha_i \kappa_\sigma(x_i, x)$, where $\alpha_i \geq 0$ and $\sum_{i=1}^N \alpha_i = 1$. In our experiments, we limit to $\alpha_i = 1/N$. To enforce compactness in our search space, we look for a solution f that has the same form of $\hat{g}$, that is

$$f(x) = \sum_{i=1}^N \beta_i f_i(x) = \sum_{i=1}^N \beta_i \kappa_\sigma(x_i, x). \tag{3–14}$$

where $\beta_i \geq 0$ and $\sum_{i=1}^N \beta_i = 1$. By fixing λ and evaluating the information potential between each pair $(x_i, x_j) \in S \times S$, we can rewrite (3–13) in matrix notation as:

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \left[(\lambda-1)\log \boldsymbol{\beta}^\top \mathbf{V} \boldsymbol{\beta} - 2\lambda \log \boldsymbol{\beta}^\top \mathbf{V} \boldsymbol{\alpha}\right] \\ &\text{subject to } \beta_i \geq 0 \text{ and } \sum_{i=1}^N \beta_i = 1, \end{aligned} \tag{3–15}$$

where $V_{ij} = \int_X \kappa(x_i, x)\kappa(x_j, x)\mathrm{d}x$. It it also important to highlight that for values of $\lambda \to 1$ the regularization term almost vanishes, and the maximum information gain (inner product term) can be related to the modes of $\hat{g}$ which are approximated by the sharpest functions available, which in our setting, are individual windows that satisfy convex combination constraint. Therefore, we can expect the modes of $\hat{g}$ to be almost orthogonal components. When λ is very close to one, the cross- entropy term is dominant in the objective and the solution will lie on the corners in one of the corners of the simplex. Figure 3-5 depicts the geometrical elements that explain this phenomenon. The semicircles are loci of minimum and maximum L^2 norm that intersect with the simplex being the solution f^* with the largest norm the one maximizing the dot product with the target distribution g.

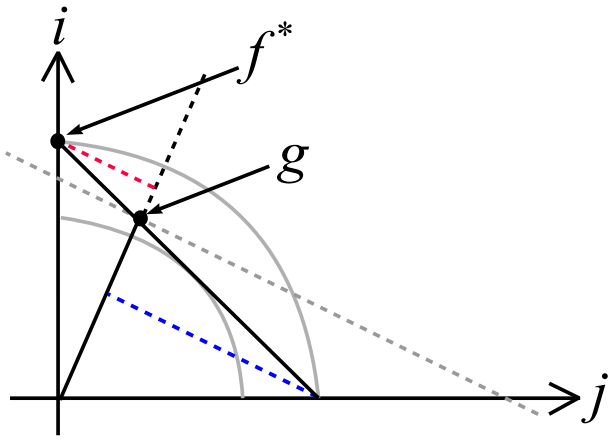


Figure 3-5. Corner effect of the PRI objective based on weights

This insight highlights one of the major differences from the predecessor of the this alternative formulation, and identifies the clustering behavior of the self-organization principle as a local minimum of the objective function. Nevertheless, this is not to be be taken as a disadvantage, it actually suggest a connection with the energy-models for learning that we reviewed in Section C. This corner phenomena also motivates the extraction of more than one solution. A reasonable approach is to deflate the target PDF

$\hat{g}$ and run the algorithm on the new function. The deflation is given by:

$$\boldsymbol{\alpha}_{\text{new}} = \boldsymbol{\alpha}_{\text{crt}} - \frac{\boldsymbol{\beta}^{\top}\mathbf{V}\boldsymbol{\alpha}_{\text{crt}}}{(\boldsymbol{\beta}^{\top}\mathbf{V}\boldsymbol{\beta})}\boldsymbol{\beta}. \tag{3–16}$$

the deflation just seems the right thing to do, but we don't have any theoretical justification for doing this. The optimization problem in (3–15) can be solved using methods such as projected gradient along with a penalty function, but we run into difficulties to set stepsize, scheduling the penalty function [14], and a matrix inversion is required in the projection step. Application of second order methods such as Newton give rise to at least $\mathcal{O}(N^2)$ complexities in memory and calculations. In the next section, we investigate a sequential optimization scheme that alleviates the memory requirement, making the algorithm scalable to more realistic scenarios where sample sizes are in the order tenths of thousands.

3.3.2 Sequential Minimal Optimization for the PRI

Notice that the form of the problem adopted in (3–15) is not a convex program. Nevertheless, it can be turned into an equivalent form that can be recognized as a convex program.

Proposition 3.3.1. *The convex program,*

$$\begin{aligned} \min_{\boldsymbol{\beta}} \ & \boldsymbol{\beta}^{\top}\mathbf{V}\boldsymbol{\beta} \\ \text{subject to } & \boldsymbol{\beta} \geq 0 \\ & \mathbf{q}^{\top}\boldsymbol{\beta} - \eta = 0 \\ & \mathbf{1}^{\top}\boldsymbol{\beta} - 1 = 0, \end{aligned} \tag{3–17}$$

is equivalent to (3–15)*, where* $\mathbf{q} = \mathbf{V}\boldsymbol{\alpha}$ *and some* $\eta > 0$*.*

Proof 3.3.1. *By definition* $\eta > 0$*, thus the constraint* $\log \mathbf{q}^{\top}\boldsymbol{\beta} = \log \eta$ *is equivalent to* $\mathbf{q}^{\top}\boldsymbol{\beta} - \eta = 0$*. The positive semi definiteness of the information potential tell us that* $\boldsymbol{\beta}^{\top}\mathbf{V}\boldsymbol{\beta} \geq 0$*. However, taking into account* $\mathbf{q}^{\top}\boldsymbol{\beta} - \eta = 0$ *guarantees strict inequality; therefore, the minimizers of* $\log \boldsymbol{\beta}^{\top}\mathbf{V}\boldsymbol{\beta}$ *and* $\boldsymbol{\beta}^{\top}\mathbf{V}\boldsymbol{\beta}$ *on the constraint set defined in* (3–17)

are the same. Hence, solving the following pseudo-convex program

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \log \boldsymbol{\beta}^{\mathsf{T}} \mathbf{V} \boldsymbol{\beta} \\ \text{subject to} \quad & \boldsymbol{\beta} \geq 0 \\ & \log \mathbf{q}^{\mathsf{T}} \boldsymbol{\beta} = \log \eta \\ & \mathbf{1}^{\mathsf{T}} \boldsymbol{\beta} - 1 = 0, \end{aligned} \tag{3–18}$$

should yield the same solution. Now, the gradient of the objective in (3–15) *with respect to the weight vector* $\boldsymbol{\beta}$ *is,*

$$\nabla J(\boldsymbol{\beta}) = 2 \left(\frac{\lambda - 1}{\boldsymbol{\beta}^{\mathsf{T}} \mathbf{V} \boldsymbol{\beta}} \right) \mathbf{V} \boldsymbol{\beta} - 2 \left(\frac{\lambda}{\boldsymbol{\beta}^{\mathsf{T}} \mathbf{V} \boldsymbol{\alpha}} \right) \mathbf{V} \boldsymbol{\alpha}. \tag{3–19}$$

By including the constraints $\mathbf{1}^{\mathsf{T}} \boldsymbol{\beta} = 1$ *and* $\boldsymbol{\beta} \geq 0$*, for* $\lambda > 1$*, the set of KKT necessary conditions for local optimality in the Lagrangian* $L(\boldsymbol{\beta}, \boldsymbol{\mu}, \gamma) = J(\boldsymbol{\beta}) + \sum_{i=1}^{N} \mu_i c_i(\boldsymbol{\beta}) + \gamma e(\boldsymbol{\beta})$ *is*

$$\begin{cases} \frac{\partial}{\partial \boldsymbol{\beta}} L(\boldsymbol{\beta}, \boldsymbol{\mu}, \gamma) = \nabla J(\boldsymbol{\beta}) + \sum_{i=1}^{N} \mu_i \frac{\partial}{\partial \boldsymbol{\beta}} c_i(\boldsymbol{\beta}) + \gamma \frac{\partial}{\partial \boldsymbol{\beta}} e(\boldsymbol{\beta}) = 0, \\ \frac{\partial}{\partial \boldsymbol{\mu}} L(\boldsymbol{\beta}, \boldsymbol{\mu}, \gamma) = \mathbf{c}(\boldsymbol{\beta}) \leq 0, \\ \boldsymbol{\mu}^{\mathsf{T}} \mathbf{c}(\boldsymbol{\beta}) = 0 = -\boldsymbol{\mu}^{\mathsf{T}} \boldsymbol{\beta}, \\ \boldsymbol{\mu} \geq 0, \\ \frac{\partial}{\partial \gamma} L(\boldsymbol{\beta}, \boldsymbol{\mu}, \gamma) = e(\boldsymbol{\beta}) = 0 = \mathbf{1}^{\mathsf{T}} \boldsymbol{\beta} - 1. \end{cases} \tag{3–20}$$

There are two possible cases for each β_i^*

- $\beta_i^* > 0$.
  For which $\mu_i^* = 0$ *and*

  $$2 \frac{t_i}{\boldsymbol{\beta}^{*\mathsf{T}} \mathbf{V} \boldsymbol{\beta}^*} - 2 \left(\frac{\lambda}{\lambda - 1} \right) \frac{q_i}{\boldsymbol{\beta}^{*\mathsf{T}} \mathbf{q}} + \gamma = 0, \tag{3–21}$$

  where $\mathbf{t} = \mathbf{V} \boldsymbol{\beta}$.

- $\beta_i^* = 0$.
  Yields

  $$2 \frac{t_i}{\boldsymbol{\beta}^{*\mathsf{T}} \mathbf{V} \boldsymbol{\beta}^*} - 2 \left(\frac{\lambda}{\lambda - 1} \right) \frac{q_i}{\boldsymbol{\beta}^{*\mathsf{T}} \mathbf{q}} - \mu_i^* + \gamma = 0. \tag{3–22}$$

Notice that $\gamma = 2\left(\frac{1}{\lambda-1}\right)$*, therefore,*

$$2\frac{\mathbf{V}\boldsymbol{\beta}^*}{\boldsymbol{\beta}^{*\top}\mathbf{V}\boldsymbol{\beta}^*} - 2\left(\frac{\lambda}{\lambda-1}\right)\frac{\mathbf{q}}{\boldsymbol{\beta}^{*\top}\mathbf{q}} - \boldsymbol{\mu}^* + 2\frac{\mathbf{1}}{\lambda-1} = 0. \tag{3–23}$$

Pre-multiplying (3–23) *by* $(\boldsymbol{\beta} - \boldsymbol{\beta}^*)$ *in the constraint set, yields the following set of conditions*

$$\begin{aligned}
(\boldsymbol{\beta} - \boldsymbol{\beta}^*)^\top \left[2\frac{\mathbf{V}\boldsymbol{\beta}^*}{\boldsymbol{\beta}^{*\top}\mathbf{V}\boldsymbol{\beta}^*} - \boldsymbol{\mu}^*\right] &\geq 0, \ \forall \boldsymbol{\beta} \geq 0 : \ \mathbf{q}^\top\boldsymbol{\beta} = \eta, \mathbf{1}^\top\boldsymbol{\beta} = 1 \\
\boldsymbol{\mu}^{*\top}\boldsymbol{\beta}^* &= 0 \\
-\boldsymbol{\beta}^* &\leq 0 \\
\boldsymbol{\mu}^* &\geq 0 \\
\log \mathbf{q}^\top\boldsymbol{\beta}^* &= \log \eta \\
\mathbf{1}^\top\boldsymbol{\beta}^* &= 1,
\end{aligned} \tag{3–24}$$

which by Theorem B.1.1 are sufficient conditions for the solution of a pseudo-convex function defined on an open set with convex inequality constraints, that in our case corresponds to (3–18)*.*

□

Two important results come from the above proposition. One is obvious from the statement in the proposition that tells us there exist an equivalent convex program that solves (3–15). But even better is the one that comes as a byproduct of the proof. The KKT first order conditions in (3–20) are necessary and sufficient to solve (3–15).

3.3.2.1 Decomposition into smaller subproblems

In the proof of Proposition 3.3.1, we solve a more convenient form of (3–15), for which we factorize $(\lambda - 1)$ from the objective. If we derive the the solution for the original problem, the two cases (3–21) and (3–22) are replaced by:

- $\beta_i^* > 0$.
  With $\mu_i^* = 0$ and
  $$2\frac{\lambda-1}{\boldsymbol{\beta}^{*\top}\mathbf{V}\boldsymbol{\beta}^*}t_i - 2\frac{\lambda}{\boldsymbol{\beta}^{*\top}\mathbf{q}}q_i + \gamma = 0, \tag{3–25}$$

where $\mathbf{t} = \mathbf{V}\boldsymbol{\beta}$.

- $\beta_i^* = 0$.

$$2\frac{\lambda - 1}{\boldsymbol{\beta}^{*\top}\mathbf{V}\boldsymbol{\beta}^*}t_i - 2\frac{\lambda}{\boldsymbol{\beta}^{*\top}\mathbf{q}}q_i - \mu_i^* + \gamma = 0. \tag{3–26}$$

Note that combining (3–25) and (3–26) with the optimal $\boldsymbol{\beta}^*$, we have that $\gamma = 2$. Using this fact along with the non-negativity of $\boldsymbol{\mu}$, yield the following condition,

$$At_i - Bq_i > 1, \tag{3–27}$$

where $A = \frac{\lambda - 1}{\boldsymbol{\beta}^{*\top}\mathrm{V}\boldsymbol{\beta}^*}$ and $B = \frac{\lambda}{\boldsymbol{\beta}^{*\top}\mathrm{q}}$.

Let's partition the set of indexes of the entries of $\boldsymbol{\beta}$ into W, the working set, and P the complementary set of inactive elements. Then, $\boldsymbol{\beta} = \left(\boldsymbol{\beta}_W^\top,\ \boldsymbol{\beta}_P^\top\right)^\top$, for which we define the following subproblem:

$$\begin{aligned} \min_{\boldsymbol{\beta}_W} \big[&(\lambda - 1)\log\left(\boldsymbol{\beta}_W^\top\mathbf{V}_{WW}\boldsymbol{\beta}_W + 2\boldsymbol{\beta}_P^\top\mathbf{V}_{PW}\boldsymbol{\beta}_W + \boldsymbol{\beta}_P^\top\mathbf{V}_{PP}\boldsymbol{\beta}_P\right) + \\ &- 2\lambda\log\left(\boldsymbol{\beta}_W^\top\mathbf{q}_W + \boldsymbol{\beta}_P^\top\mathbf{q}_P\right)\big] \\ \text{subject to}\quad &-\boldsymbol{\beta}_W \leq 0, \quad \text{and} \quad \left[\boldsymbol{\beta}_W^\top\mathbf{1} + \boldsymbol{\beta}_P^\top\mathbf{1}\right] = 1 \end{aligned} \tag{3–28}$$

Similar remarks to the ones made in [55] can be obtained for (3–28):

- The terms $\varphi_A = \boldsymbol{\beta}_P^\top\mathbf{V}_{PP}\boldsymbol{\beta}_P$ and $\varphi_B = \boldsymbol{\beta}_P^\top\mathbf{q}_P$ are constant in the subproblem
- The computation of $2\boldsymbol{\beta}_P^\top\mathbf{V}_{PW}\boldsymbol{\beta}_W$ is independent of the size of P and also of the number of nonzero β_i's
- Replacing β_i, with $i \in W$ with β_j with $j \in P$ leaves the cost unchanged and the feasibility remains intact.
- If the subproblem is optimal before the above replacement, the new subproblem is optimal if and only if β_j satisfies the optimality conditions.

The so called "*Build down*" step is rather obvious. Now the "*Build up*" step that states that moving a variable from P to W gives a strict improvement in the cost when the subproblem is re-optimized. In our case, we can justify the build up since we prove that the KKT first order conditions are necessary and sufficient for a solution to (3–15). These build down and build up steps suggest a strategy for optimizing (3–15) by solving

smaller subproblems. At each iteration, solve a subproblem that includes a constraint violator picked from the complementary set P. Iterate until optimality conditions are satisfied up to some desired level of accuracy.

3.3.2.2 Sequential minimal optimization algorithm

In the previous section (3.3.2.1), the optimization problem related to the principle of information was decomposed into smaller subproblems that can be solved iteratively to achieve the solution of the full problem. An important characteristic of such decomposition is that the size of the working set W and the complementary set P are independent of the number of support vectors in the solution, that is, the β_i^*'s greater than zero. The sequential minimal optimization proposed in [60] chooses the smallest subproblem that can be solved at each iteration. This corresponds to solving for two variables at the time, which can be found analytically. The latter is of particular appeal to solve PRI since our cost does not have an standard form as it is the case for SVMs (quadratic program), therefore, we cannot resort to off the shelf solvers for our problem.

Without loss of generality, we will refer to our variables in the working set as β_1 and β_2 and the complementary set as P. By the equality constraint in the subproblem (3–28) we have that $\beta_1 + \beta_2 = 1 - \boldsymbol{\beta}_P^{\mathsf{T}}\mathbf{1} = w$ and thence $\beta_1 = w - \beta_2$. Let us denote by $\bar{\beta}_i$ the value of β_i from the previous iteration. We can formulate the subproblem in terms of β_2 as:

$$\begin{aligned} &\min_{\beta_2} \left[(\lambda - 1)\log A(\beta_2) - 2\lambda \log B(\beta_2)\right] \\ &\text{subject to} \quad 0 \leq \beta_2 \leq w, \quad \text{and} \quad w = \bar{\beta}_1 + \bar{\beta}_2, \end{aligned} \tag{3–29}$$

with

$$A(\beta_2) = \beta_2^2(V_{11} + V_{22} - 2V_{12}) + 2\beta_2(w(V_{12} - V_{11}) + (\upsilon_2 - \upsilon_1)) + w^2 V_{11} + 2w\upsilon_1 + \varphi_A$$

where $\upsilon_i = \mathbf{V}_i\bar{\boldsymbol{\beta}} - V_{1i}\bar{\beta}_1 - V_{1i}\bar{\beta}_2$, and $\varphi_A = \boldsymbol{\beta}_P{}^{\mathsf{T}}\mathbf{V}_{PP}\boldsymbol{\beta}_P$; and

$$B(\beta_2) = \beta_2(q_2 - q_1) + wq_1 + \varphi_B$$

where $\varphi_B = \boldsymbol{\beta}_P{}^{\top}\mathbf{q}_P$. The solution to problem (3–29) lies on the line segment $\beta_1 = w - \beta_2$ with $0 \leq \beta_2 \leq w$. Computing the derivative of the objective in (3–29) yields a second order polynomial on β_2 (Details are given in Appendix B.2), thus solving

$$c_2\beta_2^2 + c_1\beta_2 + c_0 = 0 \tag{3–30}$$

with coefficients:

$$\begin{aligned}
c_2 &= -2(V_{11} + V_{22} - 2V_{12})(q_2 - q_1) \\
c_1 &= 2(\lambda - 1)(wq_1 + \varphi_B)(V_{11} + V_{22} - 2V_{12}) + \\
&\quad -2(\lambda + 1)(w(V_{12} - V_{11}) + (\upsilon_2 - \upsilon_1))(q_2 - q_1) \\
c_0 &= 2(\lambda - 1)(w(V_{12} - V_{11}) + (\upsilon_2 - \upsilon_1))(wq_1 + \varphi_B) + \\
&\quad -2\lambda(w^2 V_{11} + 2w\upsilon_1 + \varphi_A)(q_2 - q_1)
\end{aligned}$$

conveys candidate solutions that ought to be checked along with the end points of the line segment. Let r_1 and r_2 be the roots of (3–30). Ruling out cases with complex numbers, we have:

$$L = \min\{r_1,\ r_2\} \quad \text{and} \quad U = \max\{r_1,\ r_2\}$$

the candidate solutions are,

$$s_1 = \begin{cases} 0 & L \leq 0 \\ L & 0 < L < w \\ w & L \geq w \end{cases} \quad \text{and} \quad s_2 = \begin{cases} 0 & U \leq 0 \\ U & 0 < U < w \\ w & U \geq w \end{cases} \tag{3–31}$$

If $s_1 \neq s_2$ we check $J(s_i) = [(\lambda - 1)\log A(s_i) - 2\lambda \log B(s_i)]$ and

$$\beta_2 = \arg \min_{s_i \in \{s_1, s_2\}} \{J(s_i)\} \tag{3–32}$$

otherwise, $\beta_2 = s_1 = s_2$.

3.3.2.3 SMO algorithm

The algorithm can be described into three basic steps:

Step 1: Initialization

$$\begin{aligned} \mathbf{q} &\leftarrow \mathbf{V}\boldsymbol{\alpha} \\ \mathbf{f} &\leftarrow \mathbf{q} \\ \boldsymbol{\beta} &\leftarrow \boldsymbol{\alpha} \\ IP(\boldsymbol{\beta}) &\leftarrow \boldsymbol{\beta}^{\mathsf{T}}\mathbf{q} \\ CIP(\boldsymbol{\beta}) &\leftarrow IP(\boldsymbol{\beta}) \end{aligned}$$

Step 2: Constants within an iteration

$$\begin{aligned} v_i &\leftarrow f_i - V_{1i}\bar{\beta}_1 - V_{2i}\bar{\beta}_2 \\ \varphi_A &\leftarrow IP(\bar{\boldsymbol{\beta}}) - (2(\bar{\beta}_2 f_1 + \bar{\beta}_2 f_2) - (\bar{\beta}_1^2 V_{11} + \bar{\beta}_2^2 V_{22} + 2\bar{\beta}_1 V_{12}\bar{\beta}_2)) \\ \varphi_B &\leftarrow CIP(\bar{\boldsymbol{\beta}}) - (\bar{\beta}_1 q_1 + \bar{\beta}_2 q_2) \\ w &\leftarrow \bar{\beta}_1 + \bar{\beta}_2 \end{aligned}$$

Step 3: Updates

$$\begin{aligned} \beta_2 &\leftarrow \text{solution described in (3–32)} \\ \beta_1 &\leftarrow w - \beta_2 \\ \mathbf{f} &\leftarrow \bar{\mathbf{f}} + (\bar{\beta}_2 - \beta_2)\mathbf{V}_1^{\mathsf{T}} + (\beta_2 - \bar{\beta}_2)\mathbf{V}_2^{\mathsf{T}} \\ IP(\boldsymbol{\beta}) &\leftarrow \varphi_A + (2(\beta_1 f_1 + \beta_2 f_2) - (\beta_1^2 V_{11} + \beta_2^2 V_{22} + 2\beta_1 V_{12}\beta_2)) \\ CIP(\boldsymbol{\beta}) &\leftarrow \varphi_B + (\beta_1 q_1 + \beta_2 q_2) \end{aligned}$$

Steps 2 and 3 are iterated for different working sets chosen according to some heuristics that are described below.

3.3.2.4 Selecting the working set

There are two types of constraint violations: an equality constraint (3–25) if $\beta_i > 0$, and the inequality constraint (3–27) if $\beta_i = 0$. The constraint violations are easy to

compute at each iteration. Let $\boldsymbol{\xi}$ be defined as

$$\boldsymbol{\xi} = 2\frac{\lambda - 1}{IP(\boldsymbol{\beta})}\mathbf{f} - 2\frac{\lambda}{CIP(\boldsymbol{\beta})}\mathbf{q}, \tag{3–33}$$

the constraint qualifications are $\xi_i = 2$ if $\beta_i > 0$, and $\xi_i > 2$ if $\beta_i = 0$. In the description of our algorithm, we chose to initialize $\boldsymbol{\beta}$ with the same values of $\boldsymbol{\alpha}$. However, our cost function suggest that points for which q_i is large will be expected to become support vectors, that is $\beta_i > 0$. We can then use $\boldsymbol{\beta} = \mathbf{q}/(\mathbf{q}^\top \mathbf{1})$ as the initial guess. However, this would imply the computation of $\mathbf{f}$ at the initialization. It is customary to choose $\boldsymbol{\alpha} = \frac{1}{N}\mathbf{1}$. Then at the initial iteration all constraints will be violated (unless $\lambda \to \infty$). One pass through the whole set taking pairs of indexes $(i,\ j)$, where i correspond to a descending order of the samples according to $\mathbf{f}$ and j's taken at random will create the first stage of sparseness in our weight vector $\boldsymbol{\beta}$; this is our first heuristic. After this pass, we can check whether (3–27) is satisfied for the current β_i's that are zero. A second stage suggest checking the within the set of samples with $\beta_i > 0$, and for which $\xi_i > 2$ since they are most likely to vanish. We will stop when conditions are fulfilled within ϵ tolerance.

3.3.3 Experiments

3.3.3.1 Synthetic data

Here, we are concerned with the computation of the principle of relevant information on large sample sizes. The purpose of this experimental setup is to observe the behavior of the algorithm in terms of λ which controls the number of nonzero weights and therefore the number of equality constraints that are much harder to satisfy. Data is obtained by sampling from a two dimensional mixture of three Gaussians with centers $(0,\ 0)$, $(3,\ 3)$, and $(-6,\ 4)$; spherical covariances $0.8^2\mathbf{I}$, $1.2^2\mathbf{I}$, and $\mathbf{I}$; and mixing proportions 0.2, 0.3, and 0.5, respectively. The kernel employed in our experiments is the Gaussian kernel $\kappa(x, y) = \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$, with $\sigma = 0.2$. Figure 3-6 depicts the computation times for different tolerance levels on the constraint violations as well as

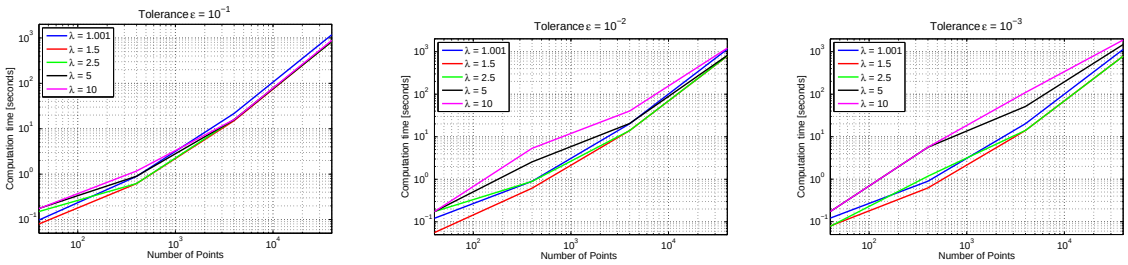


Figure 3-6. Computation times for different tolerance levels and sample sizes

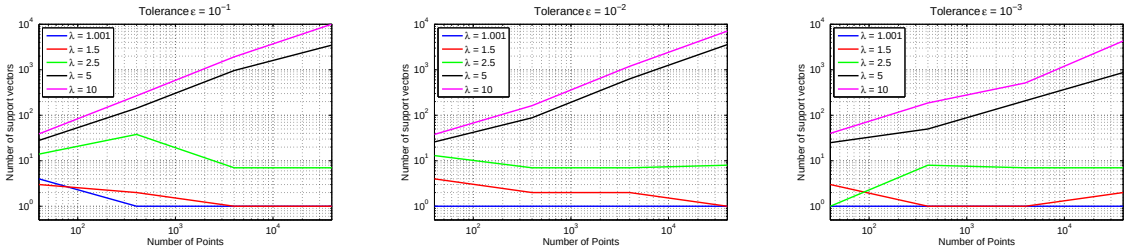


Figure 3-7. Number of support vectors for different tolerance levels and sample sizes

various sample sizes and trade off parameter λ. Figure 3-7 shows the final number of support vectors (nonzero weights) when the above mentioned parameters are varied.

First, notice that the kernel size σ was kept fixed regardless of the size of the sample. This allows for studying the algorithm behavior in terms of sparsity of data, which in this case corresponds to small sample sizes. The tolerance level has a clear effect on the computation time, but more interesting is the effect on the number of support vectors which reduces when the level of accuracy increases. On the small sample regime, the increment on the computation time due to the more demanding tolerance level $\epsilon = 10^{-3}$ can be attributed to the scarcity of data which makes the cost function much more sensitive to any change in the weight vector β. In terms of computational complexity, the algorithm behaves within the reasonable levels. In the experiments carried we bound the maximum number of iterations by $N \log N$. This upper bound was never attained by the larger sample sizes and only reached by small sample sizes on the most demanding scenarios, that is, small ϵ and large λ, since the

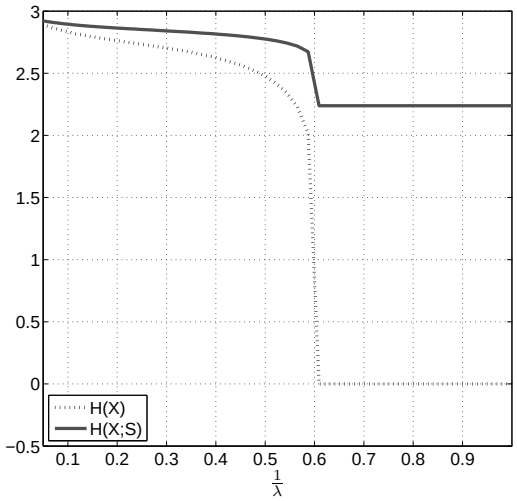


Figure 3-8. Entropy of the weighted sample and cross-entropy with respect to the original sample S for different values of λ. Notice that the values are plotted against $\frac{1}{\lambda}$ for better visualization.

trade off parameter λ is closely related with the number of support vectors and thus the proportion of constraint violators increases.

We also perform the illustrative experiments on the data shown in Figure 3-1. Figure 3-8 shows the different values of the entropy when the trade off parameter λ is varied between 1 and 20. The actual plot shows the inverse of λ for better visualization. In this case, we can observe two regimes; for $\lambda < 1.5$, all the importance is given to one point, which corresponds to the largest peak on the Parzen density estimate from the observed data S. Once λ becomes larger than 1.5 more points become active, and the weights progressively equalize across all data points. Figure 3-9 display the resulting support vectors found by the algorithm for different values of λ. Figure 3-10 shows in a gray intensities the distribution of the weights across S when $\lambda = 10$. Notice that the larger values are evenly distributed to account for the more densely populated regions of the data.

3.3.3.2 Image retrieval with partially occluded data MNIST

We employ a subset of the MNIST database to perform experiments on pattern retrieval from partially occluded queries. The weighting scheme for the principle of

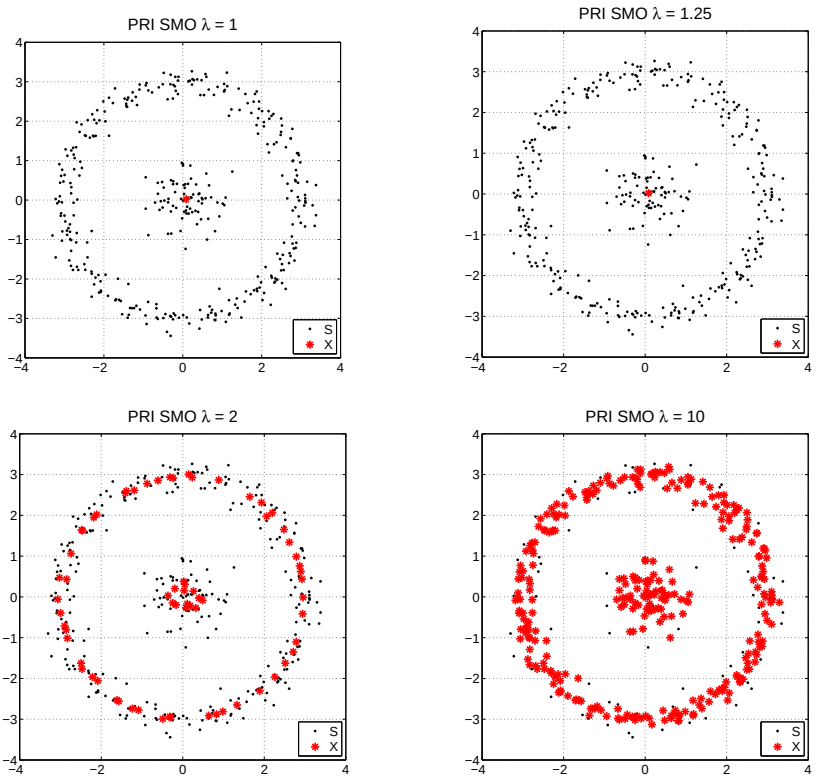


Figure 3-9. Resulting support vectors for different values of λ.

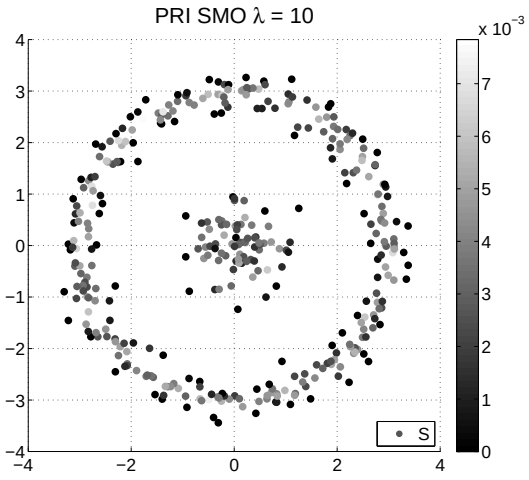


Figure 3-10. Distribution of the weights for $\lambda = 10$.

Table 3-1. Results for the image retrieval from partially occluded queries

n comp	3	8	15	30	50	100
KPCA	0.59(0.03)	0.55(0.015)	0.53(0.007)	0.50(0.007)	0.48(0.008)	0.47(0.006)
KECA	0.64(0.018)	0.57(0.018)	0.51(0.022)	0.49(0.013)	0.49(0.009)	0.48(0.007)
λ	2	3	4	5	6	8
PRI	0.58(0.013)	0.53(0.004)	0.51(0.007)	0.50(0.009)	0.49(0.01)	0.48(0.01)
n supp vectors	62.7	126	181	232	276	347

relevant information is applied to learn the representative samples from the training data. Results are compared against kernel PCA and kernel entropy component analysis for different number of components and different values of λ in the case of PRI. The pattern retrieval application requires pre-imaging of the patterns from the feature space back to the input space to apply KPCA KECA). We employ the method presented in [42] to compute the pre-images of the projected patterns in the KPCA and KECA subspaces (For more details on this problem see [37, 50]). The principle of relevant information requires a different approach since it does not provide an explicit projection method as it is the case for KPCA and KECA. A simple pre-imaging algorithm can be based on the Nadaraya-Watson kernel regression [51]. In our experiments, we use the Gaussian kernel. The preimaging for PRI consist of the following steps:

- Compute the k-nearest neighbors on the set of support vectors (training points with nonzero weights) of the query pattern x.
- Reconstruct using the following equation:

$$x_{\text{rec}} = \frac{\sum_{i=1}^{k} \kappa_\sigma(x_i, x) x_i}{\sum_{i=1}^{k} \kappa_\sigma(x_i, x)} \tag{3–34}$$

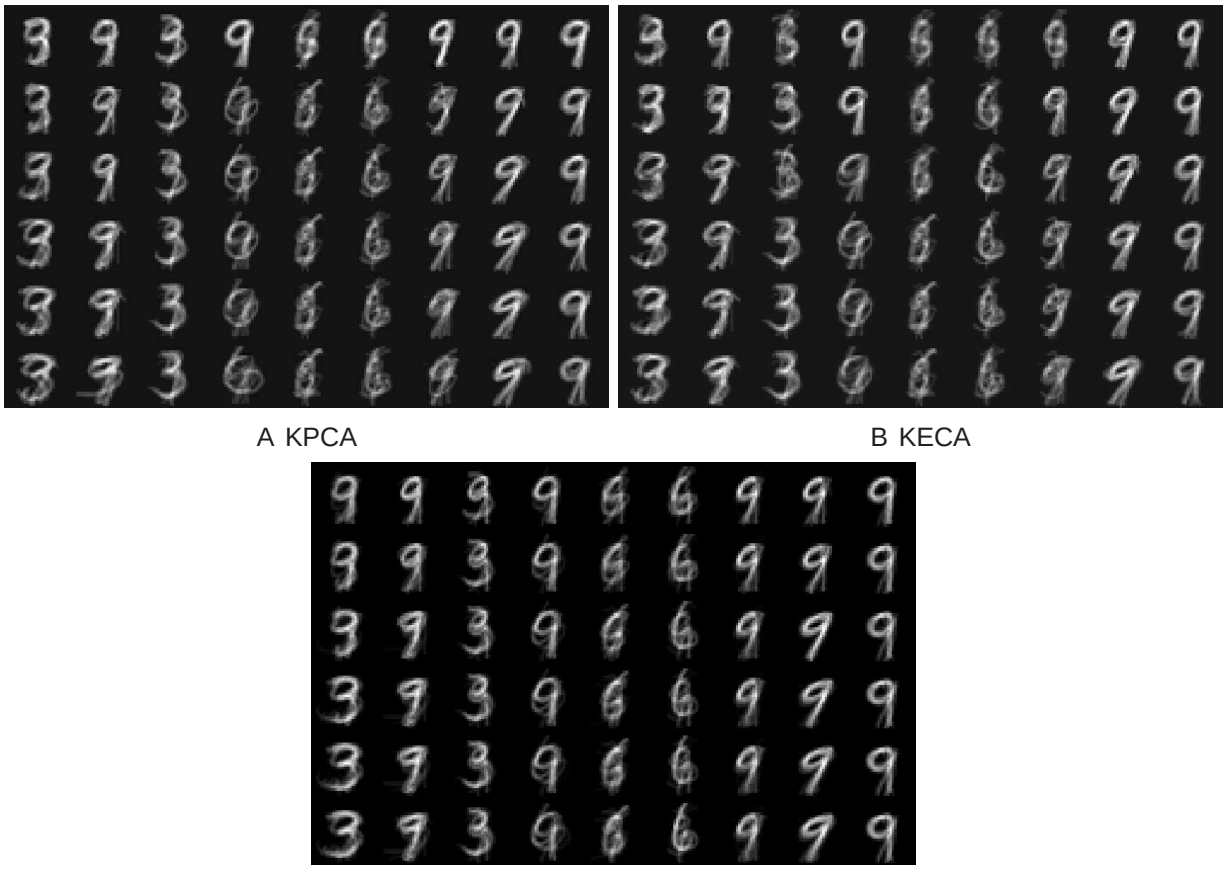
  where the indexes for x_i's are understood as the k nearest neighbors of x.

In this case, we want to see if we can effectively retrieve a digit by presenting an image with a missing portion. The model is trained with complete images and tested with images for which the lower half of the matrix has been replaced with zeros.

Table 3-1 shows the normalized SNR for the retrieved patterns using a KPCA KECA and the RKHS PRI. The numbers correspond to average of ten Montecarlo simulations

Figure 3-11. Sample queries with missing information and their corresponding complete versions

using 200 images per digit (total 600) for training and 50 incomplete images of each digit (150 total) for testing. We also present the average number of support vectors for the RKHS PRI. The standard deviations are also shown. The kernel size is 5, and the number of neighbors for pre-imaging is 10. Notice that differences between closest results for all methods and also the best performances are within the range set by $+$ $-$ the standard deviation. Figure 3-11 shows some of the exemplars that were used for testing along with the incomplete versions that were input to the algorithms. Figure 3-12A shows the reconstructed images for KPCA where each row corresponds to the number of components in Table 3-1. Likewise, Figure 3-12B displays the results for KECA, and Figure 3-12C show the results of the proposed approach for the different Lambda values in Table 3-1. Although the errors for all methods are very similar, the results for the PRI are more visually appealing.



A KPCA B KECA

C PRI

Figure 3-12. Retrieved patters from queries with missing information for different methods on the MNIST dataset.

3.4 The Information Potential RKHS Formulation of the PRI

For a positive semi definite kernel κ on $\mathcal{X}$, we can define a space of continuous bounded functionals on $\mathcal{X}$ as the completion of the span of all f's of the form

$$f(x) = \sum_{i \in \mathcal{A}} \alpha_i \kappa(x, x_i) \tag{3–35}$$

where $\alpha_i \in \mathbb{R}$ and $x_i \in \mathcal{X}$. This space is a Reproducing Kernel Hilbert space of continuous functions from $\mathcal{X}$. A similar development can be adopted for the information potentials introduced by [62]. For the set $\mathcal{F}$ of probability density functions that are square integrable in $\mathbb{R}^n$, we can define the cross-information potential $\mathcal{V}$ (CIP) as a

bilinear form that maps densities $f_i, f_j \in \mathcal{F}$ to the real numbers trough the integral,

$$\mathcal{V}(f_i, f_j) = \int_{\mathbb{R}^n} f_i(x) f_j(x) \mathrm{d}x \tag{3–36}$$

It is easy to see that for a basis of uniformly bounded, square integrable probability density functions, $\mathcal{V}$ defines a RKHS on the $\mathrm{span}\{\mathcal{F}\}$ (under completion). Now consider the set $\mathcal{G} = \{g = \sum_{i=1}^{m} \alpha_i \kappa_\sigma(x_i, \cdot) | x_i \in \mathbb{R}^n,\ \sum_{i=1}^{m} \alpha_i = 1, \text{ and } \alpha_i \geq 0\}$, where κ_σ is a "Parzen" type of kernel, that is κ_σ is symmetric, nonnegative, has bounded integral (can be normalized), and shift invariant with σ as the scale parameter; $\mathcal{V}$ also defines a RKHS $\mathcal{K}$ on $\mathcal{G}$. Clearly, for any $g \in \mathcal{G}$, we have $\|\mathcal{V}(g, \cdot)\|_{\mathcal{K}} \leq \|\kappa_\sigma(x, \cdot)\|_2$; therefore, $\mathcal{K}$ is a space of functionals on a bounded, albeit non-compact set.

The objective function for the principle of relevant information (Equation (D–8)) can be written in terms of the information potential RKHS. Using the Parzen based estimation, we restrict the search problem to $\mathcal{G} \subseteq \mathcal{F}$. In this case, we have that equation (D–8) can be rewritten as:

$$J(f) = -\log \mathcal{V}(f, f) - \lambda \log \frac{[\mathcal{V}(f, g)]^2}{\mathcal{V}(f, f)\mathcal{V}(g, g)} \tag{3–37}$$

straightforward manipulation of the terms yields an equivalent problem:

$$\arg\min_{f \in \mathcal{G}} [-(1 - \lambda) \log \mathcal{V}(f, f) - 2\lambda \log \mathcal{V}(f, g)] \tag{3–38}$$

or equivalently,

$$\arg\min_{f \in \mathcal{G}} \left[-(1 - \lambda) \log \|\phi(f)\|_{\mathcal{K}}^2 - 2\lambda \log \langle \phi(f), \phi(g) \rangle_{\mathcal{K}}\right], \tag{3–39}$$

where $\phi : \mathcal{G} \mapsto \mathcal{K}$ is an underlying mapping such that $\mathcal{V}(f, g) = \langle \phi(f), \phi(g) \rangle_{\mathcal{K}}$[2] .
Two important aspects of the above objective are: the choice of the kernel, shape and

[2] This is commonly known as the feature map within the machine learning community.

size σ, determines different spaces of functionals (RKHS's); the trade-off parameter λ defines a set of regimes for the possible solutions to the problem. Note however that there is no difference between (D–8) or (3–39) in terms of the space being searched (it is $\mathcal{G}$ for both). Therefore, the core question is whether we can perform the search in the IP RKHS. Let us focus on the case $\lambda > 1$. Consider an element $F \in \mathcal{K}$, the term $-\log \langle F, \phi(g) \rangle_{\mathcal{K}}$ will play the role of a risk functional R_g evaluated at F. Now, since we are not provided with the function g, we take empirical estimator of R_{emp}:

$$R_{\text{emp}}(F) = -2 \log \frac{1}{N} \sum_{i=1}^{N} \langle F, \phi(\kappa_\sigma(x_i, \cdot)) \rangle_{\mathcal{K}} \tag{3–40}$$

This quantity comes from mapping the empirical distribution after being convolved with the Parzen kernel κ_σ. Nonetheless, in order for this functional F to be valid as an estimator of a information theoretic quantity, we also need:

$$\langle F, \phi(\kappa_\sigma(x, \cdot)) \rangle_{\mathcal{K}} \geq 0, \quad \forall x \in \mathbb{R}^n \tag{3–41}$$

Solution to the constrained problem in the IP RKHS: In the case where $\lambda > 1$, we have the following constrained problem (for notational convenience we will denote $\phi(\kappa_\sigma(x_i, \cdot))$ by $\phi(\kappa_i)$):

$$\begin{aligned} &\min_{F \in \mathcal{K}} \left[(\lambda - 1) \log \|F\|_{\mathcal{K}}^2 - 2\lambda \log \frac{1}{N} \sum_{i=1}^{N} \langle F, \phi(\kappa_i) \rangle_{\mathcal{K}} \right] \\ &\text{s.t. } \langle F, \phi(\kappa_i) \rangle_{\mathcal{K}} \geq 0, \quad \forall x \in \mathbb{R}^n. \end{aligned} \tag{3–42}$$

Hence, (3–42) can be seen as a regularized risk minimization problem

$$\min_{F \in \mathbf{K}} \left[R_{\text{emp}}(F) + \frac{(\lambda - 1)}{\lambda} \Omega \left(\|F\|_{\mathcal{K}}^2 \right) \right],$$

where **K** denotes the set of feasible points in $\mathcal{K}$. The evaluation of the constraint in (3–42) that requires nonnegativity of the inner product in $\mathcal{K}$ for all $x \in \mathbb{R}^n$ will be relaxed by evaluating the condition only at x_i in the sample S. This yields the following

formulation,

$$\begin{aligned} &\min_{F\in\mathcal{K}} \left[(\lambda-1)\log\|F\|_{\mathcal{K}}^2 - 2\lambda \log \frac{1}{N}\sum_{i=1}^{N}\xi_i \right] \\ &\text{s.t. } \langle F, \phi(\kappa_i)\rangle_{\mathcal{K}} \geq \xi_i \\ &\quad \xi_i \geq 0 \end{aligned} \tag{3–43}$$

with Lagrangian function $L(F, \boldsymbol{\xi}, \boldsymbol{\beta}, \boldsymbol{\mu})$:

$$\begin{aligned} (\lambda-1)\log\|F\|_{\mathcal{K}}^2 - 2\lambda\log\frac{1}{N}\sum_{i=1}^{N}\xi_i + \\ - \sum_{i=1}^{N}\beta_i\left(\langle F, \phi(\kappa_i)\rangle_{\mathcal{K}} - \xi_i\right) - \sum_{i=1}^{N}\mu_i\xi_i, \end{aligned} \tag{3–44}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$ are the multipliers. Setting derivatives with respect to F and $\boldsymbol{\xi}$ to zero yields:

$$\begin{aligned} F &= \frac{\|F\|_{\mathcal{K}}^2}{2(\lambda-1)}\sum_{i=1}^{N}\beta_i\phi(\kappa_i) \\ \|F\|_{\mathcal{K}}^2 &= \frac{2(\lambda-1)}{\sum_{i,j=1}^{N}\beta_i\beta_j\langle\phi(\kappa_i),\phi(\kappa_j)\rangle_{\mathcal{K}}}\sum_{i=1}^{N}\beta_i\xi_i \\ 0 &\leq \mu_i = \beta_i + \frac{2\lambda}{\sum_{i=1}^{N}\xi_i}. \end{aligned} \tag{3–45}$$

Replacing (3–45) in the Lagrangian (3–44), we obtain the dual problem

$$\begin{aligned} &\min_{\boldsymbol{\beta}\in\mathbb{R}^N} (\lambda-1)\log\sum_{i,j=1}^{N}\beta_i\beta_j\langle\phi(\kappa_i),\phi(\kappa_j)\rangle_{\mathcal{K}} \\ &\text{s.t. } \ 2\lambda\log\frac{1}{N}\sum_{i,j=1}^{N}\beta_j\langle\phi(\kappa_i),\phi(\kappa_j)\rangle_{\mathcal{K}} = \eta, \\ &\quad \boldsymbol{\beta}\geq 0, \ \text{ and } \ \sum_{i=1}^{N}\beta_i = 1 \end{aligned} \tag{3–46}$$

CHAPTER 4
ESTIMATING ENTROPY-LIKE QUANTITIES WITH INFINITELY DIVISIBLE KERNELS

The operational quantities in information theory are based on the probability laws underlying the data generation process, but these are rarely or never known in the statistical learning setting were the only information available comes from a sample $\{z_i\}_i = 1^N$. The use of Renyi's definition of entropy along with Parzen density estimation were proposed as the main tool to work learning problems in term of information theory quantities such entropy and relative entropy [62]. Renyi's entropy of order α is a generalization of Shannon's entropy by relaxing the condition of additivity of entropy to the generalized ψ mean[1] . The entropy of a random variable X as a function of the parameter α is given by,

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int_{\mathcal{X}} p^{\alpha-1}(x) p(x) \mathrm{d}x \tag{4–1}$$

where p is the probability density function (continuous case) or the probability mass function (taking the integral as a sum) of the random variable X with support $\mathcal{X}$. The limiting case $\alpha \to 1$ brings back Shannon's entropy. A plug-in estimator of (4–1) for $\alpha = 2$ can be derived using Parzen window approximation. For a sample $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$, an estimator of Rényi's 2-order entropy based on Parzen window with Gaussian kernel of size σ is given by:

$$\hat{H}_2(X) = -\log \frac{1}{n^2} \sum_{i,j=1}^{n} \kappa_{\sqrt{2}\sigma}(x_i, x_j) \tag{4–2}$$

where $\kappa_\sigma(x, y) = C \exp\left(-\frac{1}{2\sigma^2}\|x - y\|^2\right)$ and C is a normalization constant. An important conceptual element derived from this framework is the functional that corresponds to the argument of the $\log$, which has been called information potential in analogy to the potential fields arising in physics [62]. More recent work has shown that the

[1] $\langle X \rangle_\psi = \psi^{-1}\left(\sum_{i=1}^n w_i \psi(x_i)\right)$, where $\sum w_i = 1$, $w_i \geq 0$ and ψ is a Kolmogorov-Nagumo function

information potential is a special case of a positive definite kernel called the cross information potential between two pdfs that maps probability density functions in L_2 to a Reproducing kernel Hilbert Space of functions [95]. Moreover, this framework has been already exploited to solve optimization problems with information theoretic objective functions that bear close resemblance to kernel methods [71]. These recent insights motivate views that go beyond the Parzen density estimation that initially motivated its study.

In this work, we show how by using positive definite kernels with specific properties, we can obtain entropy-like quantities without assuming that probabilities of events are known or have been estimated. Other approaches with similar motivation based on the concept of entropic graphs have been recently addressed in [15, 31, 64]. However, one of the main disadvantages the quantities proposed by these graph based formulation are not well suited for adaptation schemes since they are not differentiable. This is not the case for the proposed framework as we show in Chapter 5. We will follow the statistical learning setting where the only available information is contained in a finite *i.i.d.* sample $Z = \{z_i\}_{i=1}^n$. In this sense we think about entropy as a measure of the lack (uncertainty) or presence (structure) of statistical regularities in a given sample represented by the a Gram matrix. From the axiomatic characterization of entropy that leads to Rényi's definition, we develop an analogue version of this function that is applied to positive definite matrices. Then, we look at some basic inequalities of information and how they can be translated to the setting of positive definite matrices. The purpose of this characterization is to establish some desirable properties on the positive definite kernels that can be employed to construct the Gram matrix for which our extension of entropy makes sense in terms of information.

4.1 Motivation

The use of Hilbert spaces to represent data is not a new idea [1, 2], and it has become of common practice in machine learning under the name of kernel methods.

Kernel methods provide an appealing framework to deal with data of different nature by embedding abstract set in reproducing kernel Hilbert spaces, where it is possible to carry out manipulations of the representation of data by the operations of addition, scalar multiplication and inner product. This allows one to deal with algorithms in a rather generic way provided the kernel is well-fitted to the particular problem. These property has been recently exploited in many practical applications where data is not necessarily given as vector in $\mathbb{R}^p$, for example text, trees, point processes, functional data, among others [82]. It has been noticed that kernel induced map can be understood as a mean for computing high order statistics of the data and manipulating them in a linear fashion as first order statistics. Methods such as kernel independent component analysis [4], the work on measures of dependence and independence using Hilbert-Schmidt norms [29], and recent work on quadratic measures of independence [79] are just among the examples of this emerging line. As we shall see, the Gram matrix plays a fundamental role in establishing the connection between information theoretic concepts and kernel methods.

4.1.1 Hilbert Space Representation of Data

In this section, we want to motivate the use of positive definite matrices as suitable descriptors of data. For this, we need to understand the role of the Hilbert space representation and how it naturally arises from the fundamental ideas of pattern analysis. Let $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ be the object space, this with σ-algebra $\mathcal{B}_\mathcal{X}$ and a probability measure $P_\mathcal{X}$ defined on it. A function $\phi : \mathcal{X} \mapsto \mathbb{R}$ is called a feature. A representation is a family of features $\{\phi_t\}_{t\in\mathcal{T}}$, where $(\mathcal{T}, \mathcal{B}_\mathcal{T}, \mu_\mathcal{T})$ is a measure space and $\mu_\mathcal{T}$ is σ-finite. Let ϕ_t be also bounded for all $t \in \mathcal{T}$, and let us denote $\phi_t(x)$ by $\phi(t, x)$ where $t \in \mathcal{T}$ and $x \in \mathcal{X}$. If we also require that for all fixed x and y in $\mathcal{X}$,

$$G(x, y) = \int_\mathcal{T} \phi(t, x)\phi(t, y)\mathrm{d}\mu_\mathcal{T}(t) < \infty. \tag{4–3}$$

Then, the set $\mathcal{F}$ defined as the completion of the set of functions F of the form,

$$F(t) = \sum_{i=1}^{N} \alpha_i \phi(t, x_i), \tag{4–4}$$

where $\alpha_i \in \mathbb{R}$, $x_i \in \mathcal{X}$, and $\forall N \in \mathbb{N}$ [2] . The space $\mathcal{F}$ is a Hilbert space representation of the set $\mathcal{X}$. Nevertheless, dealing explicitly with such an $\mathcal{F}$ may be difficult if not impossible for practical purposes. The following result gives an alternative way to deal with the problem based on the bivariate function $G(x, y)$ defined above. Consider the set of functions on $\mathcal{X}$ of the form

$$f(x) = \sum_{i=1}^{N} \alpha_i G(x, x_i), \tag{4–5}$$

where $\alpha_i \in \mathbb{R}$, $x_i \in \mathcal{X}$, and $\forall N \in \mathbb{N}$. Let us define the inner product between elements $f = \sum_{i=1}^{N} \alpha_i G(x, x_i)$ and $g = \sum_{j=1}^{M} \beta_i G(x, x_j)$ of the above set as:

$$\langle f, g \rangle = \sum_{i=1}^{N} \sum_{j=1}^{M} \alpha_i \beta_j G(x_i, x_j), \tag{4–6}$$

the completion of the above set is a Hilbert space $\mathcal{H}$ of functions on $\mathcal{X}$. Moreover, $\mathcal{H}$ is a reproducing kernel Hilbert space with kernel G. Notice that for any finite set $\{x_i\}_{i=1}^{N}$ we have that

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j G(x_i, x_j) \geq 0, \tag{4–7}$$

for all $\boldsymbol{\alpha} \in \mathbb{R}^N$. Functions satisfying the above condition are called positive definite.

Theorem 4.1.1. (Basic Congruence Theorem [58]): *Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two abstract Hilbert spaces. Let $\mathcal{X}$ be an index set. Let $\{F(x),\ x \in \mathcal{X}\}$, be a family of vectors which span $\mathcal{H}_1$. Similarly, let $\{f(x),\ x \in \mathcal{X}\}$ be a family of vectors which span $\mathcal{H}_2$. Suppose*

[2] Even though is not explicitly stated, we assume the construction of a linear space of real functions with domain $\mathcal{T}$

that, for every x and y in $\mathcal{X}$,

$$\langle F(x), F(y)\rangle_1 = \langle f(x), f(y)\rangle_2 \tag{4–8}$$

Then the spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ are congruent, and one can define a congruence Ψ from $\mathcal{H}_1$ to $\mathcal{H}_2$ which has the property that $\Psi(F(x)) = f(x)$ for $x \in \mathcal{X}$.

Proposition 4.1.1. *Let $\mathcal{X}$ be a compact space. The spaces $\mathcal{F}$ and $\mathcal{H}$ are congruent.*

Proof 4.1.1. *The congruence follows from definition of $\mathcal{F}$ and $\mathcal{H}$. For $F = \sum_{i=1}^{n} \alpha_i \phi(t, x_i)$ simply take $\Psi : \mathcal{F} \mapsto \mathcal{H}$ as:*

$$\Psi(F) = \sum_{i=1}^{n} \alpha_i \int_{\mathcal{X}} \phi(t, x_i)\phi(t, \cdot)\mathrm{d}\mu_{\mathcal{T}}(t) = f$$

□

The above proposition allows us to perform the analysis of the representation of $\mathcal{X}$ on the equivalence classes that can be formed by using the function G to define relations between the elements of $\mathcal{X}$. From the congruence, we can define a distance function between the representations of two elements $x, y \in \mathcal{X}$ using the function G as follows:

$$d^2(\phi(t, x), \phi(t, y)) = G(x, x) + G(y, y) - 2G(x, y), \tag{4–9}$$

for convenience we write $d^2(\phi(t, x), \phi(t, y))$ as $d^2(x, y)$.

4.1.2 The Cross-Information Potential RKHS

For the set $\mathcal{F}$ of probability density functions that are square integrable in $\mathbb{R}^n$, we can define the cross-information potential $\mathcal{V}$ (CIP) as a bilinear form that maps densities $f_i, f_j \in \mathcal{F}$ to the real numbers trough the integral,

$$\mathcal{V}(f_i, f_j) = \int_{\mathbb{R}^n} f_i(x) f_j(x)\mathrm{d}x \tag{4–10}$$

It is easy to see that for a basis of uniformly bounded, square integrable probability density functions, $\mathcal{V}$ defines a RKHS on the $\mathrm{span}\{\mathcal{F}\}$ (up to completion). Now consider the set $\mathcal{G} := \{g = \sum_{i=1}^{m} \alpha_i \kappa_\sigma(x_i, \cdot) | x_i \in \mathbb{R}^n,\ \sum_{i=1}^{m} \alpha_i = 1, \text{ and } \alpha_i \geq 0\}$, where κ_σ is a

"Parzen" type of kernel, that is κ_σ is symmetric, nonnegative, has bounded integral (can be normalized), and shift invariant with σ as the scale parameter; $\mathcal{V}$ also defines a RKHS $\mathcal{K}$ on $\mathcal{G}$. Clearly, for any $g \in \mathcal{G}$, we have $\|\mathcal{V}(g,\cdot)\|_{\mathcal{K}} \leq \|\kappa_\sigma(x,\cdot)\|_2$; therefore, $\mathcal{K}$ is a space of functionals on a bounded, albeit non-compact set. Notice that the cross information potential, by definition, is a positive definite function that is data dependent, and differs from the instance-based representation in this respect. Nevertheless, the empirical estimator (4–2) links both Hilbert space representations. If we construct the Gram matrix $\mathbf{K}$ with elements $K_{ij} = \kappa_\sigma(x_i, x_j)$, it is easy to verify that (4–2) corresponds to:

$$\hat{H}_2(X) = -\log\left(\frac{1}{n^2}\mathrm{tr}(\mathbf{KK})\right). \tag{4–11}$$

As we can see, the estimator if the information potential can be related to the norm of the Gram matrix $\mathbf{K}$ defined as $\|\mathbf{K}\|^2 = \mathrm{tr}(\mathbf{KK})$. In the next section, we extend this concept to other norms and show how the properties of Renyi's definition of entropy carry on.

4.2 Positive Definite Matrices, and Renyi's Entropy Axioms

Hermitian matrices are considered as generalizations of real numbers. It is possible to define a partial ordering on this set by using positive definite matrices; for two Hermitian matrices $A, B \in M_n$, we say $A \succcurlyeq B$ if $A - B$ is positive definite. Likewise, $A \succ B$ means that $A - B$ is strictly positive definite. The following spectral decomposition theorem relates to the functional calculus on matrices and provides a reasonable way to extend continuous scalar-valued functions to Hermitian matrices.

Theorem 4.2.1. *Let $D \subset \mathbb{C}$ be a given set and let $\mathcal{N}_n(D) := \{A \in M_n : A \text{ is normal and } \sigma(A) \subset D\}$. If $f(t)$ is a continuous scalar-valued function on D, then the primary matrix function*

$$f(A) = U \begin{pmatrix} f(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f(\lambda_n) \end{pmatrix} U^* \tag{4–12}$$

is continuous on $\mathcal{N}_n(D)$, where $A = U\Lambda U^*$, $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$, and $U \in M_n$ is unitary.

Now we are ready to define a matrix analogue to Renyi's entropy that will be applied to Gram matrices constructed using a positive definite function.

Consider the set Δ_n^+ of positive definite matrices in M_n for which $\mathrm{tr}(A) \leq 1$. It is clear that this set is closed under finite convex combinations.

Proposition 4.2.1. *Let $A \in \Delta_n^+$ and $B \in \Delta_n^+$ and also $\mathrm{tr}(A) = \mathrm{tr}(B) = 1$. The functional*

$$S_\alpha(A) = \frac{1}{1-\alpha} \log_2 (\mathrm{tr} A^\alpha), \tag{4–13}$$

satisfies the following set of conditions:

(i) $S_\alpha(PAP^*) = S_\alpha(A)$ *for any orthonormal matrix* $P \in M_n$

(ii) $S_\alpha(pA)$ *is a continuous function for* $0 < p \leq 1$.

(iii) $S_\alpha(\frac{1}{n} I) = \log_2 n$ *(entropy is exhaustive).*

(iv) $S_\alpha(A \otimes B) = S_\alpha(A) + S_\alpha(B)$.

(v) If $AB = BA = \mathbf{0}$*; then for the strictly monotonic and continuous function* $g(x) = 2^{(\alpha-1)x}$ *for* $\alpha \neq 1$ *and* $\alpha > 0$*, we have that:*

$$\begin{aligned} S_\alpha(tA + (1-t)B) =& g^{-1} (tg(S_\alpha(A)) + \\ &+ (1-t) g(S_\alpha(B))) . \end{aligned} \tag{4–14}$$

Proof 4.2.1. *The proof of (i) easily follows from Theorem 4.2.1. Take $A = U\Lambda U^*$ now PU is also a unitary matrix and thus $f(A) = f(PAP^*)$ the trace functional is invariant under unitary transformations. For (ii), the proof reduces to the continuity of $\frac{1}{1-\alpha} \log_2(p)^\alpha$. For (iii), a simple calculation yields $\mathrm{tr} A^\alpha = \left(\frac{1}{n}\right)^{\alpha-1}$. Now, for property (iv), notice that if $\mathrm{tr} A = \mathrm{tr} B = 1$, then, $\mathrm{tr}(A \otimes B) = 1$. Since $A = U\Lambda U^*$ and $B = V\Gamma V^*$ we can write $A \otimes B = (U \otimes V)(\Lambda \otimes \Gamma)(U \otimes V)^*$, from which $\mathrm{tr}(A \otimes B)^\alpha = \mathrm{tr}(\Lambda \otimes \Gamma)^\alpha = \mathrm{tr}(\Lambda^\alpha)\mathrm{tr}(\Gamma^\alpha)$ and thus (iv) is proved. Finally, (v) notice that for any integer power k of $tA + (1-t)B$ we have: $(tA + (1-t)B)^k = (tA)^k + ((1-t)B)^k$ since $AB = BA = \mathbf{0}$. Under extra conditions such as $f(0) = 0$ the argument in the proof of Theorem 4.2.1 can be extended to this case. Since the eigen-spaces for the non-null eigenvalues of A and B*

are orthogonal we can simultaneously diagonalize A and B with the orthonormal matrix U, that is $A = U\Lambda U^*$ and $B = U\Gamma U^*$ where Λ and Γ are diagonal matrices containing the eigenvalues of A and B respectively. Since $AB = BA = \mathbf{0}$, then $\Lambda\Gamma = \mathbf{0}$. Under the extra condition $f(0) = 0$, we have that $f(tA + (1-t)B) = f(tA) + f((1-t)B)$ yielding the desired result for (v).

□

Notice also that if $\rho(A) = 1$, that is A is rank one matrix, $S_\alpha = 0$ for $\alpha \neq 0$.

The following important property is also true.

Proposition 4.2.2. *Let $A \in \Delta_n^+$, and $\mathrm{tr}(A) = 1$. For $\alpha > 1$*

$$S_\alpha(A) \leq S_\alpha(\frac{1}{n}I) \tag{4–15}$$

Proof 4.2.2. *Let $\{\lambda_i\}$ be the set of eigenvalues of A. Then we have that,*

$$\begin{aligned}
S_\alpha(A) - S_\alpha(\tfrac{1}{n}I) &= \frac{1}{1-\alpha}\log_2\left[\frac{\mathrm{tr}(AA^{\alpha-1})}{n^{-(\alpha-1)}}\right]; && (4–16)\\
&= \frac{1}{1-\alpha}\log_2\left[\mathrm{tr}\big(A(nA)^{\alpha-1}\big)\right]; && (4–17)\\
&= \frac{1}{1-\alpha}\log_2\left[\sum_i \lambda_i f_\alpha(n\lambda_i)\right]; && (4–18)\\
&\leq \frac{1}{1-\alpha}\sum_i \lambda_i \log_2 f_\alpha(n\lambda_i); && (4–19)\\
&= -\sum_i \lambda_i \log_2 \frac{\lambda_i}{\frac{1}{n}}; && (4–20)\\
&\leq \log_2\left[\sum_i \lambda_i \frac{\frac{1}{n}}{\lambda_i}\right] = 0. && (4–21)
\end{aligned}$$

Where (4–19) and (4–21) are due to Jensen's inequality.

□

However, such a characterization may not be enough to tell what unit-trace positive definite matrices have an information theoretic interpretation. For example consider the matrix $\mathbf{L} = \frac{1}{n}\mathbf{l}\mathbf{l}^\top$ where n_1 of the entries are 1 and the remaining $n - n_1$ entries are -1. This can be seen as a vector encoding classes. Notice that, evaluating the entropy

functional defined above yields 0 for any α. However, with a two-column matrix $\mathbf{M}$, for which the columns represent the class and the rows the samples, that encodes the class memberships as $M_{ij} = 1$ if the i-th sample belongs to the j-th class, and 0 otherwise, we obtain a more reasonable quantity using $\mathbf{L} = \mathbf{M}\mathbf{M}^{\top}$, related to a binomial distribution with $p = \frac{n_1}{n}$. Interestingly, we can simple relate $\mathbf{I}$ and $\mathbf{M}$ by $\mathbf{I} = \mathbf{M}\,(1, -1)^{\top}$. In the following sections, we will address this issue by considering a particular context in which Hadamard products between positive definite matrices arise.

4.2.1 Entropy inequalities for Hadamard Products

In the properties listed above in Propositions 4.2.1 and 4.2.2, we did not considered Hadamard products of positive definite matrices. This product may be of interest in the case we have two matrices A and B in Δ_n with unit trace where there exists some relation between the elements A_{ij} and B_{ij} for all i and j. The Hadamard product can be useful in developing analogues to joint entropies, where each one the matrices involved in the Hadamard product represents a random variable. Before we present the main result of this part of the section, we need to introduce the concept of majorization and some results pertaining the ordering that arises from this definition.

Definition 4.2.1. (Majorization): *Let p and q be two nonnegative vectors in $\mathbb{R}^n$ such that $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i < \infty$. We say $p \preccurlyeq q$, q majorizes p, if their respective ordered sequences $p_{[1]} \geq p_{[2]} \geq \cdots \geq p_{[n]}$ and $q_{[1]} \geq q_{[2]} \geq \cdots \geq q_{[n]}$ denoted by $\{p_{[i]}\}_{i=1}^{n}$ and $\{p_{[i]}\}_{i=1}^{n}$, satisfy:*

$$\sum_{i=1}^{k} p_{[i]} \leq \sum_{i=1}^{k} q_{[i]} \ \text{ for } \ k = 1, \ldots, n \tag{4–22}$$

It can be shown that if $p \preccurlyeq q$ then $p = Aq$ for some doubly stochastic matrix A [11]. It is also easy to verify that if $p \preccurlyeq q$ and $p \preccurlyeq h$ then $p \preccurlyeq tq + (1-t)h$ for $t \in [0, 1]$. The majorization order is important because it can be associated with the definition of *Schur-concave (convex)* functions. A real valued function f on $\mathbb{R}^n$ is called Schur-convex if $p \preccurlyeq q$ implies $f(p) \leq f(q)$ and Schur-concave if $f(q) \leq f(p)$.

Lemma 4.2.1. *The function $f_\alpha : \mathcal{S}^n \mapsto \mathbb{R}_+$ ($\mathcal{S}^n$ denotes the n dimensional simplex), defined as,*

$$f_\alpha(p) = \frac{1}{1-\alpha} \log_2 \sum_{i=1}^{n} p_i^\alpha, \tag{4–23}$$

is Schur-concave for $\alpha > 0$.

Notice that, Schur-concavity (Schur-convexity) cannot be confused with concavity (convexity) of a function in the usual sense. Now, we are ready to state the inequality for Hadamard products.

Proposition 4.2.3. *Let A and B be two $n \times n$ positive definite matrices with trace 1 with nonnegative entries, and $A_i i = \frac{1}{n}$ for $i = 1, 2, \dots, n$. Then, the following inequalities hold:*

(i)

$$S_\alpha\left(\frac{A \circ B}{\mathrm{tr}(A \circ B)}\right) \geq S_\alpha(B), \tag{4–24}$$

(ii) and

$$S_\alpha\left(\frac{A \circ B}{\mathrm{tr}(A \circ B)}\right) \leq S_\alpha(A) + S_\alpha(B). \tag{4–25}$$

Proof 4.2.3. *In proving (4–24) and (4–25), we will use the fact that S_α preserves the majorization order (inversely) of nonnegative sequences on the n-dimensional simplex. First look at the identity*

$$x^\top (A \circ B) x = \mathrm{tr}(A D_x B D_x) = \frac{1}{n}$$

In particular, if $\{x_i\}_{i=1}^n$ is an orthonormal basis for $\mathbb{R}^n$,

$$\mathrm{tr}(A \circ B) = \sum_{i=1}^{n} x_i^\top (A \circ B) x_i$$

If we let $\{x_i\}_{i=1}^n$ *be the eigenvectors of* $A \circ B$ *ordered according to their respective eigenvalues in decreasing order, then,*

$$\begin{aligned}\sum_{i=1}^{k} x_i^\top (A \circ B) x_i &= \sum_{i=1}^{k} \operatorname{tr}\left(A D_{x_i} B D_{x_i}\right) \\ &\leq \frac{1}{n} \sum_{i=1}^{k} \operatorname{tr}\left(\mathbf{1}\mathbf{1}^\top D_{x_i} B D_{x_i}\right) \\ &= \frac{1}{n} \sum_{i=1}^{k} x_i^\top B x_i \\ &\leq \frac{1}{n} \sum_{i=1}^{k} y_i^\top B y_i, \end{aligned} \tag{4–26}$$

where $k = 1, \ldots, n$ *and* $\{y_i\}_{i=1}^n$ *are the eigenvectors of* B *ordered according to their respective eigenvalues in decreasing order. The inequality* (4–26) *is equivalent to say that* $n\lambda(A \circ B) \preccurlyeq \lambda(B)$*, that is, the sequence of eigenvalues of* $(A \circ B)/\operatorname{tr}(A \circ B)$ *is majorized by the sequence of eigenvalues of* B*, which implies* (4–24)*.*

To prove (4–25) *notice that for* A *we have two extreme cases* $A = \frac{1}{n} I$ *and* $A = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$*. Taking* $A = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ *we have that*

$$\sum_{i=1}^{k} \lambda_i(B) = n \sum_{i=1}^{k} \frac{1}{n} \operatorname{tr}\left(\mathbf{1}\mathbf{1}^\top D_{x_i} B D_{x_i}\right) = \sum_{i=1}^{k} \lambda_i \left(\frac{A \circ B}{\operatorname{tr}(A \circ B)}\right) \tag{4–27}$$

the other extreme case where $A = \frac{1}{n} I$ *we have,*

$$\frac{1}{n} \sum_{i=1}^{k} \lambda_i(B) \leq \frac{1}{n} \leq n \sum_{i=1}^{k} \frac{1}{n} d_i(B) = \sum_{i=1}^{k} \lambda_i \left(\frac{A \circ B}{\operatorname{tr}(A \circ B)}\right) \tag{4–28}$$

where $\{\lambda_i(X)\}$ *are the eigenvalues of* X *in decreasing order and* $\{d_i(X)\}$ *are the elements of the diagonal of* X *ordered in decreasing order. The inequalities* (4–27) *and* (4–28) *imply 4–25*

□

4.2.2 The Tensor and Hadamard Product Entropy Gap

The mutual information of a pair of random variables X and Y can be seen as the gain of information from assuming X and Y to be independent to knowing the joint probability distribution. In other words, the amount of uncertainty reduced from from knowing the marginal distributions to knowing the joint distribution. In the Shannon definition this information gain can be expressed as:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{4–29}$$

where $H(X)$ and $H(Y)$ are the marginal entropies of X and Y, and $H(X,Y)$ is their joint entropy. In analogy we can compute the quantity:

$$I_\alpha(A;B) = S_\alpha(A) + S_\alpha(B) - S_\alpha\left(\frac{A \circ B}{\mathrm{tr}(A \circ B)}\right) \tag{4–30}$$

for positive semidefinite A and B with nonnegative entries and unit trace such that $A_{ii} = \frac{1}{n}$ for all $i = 1, \dots, n$. Notice that the above quantity is nonnegative and satisfies

$$S_\alpha(A) \geq I_\alpha(A;A).$$

4.2.3 The Single and Hadamard Product Entropy Gap

Another quantity of interest is related to conditional entropy of X given Y, which can be understood as the uncertainty about X that remains after knowing the joint distribution of X and Y. In Shannon's definition the Conditional entropy $H(X|Y)$ can be expressed as:

$$H(X|Y) = H(X,Y) - H(Y). \tag{4–31}$$

Extending this idea to the matrix case yields:

$$H_\alpha(A|B) = S_\alpha\left(\frac{A \circ B}{\mathrm{tr}(A \circ B)}\right) - S_\alpha(B) \tag{4–32}$$

for positive semidefinite A and B with nonnegative entries and unit trace such that $A_{ii} = \frac{1}{n}$ for all $i = 1, \dots, n$. The above quantity is nonnegative and upper bounded by

$S_\alpha(A)$. Notice that the normalization procedure for infinitely divisible matrices proposed in Theorem 4.3.3 is now beautifully justified as the maximum entropy matrix among all matrices for which the Hilbert space embeddings are isometrically isomorphic. In the following section, we will see how infinite divisible matrices relate the Hadamard products with concatenation of the representations of the variables we want to analyze jointly.

4.3 Infinitely Divisible Functions

4.3.1 Direct-Sum and Product kernels

4.3.1.1 Direct-sum kernels

Let κ_1 and κ_2 be two positive definite kernels defined on $\mathcal{X} \times \mathcal{X}$. The kernel $\kappa_\oplus = \kappa_1 + \kappa_2$, defined as $\kappa_\oplus(x, y) = \kappa_1(x, y) + \kappa_2(x, y)$, is a positive definite kernel. The above function is called direct sum kernel and it is the reproducing kernel of a space $\mathcal{H}_\oplus$ of functions of the form $f = f_1 + f_2$, where $f_1 \in \mathcal{H}_1$ and $f_2 \in \mathcal{H}_2$, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are the RKHSs defined by κ_1 and κ_2, respectively. Consider the Hilbert space $\mathcal{H} = \mathcal{H}_1 \times \mathcal{H}_2$ formed by all pairs (f_1, f_2) coming from $\mathcal{H}_1$ and $\mathcal{H}_2$,respectively. It is possible that some functions $f \neq 0$ belong to the $\mathcal{H}_1$ and $\mathcal{H}_2$ at the same time. These functions form a set of pairs $(f, -f) \in \mathcal{H}$, which turn out to be a closed subspace of $\mathcal{H}$ denoted by $\mathcal{H}_0$, such that, $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_0^\perp$. Therefore, the linear correspondence $f(x) = f_1(x) + f_2(x)$ between $f \in \mathcal{H}_\oplus$ and $(f_1, f_2) \in \mathcal{H}$ is such that all elements in $\mathcal{H}_0$ map to the zero function in $\mathcal{H}_\oplus$ and the elements of $\mathcal{H}_\oplus$ and $\mathcal{H}_0^\perp$ are in one to one correspondence. The norm of $f \in \mathcal{H}_\oplus$ can be defined from the correspondence $f \mapsto (g_1(f), g_2(f))$ as:

$$\|f\|^2_{\mathcal{H}_\oplus} = \|(g_1(f), g_2(f))\|^2_{\mathcal{H}} = \|g_1(f)\|^2_{\mathcal{H}_1} + \|g_2(f)\|^2_{\mathcal{H}_2} \tag{4–33}$$

Notice that, $(g_1(f), g_2(f))$ is the decomposition of f into the pair $\mathcal{H}$ with minimum norm in this space. The following theorem states the result [3].

Theorem 4.3.1. *If $\kappa_i(x, y)$ is the reproducing kernel of the class $\mathcal{H}_i$, with norm $\|\cdot\|_i$, then $\kappa(x, y) = \kappa_1(x, y) + \kappa_2(x, y)$ is the reproducing kernel of the class of functions $\mathcal{H}_\oplus$ of all*

functions $f = f_1 + f_2$ with $f_i \in \mathcal{H}_i$, and with the norm defined by

$$\|f\|_{\oplus}^2 = \min\left\{\|f_1\|_1^2 + \|f\|_2^2\right\} \tag{4–34}$$

the minimum is taken over all decompositions $f = f_1 + f_2$ with $f_i \in \mathcal{H}_i$

4.3.1.2 Product kernel and tensor product spaces

Consider two positive definite kernels κ_1 and κ_2 defined on $\mathcal{X} \times \mathcal{X}$ and $\mathcal{Y} \times \mathcal{Y}$, respectively. Their tensor product $\kappa_1 \otimes \kappa_2 : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y})$ defined by:

$$\kappa_1 \otimes \kappa_2((x_i, y_i), (x_j, y_j)) = \kappa_1(x_i, x_j)\kappa_2(y_i, y_j) \tag{4–35}$$

is also a positive definite kernel. Note that we can consider two kernels $\tilde{\kappa}_1$ and $\tilde{\kappa}_2$, both defined on $(\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y})$, such that $\tilde{\kappa}_1((x_i, y_i), (x_j, y_j)) = \kappa_1(x_i, x_j)$ and $\tilde{\kappa}_2((x_i, y_i), (x_j, y_j)) = \kappa_2(y_i, y_j)$; the kernel

$$\tilde{\kappa}_1 \cdot \tilde{\kappa}_2((x_i, y_i), (x_j, y_j)) = \tilde{\kappa}_1((x_i, y_i), (x_j, y_j))\tilde{\kappa}_2((x_i, y_i), (x_j, y_j)) = \kappa_1 \otimes \kappa_2((x_i, y_i), (x_j, y_j))$$

is positive definite by Schur Theorem [34]. Let us look at the space of functions that $\kappa_\otimes = \kappa_1 \otimes \kappa_2$ spans. Let $\mathcal{H}_\otimes = \mathcal{H}_1 \otimes \mathcal{H}_2$, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are the RKHSs spanned by κ_1 and κ_2, respectively. The space $\mathcal{H}_\otimes$ is the completion of the space of all functions f on $\mathcal{X} \times \mathcal{Y}$ of the form:

$$f(x, y) = \sum_{i=1}^{n} f_1^{(i)}(x) f_2^{(i)}(y) \tag{4–36}$$

with $f_1^{(i)} \in \mathcal{H}_1$ and $f_2^{(i)} \in \mathcal{H}_2$, and inner product,

$$\langle f, g\rangle_\otimes = \sum_{i=1}^{n}\sum_{j=1}^{m}\langle f_1^{(i)}, g_1^{(j)}\rangle_1 \langle f_2^{(i)}, g_2^{(j)}\rangle_2. \tag{4–37}$$

The functions f and g may have multiple representations of the form (4–36) without changing $\langle f, g\rangle_\otimes$. Let us look at the case where $\mathcal{X}$ and $\mathcal{Y}$ are the same set. The following theorem describes the kernel derived from the restriction of $\kappa_1 \otimes \kappa_2$ to the diagonal subset of $\mathcal{X} \times \mathcal{X}$ [3].

Theorem 4.3.2. *For $x, y \in \mathcal{X}$, the kernel $\kappa(x,y) = \kappa_1(x,y)\kappa_2(x,y)$ is the reproducing kernel of the class $\mathcal{H}$ of the restrictions of the direct product $\mathcal{H}_\otimes = \mathcal{H}_1 \otimes \mathcal{H}_2$ to the diagonal set formed by all elements $(x,x) \in \mathcal{X} \times \mathcal{X}$. For any such restriction f, $\|f\| = \min \|g\|_\otimes$ for all $g \in \mathcal{H}_\otimes$ such that $f(x) = g(x,x)$.*

4.3.2 Negative Definite Functions and Infinite Divisible Matrices

4.3.2.1 Negative definite functions and Hilbertian metrics

Let $\mathcal{M} = (\mathcal{X}, d)$ be a separable metric space, a necessary and sufficient condition for $\mathcal{M}$ to be embeddable in a Hilbert space $\mathcal{H}$ is that for any set $\{x_i\} \subset \mathcal{X}$ of $n+1$ points, the following inequality holds:

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j \left(d^2(x_0, x_i) + d^2(x_0, x_j) - d^2(x_i, x_j) \right) \geq 0, \tag{4–38}$$

for any $\boldsymbol{\alpha} \in \mathbb{R}^n$. This condition is equivalent to

$$\sum_{i,j=0}^{n} \alpha_i \alpha_j d^2(x_i, x_j) \leq 0, \tag{4–39}$$

for any $\boldsymbol{\alpha} \in \mathbb{R}^{n+1}$, such that $\sum_{i=0}^{n} \alpha_i = 0$. This condition is known as negative definiteness. Interestingly, the above condition implies that $\exp(-rd^2(x_i, x_j))$ is positive definite in $\mathcal{X}$ for all $r > 0$ [75]. Indeed, matrices derived from functions satisfying the above property conform a special class of matrices know as infinite divisible.

4.3.2.2 Infinite divisible matrices

According to the Schur product theorem $A \succcurlyeq 0$ implies that $A^{\circ n} = A \circ A \circ \cdots \circ A \succcurlyeq 0$ for any positive integer n. An interesting question is when the above holds if one were to take fractional powers of A, that is, when $A^{\circ \frac{1}{m}} \succcurlyeq 0$ for any positive integer m. This lead to the concept of infinite divisible matrices [12, 33].

Definition 4.3.1. *Suppose that $A \succcurlyeq 0$ and $a_{ij} \geq 0$ for all i and j. A is said to be infinite divisible if $A^{\circ r} \succcurlyeq 0$ for every nonnegative r.*

Infinite divisible matrices are intimately related to negative definiteness as we can see from the following proposition

Proposition 4.3.1. *If A is infinite divisible, then the matrix $B_{ij} = -\log A_{ij}$ is negative definite*

From this fact it is possible to relate infinitely divisible matrices with isometric embedding into Hilbert spaces. If we construct the matrix,

$$D_{ij} = B_{ij} - \frac{1}{2}(B_{ii} + B_{jj}), \tag{4–40}$$

using the matrix **B** from proposition 4.3.1. There exist a Hilbert space $\mathcal{H}$ and a mapping ϕ such that

$$D_{ij} = \|\phi(i) - \phi(j)\|^2_{\mathcal{H}}. \tag{4–41}$$

Moreover, notice that if A is positive definite $-A$ is negative definite and $\exp A_{ij}$ is infinitely divisible. In a similar way, we can construct a matrix,

$$D_{ij} = -A_{ij} + \frac{1}{2}(A_{ii} + A_{jj}), \tag{4–42}$$

with the same property (4–41). This relation between (4–40) and (4–42) suggests a normalization of infinitely divisible matrices with non-zero diagonal elements that can be formalized in the following theorem.

Theorem 4.3.3. *Let $\mathcal{X}$ be a nonempty set and d_1 and d_2 two metrics on it, such that for any set $\{x_i\}_{i=1}^n$,*

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j d_\ell^2(x_i, x_j) \leq 0, \tag{4–43}$$

for any $\boldsymbol{\alpha} \in \mathbb{R}^n$, and $\sum_{i=1}^n \alpha_i = 0$, is true for $\ell = 1, 2$. Consider the matrices $A_{ij}^{(\ell)} = \exp -d_\ell^2(x_i, x_j)$ and their normalizations $\hat{A}^{(\ell)}$, defined by:

$$\hat{A}_{ij}^{(\ell)} = \frac{A_{ij}^{(\ell)}}{\sqrt{A_{ii}^{(\ell)}}\sqrt{A_{jj}^{(\ell)}}}. \tag{4–44}$$

Then, if $\hat{A}^{(1)} = \hat{A}^{(2)}$ for any finite set $\{x_i\}_{i=1}^n \subseteq \mathcal{X}$, there exist isometrically isomorphic Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, that contain are Hilbert space embeddings of the metric spaces $(\mathcal{X}, d_\ell)$, $\ell = 1, 2$. Moreover, $\hat{A}^{(\ell)}$ are infinitely divisible.

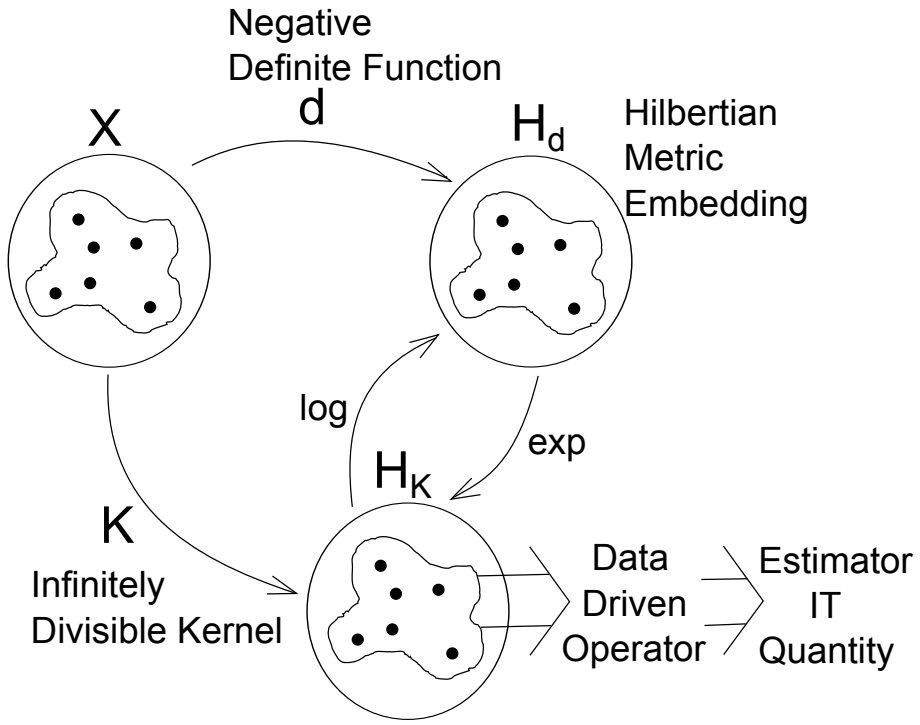


Figure 4-1. Spaces involved in the computation of the data-driven information theoretic quantities

Figure 4-1 provides an illustration of the relations between the spaces involved in the proposed matrix framework. The normalized infinite divisible kernel provides a direct representation that is suitable for the computation of the data-driven information theoretic quantities. A two-step process that is equivalent requires embedding a negative definite metric space into a Hilbert space followed by the exponential function on the squared distances.

4.4 Statistical Properties of Gram Matrices and their connection with ITL

Let $(\mathcal{X}, \mathcal{B}_{\mathcal{X}}, P_{\mathcal{X}})$ be a countably generated measure space. Let $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a reproducing kernel and the mapping $\phi : \mathcal{X} \mapsto \mathcal{H}$ such that $\kappa(x, y) = \langle \phi(x), \phi(y) \rangle$, and:

$$\begin{aligned} \mathrm{E}_X \left[\kappa(X, X) \right] &= \mathrm{E}_X \left[\| \phi(X) \|^2 \right] \\ &= \int_{\mathcal{X}} \langle \phi(x), \phi(x) \rangle \mathrm{d} P_{\mathcal{X}}(x) = 1 \end{aligned} \tag{4–45}$$

Since $\mathrm{E}_X\left[\|\phi(X)\|^2\right] < \infty$ we can define an operator $G : \mathcal{H} \mapsto \mathcal{H}$ through the following bilinear form[3] :

$$\mathcal{G}(f, g) = \langle f, Gg\rangle = \int_{\mathcal{X}} \langle f, \phi(x)\rangle\langle\phi(x), g\rangle \mathrm{d}P_{\mathcal{X}}(x) \tag{4–46}$$

notice that f and g belong to $\mathcal{H}$ and from the reproducing property of κ, we have that $f(x) = \langle f, \phi(x)\rangle$ and thus $\mathcal{G}(f, g) = \mathrm{E}_X\left[f(X)g(X)\right]$. From the normalization condition (4–45) we have that:

$$\begin{aligned} \mathrm{tr}(G) &= \sum_{i=1}^{N_{\mathcal{H}}} \mathcal{G}(\psi_i, \psi_i) \\ &= \sum_{i=1}^{N_{\mathcal{H}}} \int_{\mathcal{X}} \langle\psi_i, \phi(x)\rangle\langle\phi(x), \psi_i\rangle \mathrm{d}P_{\mathcal{X}}(x) = 1 \end{aligned} \tag{4–47}$$

where $\{\psi_i\}_{i=1}^{N_{\mathcal{H}}}$ is a complete orthonormal basis for $\mathcal{H}$, and thus G is trace class.

4.4.1 The trace of G^α

Our definition of the entropy like quantity for positive definite matrices, we employ the functional calculus using the spectral theorem to compute $\mathrm{tr}(A^\alpha)$. In particular, we consider the Gram matrix **K** constructed by all pairwise evaluations of a normalized infinite divisible kernel κ and scale by $\frac{1}{N}$ such that $\frac{1}{N}\sum_{i=1}^{N}\kappa(x_i, x_i) = 1$. The above scaling can be though as normalizing the kernel such that for the empirical distribution P_N,

$$\begin{aligned} \mathrm{E}_{\mathrm{emp}}\left[\kappa(X, X)\right] &= \mathrm{E}_{\mathrm{emp}}\left[\|\phi(X)\|^2\right] \\ &= \int_{\mathcal{X}} \langle\phi(x), \phi(x)\rangle \mathrm{d}P_N(x) \\ &= \frac{1}{N}\sum_{i=1}^{N}\kappa(x_i, x_i) = 1 \end{aligned} \tag{4–48}$$

[3] Notice, that $f \in \mathcal{H} \Rightarrow f \in L_2(P\mathcal{X})$. First, $|f(x)| = |\langle f, \phi(x)\rangle| \leq \|f\|\kappa(x, x)^{\frac{1}{2}}$, and thus $f(x)^2 \leq \|f\|^2\kappa(x, x)$. Since $\int \kappa(x, x)\mathrm{d}P_{\mathcal{X}} = 1$, we have $\|f\|_2^2 = \int f^2 \mathrm{d}P_{\mathcal{X}} \leq \|f\|^2$

It follows immediately from Proposition 4.4.1 that $\mathrm{tr}\left(\widehat{G}_N^{\alpha}\right) = \mathrm{tr}\left(\left(\frac{1}{N}\mathbf{K}\right)^{\alpha}\right)$. As we have seen, G defines a bilinear form $\mathcal{G}$ that coincides with the correlation of functions on $\mathcal{X}$ that belong to the RKHS induced by κ. Let us look at the case $\alpha = 2$, which is the initial motivation of this study and has been extensively treated in ITL in relation to plug in estimators of Renyi's entropy. This case is also important since there are interesting links with maximum discrepancy and Hilbert Schmidt norms. In the limit case we have:

$$\begin{aligned}
\mathrm{tr}\left(G^2\right) &= \sum_{i=1}^{N_{\mathcal{H}}} \langle \psi_i, G^2\psi_i\rangle = \sum_{i=1}^{N_{\mathcal{H}}} \langle G\psi_i, G\psi_i\rangle \\
&= \sum_{i=1}^{N_{\mathcal{H}}} \|G\psi_i\|^2 = \|G\|_{\mathrm{HS}}^2 \\
&= \sum_{i=1}^{N_{\mathcal{H}}} \int_{\mathcal{X}}\int_{\mathcal{X}} \langle \phi(x)\langle\phi(x), \psi_i\rangle, \ldots \\
&\qquad \phi(y)\langle\phi(y), \psi_i\rangle\rangle \mathrm{d}P_{\mathcal{X}}(x)\mathrm{d}P_{\mathcal{X}}(y) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \langle\phi(x), \phi(y)\rangle\langle\phi(x), \ldots \\
&\qquad \sum_{i=1}^{N_{\mathcal{H}}} \psi_i\langle\psi_i, \phi(y)\rangle\rangle \mathrm{d}P_{\mathcal{X}}(x)\mathrm{d}P_{\mathcal{X}}(y) \\
&= \int_{\mathcal{X}}\int_{\mathcal{X}} \langle\phi(x), \phi(y)\rangle^2 \mathrm{d}P_{\mathcal{X}}(x)\mathrm{d}P_{\mathcal{X}}(y) \\
&= \|\mu_X\|_{\mathcal{K}}^2 \qquad (4\text{–}49)
\end{aligned}$$

where $\|\mu_X\|_{\kappa^2}^2$ denotes the squared norm of a the a mapping $P_{\mathcal{X}} \mapsto \mu_X$ in the RKHS $\mathcal{K}$ induced by the kernel $\kappa^2(x, y) = \kappa(x, y)\kappa(x, y)$. In the more general case of any $\alpha > 1$

we have,

$$\begin{aligned}
\mathrm{tr}\,(G^{\alpha}) &= \sum_{i=1}^{N_{\mathcal{H}}} \langle \psi_i,\, G^{\alpha}\psi_i \rangle = \sum_{i=1}^{N_{\mathcal{H}}} \langle G\psi_i,\, G^{\alpha-1}\psi_i \rangle \\
&= \sum_{i=1}^{N_{\mathcal{H}}} \int_{\mathcal{X}} \langle \psi_i,\, \phi(x) \rangle \langle \phi(x),\, G^{\alpha-1}\psi_i \rangle \mathrm{d}P_{\mathcal{X}}(x) \\
&= \int_{\mathcal{X}} \langle \phi(x),\, G^{\alpha-1}\phi(x) \rangle \mathrm{d}P_{\mathcal{X}}(x) \\
&= \int_{\mathcal{X}} h(x,x) \mathrm{d}P_{\mathcal{X}}(x)
\end{aligned} \tag{4–50}$$

notice that $h(x,y)$ itself, is a positive definite function on $\mathcal{X} \times \mathcal{X}$ that also depends on $P_{\mathcal{X}}(x)$.

4.4.2 The Spectrum of G and Consistency of its Estimator

By definition, it is obvious that the bilinear form $\mathcal{G}$ is a positive definite kernel in $\mathcal{H}$ since

$$\sum_{i,j=1}^{N} \alpha_i \alpha_j \mathcal{G}(f_i, f_j) \geq 0 \tag{4–51}$$

for any finite set $\{f_i\}_{i=1}^{N} \subseteq \mathcal{H}$. Notice from (4–46) $\mathcal{G}$ is symmetric and thus G is self adjoint. Moreover, since $\mathcal{G}$ is positive definite, it can be shown that G is a positive definite operator. Instead of dealing directly with the spectrum of G, for which we should know the probability measure $P_{\mathcal{X}}$, we are going to look at the spectrum of $\widehat{G}_N$ and the convergence properties of this operator. Based on the empirical distribution $P_N = \frac{1}{N}\delta_{x_i}(x)$, the empirical version $\widehat{G}_N$ of G obtained from a sample $\{x_i\}$ of size N is given by:

$$\begin{aligned}
\langle f,\, \widehat{G}_N g \rangle = \widehat{\mathcal{G}}(f,g) &= \int_{\mathcal{X}} \langle f,\, \phi(x) \rangle \langle \phi(x),\, g \rangle \mathrm{d}P_N(x) \\
&= \frac{1}{N} \sum_{i=1}^{N} \langle f,\, \phi(x_i) \rangle \langle \phi(x_i),\, g \rangle
\end{aligned} \tag{4–52}$$

Proposition 4.4.1. (Spectrum of $\widehat{G}_N$): *For a sample $\{x_i\}_{i=1}^N$, let $\widehat{G}_N$ be defined as in (4–52), and let* **K** *be the Gram matrix of products $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$. Then, $\widehat{G}_N$ has at most N positive eigenvalues λ_k satisfying:*

$$\frac{1}{N}\mathbf{K}\boldsymbol{\alpha}_i = \lambda_i \boldsymbol{\alpha}_i. \tag{4–53}$$

Moreover, $N\lambda_i$ are all the positive eigenvalues of **K**.

Proof 4.4.1. *First notice that for all $f \perp \text{span}\,\{\phi(x_i)\}$, we have $\widehat{G}_N f = 0$, and thus any eigenvector with a corresponding positive eigenvalue must belong to the $\text{span}\,\{\phi(x_i)\}$, which is an N dimensional subspace and therefore, since $\widehat{G}_N$ is normal there can be at most N positive eigenvalues. Now let v be an eigenvector of $\widehat{G}_N$,, we have that*

$$\langle \cdot, \widehat{G}_N v \rangle = \frac{1}{N}\sum_{j=1}^{N} \langle \cdot, \phi(x_j) \rangle \langle \phi(x_j), v \rangle = \langle \cdot, \lambda v \rangle.$$

Then, for each $\phi(x_i)$ it is true that

$$\langle \phi(x_i), \widehat{G}_N v \rangle = \frac{1}{N}\sum_{j=1}^{N} \langle \phi(x_i), \phi(x_j) \rangle \langle \phi(x_j), v \rangle = \lambda \langle \phi(x_i), v \rangle.$$

By taking $\alpha_i = \langle \phi(x_i), v \rangle$ we can form the following system of equations:

$$\frac{1}{N}\mathbf{K}\boldsymbol{\alpha} = \lambda \boldsymbol{\alpha}. \tag{4–54}$$

which is true for all positive eigenvalues of $\widehat{G}_N$.

□

Proposition 4.4.2. (Compactness of G): *$G : \mathcal{H} \mapsto \mathcal{H}$ defined by (4–46) is compact.*

Proof 4.4.2. *We will show that if $g_n \overset{\text{w}}{\to} g$ in $\mathcal{H}$ ($\{g_n\}$ is weakly convergent), implies that $Gg_n \to g$ strongly in $\mathcal{H}$. Since $\mathcal{H}$ is a Hilbert space we only need to show that*

$$\|Gg_n\| \mapsto \|Gg\|. \tag{4–55}$$

Since any $f \in \mathcal{H}$ is also in $L_2(P_{\mathcal{X}})$,

$$\begin{aligned} \langle Gg_n, Gg_n \rangle &= \int_{\mathcal{X}} g_n(x)(Gg_n)(x)\mathrm{d}P_{\mathcal{X}}(x), \\ \langle Gg, Gg_n \rangle &= \int_{\mathcal{X}} g(x)(Gg)(x)\mathrm{d}P_{\mathcal{X}}(x). \end{aligned} \tag{4–56}$$

Moreover,

$$\begin{aligned} |g_n(x)| &\leq \|g_n\|\kappa(x,x)^{\frac{1}{2}}, \\ |Gg_n(x)| &\leq \|Gg_n\|\kappa(x,x)^{\frac{1}{2}}. \end{aligned} \tag{4–57}$$

and therefore, $|g_n(Gg_n)(x)| \leq \|g_n\|\|Gg_n\|\kappa(x,x)$. Since both $\{g_n\}$ and $\{Gg_n\}$ are weakly convergent in $\mathcal{H}$ their norms are bounded; then $\{g_n(Gg_n)\}$ is bounded by the $L_1(P_{\mathcal{X}})$ norm of $\kappa(x,x)$ (up to a constant). The weak convergence property of $\{g_n\}$ implies that $g_n(x) \to g(x)$ point-wise, which also implies $g_n(Gg_n)(x) \to g(Gg)(x)$ point-wise. Since these functions are uniformly bounded by the integrable function $\kappa(x,x)$, by Lebesgue dominated convergence in $L_1(P_{\mathcal{X}})$ we have:

$$\int_{\mathcal{X}} g_n(Gg_n)(x)\mathrm{d}P_{\mathcal{X}}(x) \to \int_{\mathcal{X}} g(Gg)(x)\mathrm{d}P_{\mathcal{X}}(x), \tag{4–58}$$

which proves that $\|Gg_n\| \to \|G_g\|$, and thus G is compact.

□

The following theorem found in [39] is a variational characterization of the discrete spectrum (eigenvalues) of a compact operator in a separable Hilbert space.

Theorem 4.4.1. *Let A, B be self adjoint operators in a separable Hilbert space $\mathcal{H}$, such that $B = A + C$, where C is a compact selfadjoint operator. Let $\{\gamma_k\}$ be an enumeration of nonzero eigenvalues of C. Then there exists extended enumerations $\{\alpha_j\}$, $\{\beta_j\}$ of discrete eigenvalues for A, B, respectively, such that:*

$$\sum_j \varphi(\beta_j - \alpha_j) \leq \sum_k \varphi(\gamma_k), \tag{4–59}$$

where φ is any nonnegative convex function on $\mathbb{R}$, and $\varphi(0) = 0$.

The definition of extended enumeration $\{\alpha_i\}$ according to Theorem 4.4.1 means that a for a selfadjoint operator A in $\mathcal{H}$ only the discrete eigenvalues with finite multiplicity m are listed m times and any other values are listed as zero. If we have a bounded kernel, which in the case of a normalized version of the infinitely divisible matrix is always the case, we can apply Hoeffding's inequality. Let Φ_i be a sequence of zero mean, independent random variables taking values in a separable Hilbert space such that $\|\Phi_i\| < C$ for all i then:

$$\Pr\left[\left\|\frac{1}{N}\sum_{i=1}^{n}\Phi_n\right\| \geq \varepsilon\right] \leq 2\exp -\frac{N\varepsilon^2}{2C^2} \tag{4–60}$$

note that $(\widehat{G}_N - G)$ is compact operator. Let ψ_j be a complete orthonormal basis for $\mathcal{H}$, we can set that,

$$\sum_{j=1}^{N_{\mathcal{H}}}(\widehat{G}_N - G)\psi_j = \frac{1}{N}\sum_{i=1}^{N}\left(\phi(X_i)\|\phi(X_i)\| - \mathrm{E}[\phi(X)\|\phi(X)\|]\right). \tag{4–61}$$

Combining (4–60) with (4–61) and Theorem 4.4.1, yields the following result.

Theorem 4.4.2. *For a positive definite kernel κ satisfying (4–45), and $\kappa(x,x) \leq C$. Let λ_i and $\widehat{\lambda}_i$ the extended enumerations of the discrete eigenvalues of G and $\widehat{G}_N$, respectively. Then, with probability $1-\delta$*

$$\left(\sum_i(\lambda_i - \widehat{\lambda}_i)^2\right)^{\frac{1}{2}} \leq \sqrt{\frac{2C\log\frac{2}{\delta}}{N}} \tag{4–62}$$

Proof 4.4.3. *Apply the result of Theorem 4.4.1 using $\varphi(x) = x^2$.*

□

4.5 Experiments: Independence Test

Here, we develop a test for independence between random elements X and Y based on the gap between the entropy of the tensor and Hadamard products of their Gram matrices. Here, we report results for an experimental setup similar to [30]. We

Table 4-1. List of distributions used in the independence test along with their corresponding original and resulting kurtosis after centralization and rescaling

Distribution	Kurtosis
Student's t distribution 3 DOF	∞
Double exponential	3.00
Uniform	-1.20
Student's t distribution 5 DOF	6.00
Exponential	6.00
Mixture, 2 double exponentials	-1.16
Symmetric mixture, 2 Gaussian, multimodal	-1.68
Symmetric mixture, 2 Gaussian, transitional	-0.74
Symmetric mixture, 2 Gaussian, unimodal	-0.50
Asymmetric mixture, 2 Gaussian, multimodal	-0.53
Asymmetric mixture, 2 Gaussian, transitional	-0.67
Asymmetric mixture, 2 Gaussian, unimodal	-0.47
Symmetric mixture, 4 Gaussian, multimodal	-0.82
Symmetric mixture, 4 Gaussian, transitional	-0.62
Symmetric mixture, 4 Gaussian, unimodal	-0.80
Asymmetric mixture, 4 Gaussian, multimodal	-0.77
Asymmetric mixture, 4 Gaussian, transitional	-0.29
Asymmetric mixture, 4 Gaussian, unimodal	-0.67

draw N *i.i.d.* samples from two randomly picked densities corresponding to the ICA benchmark densities [4]. These densities are scaled and centralized such that they have zero mean and unit variance (see Table 4-1). These random variables are mixed using a 2-dimensional rotation matrix with rotation angle $\theta \in [0, \pi/4]$. Gaussian noise with unit variance and zero mean is a added as extra dimensions. Finally each one of the random vectors is rotated by a random rotation (orthonormal matrix) in $\mathbb{R}^2$, and $\mathbb{R}^3$, accordingly. This causes the resulting random vectors to be dependent across all observed dimensions. We perform experiments varying angles, samples sizes and, dimensionality. The test compares the value of the gap:

$$S_\alpha(\mathbf{K}_X) + S_\alpha(\mathbf{K}_Y) - S_\alpha\left(\frac{\mathbf{K}_X \circ \mathbf{K}_Y}{\mathrm{tr}(\mathbf{K}_X \circ \mathbf{K}_Y)}\right), \tag{4–63}$$

where $\mathbf{K}_X$ and $\mathbf{K}_Y$ are the Gram matrices (Gaussian kernel) for the X and Y components of the sample $\{(x_i, y_i)\}_{i=1}^N$, with a threshold computed by sampling a surrogate of the null

hypothesis H_0 based on shuffling one of the components of the sample k times, that is, the correspondences between x_i and y_i are broken by the random permutations. The threshold is the the estimated quantile $1-\tau$ where τ is the significance level of the test (Type I error), meaning that the test is data dependent. The hypothesis H_0, X is independent of Y, is accepted if the gap (4–63) is below the threshold, otherwise, we reject H_0. In all our experiments $k = 100$. he solid lines in Figures 4-2A, 4-2B, and 4-2C show the estimated probability of H_0 being accepted for the proposed test with $\tau = 0.05$. The results are averages over 100 simulations for each one of the parameter configurations. We compare our results with the ones obtained by using the kernel based statistic proposed in [30],

$$\begin{aligned} T_n =& \frac{1}{n^2}\sum_{i,j=1}^{n} L_h(x_i - x_j)L'_h(y_i, y_j) + \\ & - \frac{2}{n^3}\sum_{j=1}^{n}\left[\left(\sum_{i=1}^{n} L_h(x_i - x_j)\right)\left(\sum_{i=1}^{n} L'_h(y_i - y_j)\right)\right] + \\ & + \left(\frac{1}{n^2}\sum_{i,j=1}^{n} L_h(x_i - x_j)\right)\left(\frac{1}{n^2}\sum_{i,j=1}^{n} L'_h(y_i - y_j)\right), \end{aligned} \tag{4–64}$$

where L_h and L'_h are characteristic kernels on $\mathbb{R}^d$ [84]. In the case of $X, Y \in \mathbb{R}$ (Figure 4-2A), the type II error is low even for small sample sizes, whereas the dependence becomes more difficult to detect as d increases, requiring a larger N to obtain an acceptable type II error. Our results are competitive to those obtained with the kernel based statistic (4–64). The two methods perform relatively similar for large angle, but it can be noticed that the proposed method work better when the angle is close to 0. It is important to point out that in both cases, the proposed statistic using the gap and the one in (4–64), the threshold was empirically determined by approximating the null distribution using permutations on one of the variables. Whether we can provide a distribution of the null hypothesis for (4–63) is subject of future work. Figure 4-3 shows the influence of the parameters in the power of the proposed independence test. The behavior of the test for different orders α and kernel sizes σ can be explained from the

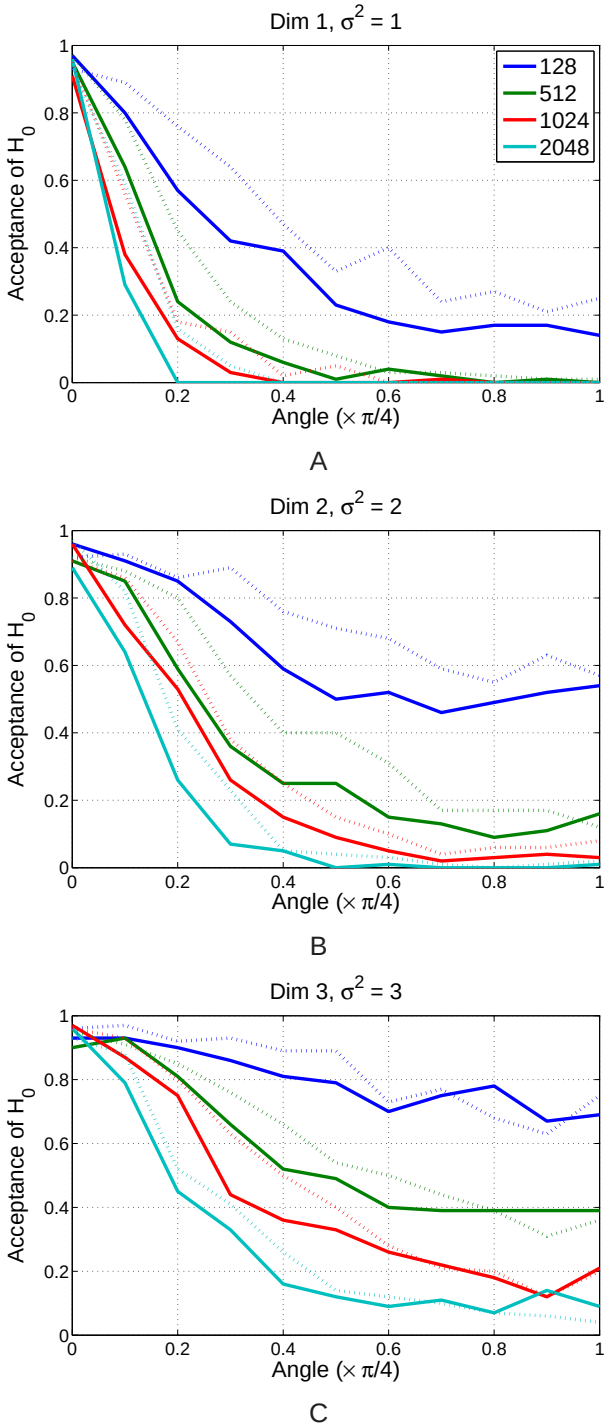


Figure 4-2. Results of the independence test based on the gap between tensor and Hadamard product entropies for different sample sizes and dimensionality. Figures 4-2A, 4-2B, and 4-2C, correspond to random variables of 1, 2, and 3 dimensions. The larger the angle the easier to reject H_0 (Independence).

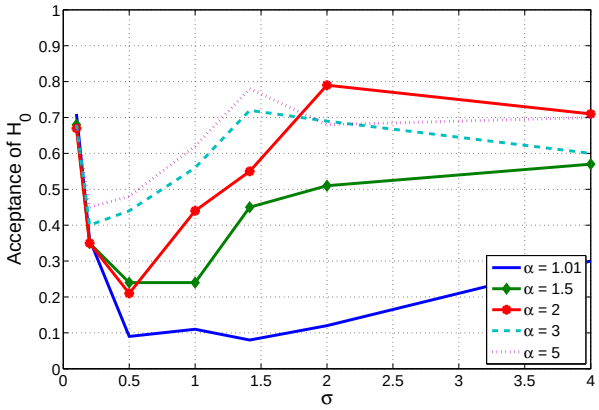


Figure 4-3. Results of the independence test based on the gap between tensor and Hadamard product entropies for different kernel sizes σ and entropy orders α for a fixed sample of size 1024 and rotation angle $\theta = \frac{\pi}{8}$. The dimensionality of the of the random variables is $d = 2$.

spectral properties of the Gram matrices. For smaller kernel sizes the Gram matrix approaches to identity and thus its eigenvalues become more similar, with $1/n$ as the limit case. Therefore, the gap (4–63) monotonically increases as $\sigma \to 0$, so does the gap for the permuted sample. Since both quantities have the same upper bound, the probability of accepting H_0 increases. The other phenomenon is related to the entropy order, it can be noticed that the larger the order α the smaller the kernel size σ that is needed to minimize the type II error. The order has an smoothing effect in the resulting operator defined in (4–50). Large α will emphasize on the largest eigenvalues of the Gram matrices that are commonly associated with slowly changing features.

CHAPTER 5
INFORMATION THEORETIC LEARNING WITH MATRIX-BASED ENTROPY

So far, we have seen how the proposed matrix based framework can be useful when comparing quantities that describe the data in a static manner. For example, we have derived quantities that can be employed to test statistical independence between two random variables X and Y. In this chapter, we want to pursue a different goal. Instead, we want to use the quantities based on matrix-entropy as objective functions to formulate learning problems as mathematical optimization problems. Thus, we are looking at maximizing or minimizing a given information theoretic quantity over a subset of the set of positive definite matrices that we will describe below. We are looking at a constrained optimization problem of the form:

$$\begin{aligned} &\underset{\mathbf{X}\in\Delta_n^+}{\text{minimize}} \;\; f_0(\mathbf{X}) \\ &\text{subject to} \quad \begin{aligned} &f_i(\mathbf{X}) \leq 0, \text{ for } i = 1, \dots, m; \\ &h_j(\mathbf{X}) = 0, \text{ for } j = 1, \dots, \ell. \end{aligned} \end{aligned} \tag{5–1}$$

Δ_n^+ denotes the set of positive definite matrices. In our information theoretic context, f_0 and in some cases f_i for $i = 1, \dots, m$ will be some of the functionals that can be derived from the entropy like functional defined in (4–13). The matrix entropy functional, fall into the family of matrix functions know as spectral functions. Let H_n denote the vector space of real Hermitian matrices of size $n \times n$ endowed with inner product $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{tr}\mathbf{XY}$; and let U_n denote the set of $n \times n$ unitary matrices. A real valued function f defines on a subset of H_n is unitarily invariant if $f(\mathbf{UXU}^*) = f(\mathbf{X})$ for any $\mathbf{U} \in U_n$; these functions only depend on the eigenvalues of $\mathbf{X}$ and therefore are called spectral functions [24]. Recalling property (i) in Proposition 4.2.1, we can clearly see that the function (4–13) belong to the class of spectral functions we have just described.

5.1 Computing Derivatives of Matrix Entropy

Associated with each spectral function f there is a symmetric function F on $\mathbb{R}^n$. By symmetric we meant that $F(\mathbf{x}) = F(\mathbf{Px})$ for any $n \times n$ permutation matrix $\mathbf{P}$. Let $\lambda(\mathbf{X})$

denote the vector of ordered eigenvalues of $\mathbf{X}$; then, a spectral function $f(\mathbf{X})$ is of the form $F(\lambda(\mathbf{X}))$ for a symmetric function F. We are interested in the differentiation of the composition $(F \circ \lambda)(\cdot) = F(\lambda(\cdot))$ at $\mathbf{X}$. The following result [46] allows us to differentiate an spectral function f at $\mathbf{X}$

Theorem 5.1.1. *Let the set $\Omega \subset \mathbf{R}^n$ be open and symmetric, that is, for any $\mathbf{x} \in \Omega$ and any $n \times n$ permutation matrix $\mathbf{P}$, $\mathbf{Px} \in \Omega$. Suppose that F is symmetric, Then, the spectral function $F(\lambda(\cdot))$ is differentiable at a matrix $\mathbf{X}$ if and only if F is differentiable at the vector $\lambda(X)$. In this case, the gradient of $F \circ \lambda$ at $\mathbf{X}$ is*

$$\nabla(F \circ \lambda)(\mathbf{X}) = \mathbf{U}\text{diag}(\nabla \text{F}(\lambda(\mathbf{X})))\mathbf{U}^*, \tag{5–2}$$

for any unitary matrix satisfying $\mathbf{X} = \mathbf{U}\text{diag}(\lambda(\mathbf{X}))\mathbf{U}^*$.

Form the above theorem it is straightforward to obtain the derivative of (4–13) at A as follows:

$$\nabla S_\alpha(A) = \frac{\alpha}{(1-\alpha)\text{tr}(A^\alpha)} U\Lambda^{\alpha-1}U^*, \tag{5–3}$$

where $A = U\Lambda U^*$. It is important to note that this decomposition can be use to our advantage. Instead of computing the full set of eigenvectors and eigenvalues of A, we can approximate the gradient of S_α by using only a few leading eigenvalues. It is easy to see that this approximation will be optimal in the Frobenius norm $\|\mathbf{X}\|_{\text{Fro}} = \sqrt{\text{tr}(\mathbf{X}^*\mathbf{X})}$.

5.2 Supervised Metric Learning

Here, we apply the proposed matrix framework to the problem of supervised metric learning. This problem can be formulated as follows. Given a set of points $\{(\mathbf{x}_i, l_i)\}_{i=1}^n$, we seek for a positive semidefinite matrix $\mathbf{AA}^\top$, that parametrizes a Mahalanobis distance between samples $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ as $d(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^\top \mathbf{AA}^\top (\mathbf{x} - \mathbf{x}')$. Our goal is to find parametrization matrix $\mathbf{A}$ such that the conditional entropy of the labels l_i given the projected samples $\mathbf{y}_i = \mathbf{A}^\top \mathbf{x}_i$ with $\mathbf{y}_i \in \mathbb{R}^p$ and $p \ll d$, is minimized. This can be posed as

the following optimization problem:

$$\begin{aligned} \underset{\mathbf{A}\in\mathbb{R}^{d\times p}}{\text{minimize}} \;& S_\alpha(L|Y) \\ \text{subject to} \;& \mathbf{A}^\top \mathbf{x}_i = \mathbf{y}_i, \text{ for } i = 1, \dots, n; \\ & \text{tr}(\mathbf{A}^\top \mathbf{A}) = p, \end{aligned} \tag{5–4}$$

where the trace constraint prevents the solution from growing unbounded. We can translate this problem to our matrix-based framework in the following way. Let $\mathbf{K}$ be the matrix representing the projected samples

$$K_{ij} = \frac{1}{n} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}\mathbf{A}^\top (\mathbf{x}_i - \mathbf{x}_j)}{2\sigma^2} \right),$$

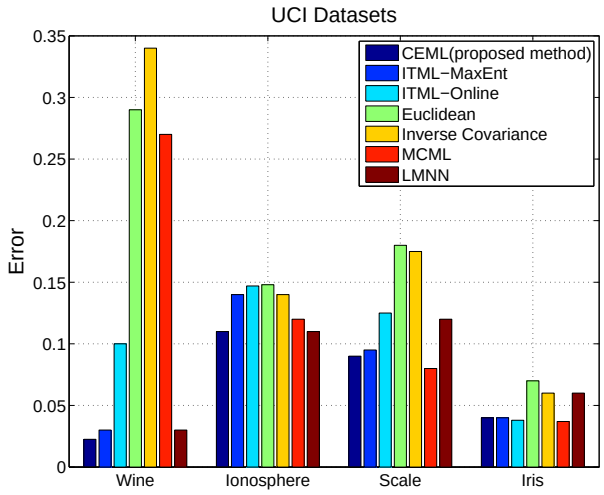
and $\mathbf{L}$ be the matrix of class co-occurrences where $L_{ij} = \frac{1}{n}$ if $l_i = l_j$ and zero otherwise. The conditional entropy can be computed as $S_\alpha(L|Y) = S_\alpha\left(n\mathbf{K} \circ \mathbf{L}\right) - S_\alpha(\mathbf{K})$, and its gradient at $\mathbf{A}$, which can be derived based on (5–2), is given by:

$$\mathbf{X}^\top (\mathbf{P} - \text{diag}(\mathbf{P}\mathbf{1})\mathbf{X}\mathbf{A}) \tag{5–5}$$

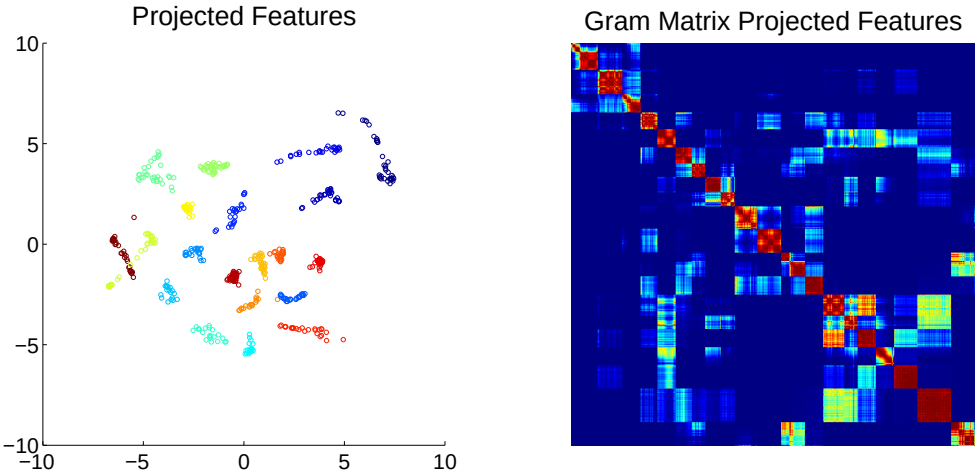
where

$$\mathbf{P} = \left(\mathbf{L} \circ \nabla S_\alpha\left(n\mathbf{K} \circ \mathbf{L}\right) - \nabla S_\alpha(\mathbf{K})\right) \circ \mathbf{K} \tag{5–6}$$

Finally, we can use (5–5) to search for $\mathbf{A}$ iteratively. To evaluate the results we use the same experimental setup proposed in [18], which compares 5 different approaches to supervised metric learning based on the classification error obtained from two-fold cross-validation using a 4-nearest neighbor classifier. The reported errors are averages errors from fives runs of each algorithm; in our case the parameters are $p = 3$, $\alpha = 1.01$ and $\sigma = \sqrt{3}$. the feature vectors were centered and scaled to have unit variance. Figure 5-1A shows the results of the proposed approach conditional entropy metric learning (CEML), two variants of information theoretic metric learning (ITML) proposed in [18], the baseline Euclidean distance, a distance based on the inverse covariance, the maximally collapsing metric learning (MCML) method from [27], and the large



A Classification error UCI data



B Projected faces UMist dataset and resulting Gram matrix $\sigma = \sqrt{2}$

Figure 5-1. Results for the Metric learning application

margin nearest neighbor (LMNN) method found in [91]. The results for the *Soybean* dataset are not reported since there is more than one possible data set in the UCI repository under that name. The errors obtained by the metric learning algorithm using the proposed matrix-based entropy framework are consistently among the best performing methods included in the comparison. We also run the algorithm on the UMist dataset; This data set consists of Grayscale faces (8 bit [0-255]) of 20 different people. The total number of images is 575 and the size of each image is 112x92 pixels for a

total of 10304 dimensions. Pixel values were normalized diving by 255 and centered. Figure 5-1B shows the projected images into $\mathbb{R}^2$. it is remarkable how a linear projection can separate the faces and it can also be seen from the Gram matrix that it tries to approximate the co-occurrence matrix **L**.

5.3 Transductive Learning with an Application to Image Super Resolution

Consider a sample of input output pairs $\{(x_i, y_i)\}_{i=1}^n$, the usual inductive learning setting tries to find a function f that captures the dependency between the input and output variables X and Y. This functions is then applied to new incoming inputs x_j, $j > n$ to predict the corresponding $y_j = f(x_j)$. On the other hand, learning by transduction as introduced in [25] can be uderstood as inference from particular to particular; in other words, a transductive learning algorithm provides a predicted value of y_j that is consistent with the previously observed pairs $\{(x_i, y_i)\}_{i=1}^n$ without explicit computation of the function f [1] .

Let us describe the information theoretic learning approach to transduction in a binary classification setting. Suppose we are given a set of $\{(x_i, y_i)\}_{i=1}^n$ of n labeled points x_i, each point x_i either belongs to class 1 or -1. There is a point x_{n+1} for which no label y_{n+1} is provided. We can construct an $(n+1) \times (n+1)$ Gram matrix by computing a given kernel on the set of inputs $\{x_i\}_{i=1}^{n+1}$ and two possible completions of the set of outputs $\mathbf{Y}_{+1} = \{y_1, \ldots, y_i, +1\}$ and $\mathbf{Y}_{-1}$ and their corresponding Gram matrices $\mathbf{L}_{+1}$ and $\mathbf{L}_{-1}$. Let $I(y)$ be defined as,

$$I_{\text{score}}(y) = \begin{cases} I(\mathbf{K}; \mathbf{L}_{-1}) & y = -1, \\ I(\mathbf{K}; \mathbf{L}_{+1}) & y = +1 \end{cases} \tag{5–7}$$

[1] Recall that a supervised learning algorithm is usually described as a mapping from the spaces of samples $\{z_i\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ to functions f on $\mathcal{X}$

The predicted label $\hat{y}_{n+1}$ for the point x_{n+1} will be chosen using the score $l_{\text{score}}(y)$ as,

$$\hat{y}_{n+1} = \arg \min_{y \in \{-1,+1\}} l_{\text{score}}(y) \tag{5–8}$$

A more general case of the above formulation problem needs to consider a set $\mathcal{Y}$ that is not necessarily discrete (*e.g.* $\mathcal{Y}$ is continuous). In this case, the prediction of y can be cast as a optimization problem.

Example-based image super resolution: Super resolution of images is a problem that has been studied from different perspectives [57]. A learning approach to image super-resolution is known as the example-based super-resolution; in this case the missing parts of information are reconstructed by incorporating prior knowledge to the process. A set of train images is partitioned into patches for which a low resolution corresponding patch is assumed to be a subsampled version of the high resolution patch. In addition to the subsampling, it may be assumed that the high resolution patch undergoes a low-pass filtering procedure before being subsampled (anti-aliasing). The method of local correlations proposed in [13] can be categorized among the example-based methods. It basic implementation employs a set of associative memories that are computed by partitioning the space of low resolution patches and training and associative memory for each one of the partition; a new image is then processed by the following steps:

1. The low resolution image is transformed into a set of patches of p size for which the mean value of the pixel intensities within a patch each removed.
2. Each patch is the compares to dictionary of centers that are the Voronoy centers of the partition done to the space of low res patches.
3. The high resolution patch is obtained as the output of the associative memory corresponding to the closest center to the low resolution image patch.

Our approach using the transductive learning method explained above can be adapted to the problem of image super resolution. Let $(\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}})$ denote the set of low res, high res pairs of training patches, and $\mathbf{X}_{\text{query}}$ denote the set of patches of a low

resolution query. The goal is to find the corresponding set high resolution patches $\mathbf{Y}_{\text{query}}$ such that the entropy gap is maximized on the full sample

$$(\mathbf{X}, \mathbf{Y}) = \left(\begin{bmatrix} \mathbf{X}_{\text{train}} \\ \mathbf{X}_{\text{query}} \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_{\text{train}} \\ \mathbf{Y}_{\text{query}} \end{bmatrix} \right). \tag{5–9}$$

That is,

$$\underset{\mathbf{Y}_{\text{query}}}{\text{maximize}}\; S_\alpha(\mathbf{K}_X) + S_\alpha(\mathbf{K}_Y) - S_\alpha\left(\frac{\mathbf{K}_X \circ \mathbf{K}_Y}{\text{tr}(\mathbf{K}_X \circ \mathbf{K}_Y)} \right), \tag{5–10}$$

where $\mathbf{K}_X$ and $\mathbf{K}_Y$ are the Gram matrices (Gaussian kernel) for the $\mathbf{X}$ and $\mathbf{Y}$ components of the full sample. This method requires an initial guess, which in our case is provided by the local correlation method. Figure 5-2 displays some of the results obtained with the proposed method with a magnification factor of $3\times$.

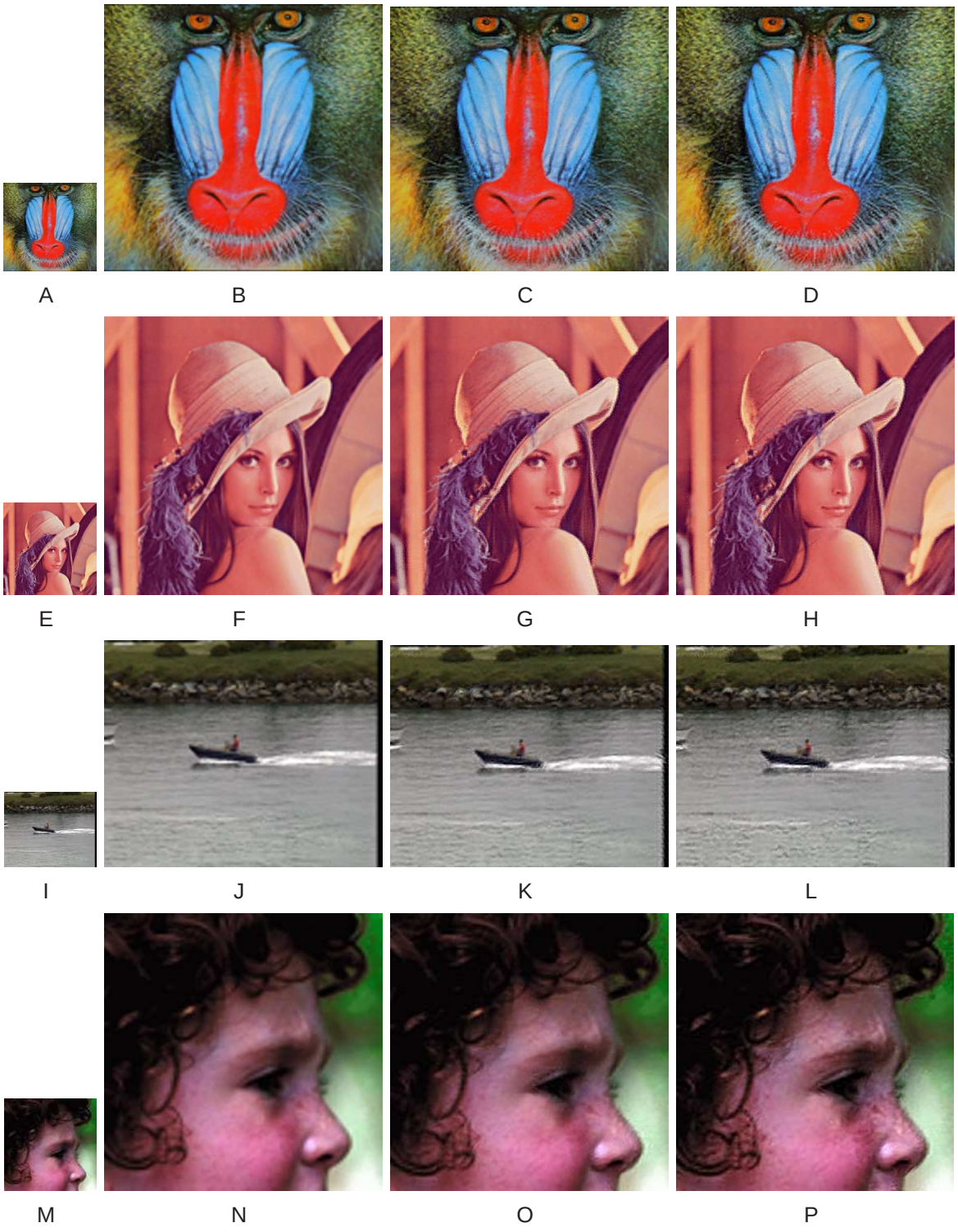


Figure 5-2. From left to right: the first column correspond to the low resolution queries, the second column is the result of applying Bi-cubic interpolation to upsample the image ($3\times$), the third column is the result of local correlation method introduced in [61], and the fourth column is the proposed approach using trasnductive learning. In the last case the initial guess corresponds to the output of the local correlation method.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

In this thesis, we extend the work on reproducing kernel Hilbert space methods for information theoretic learning that was initiated in [95]. We introduce the principle of relevant information and an alternative formulation using the information potential RKHS. This alternative formulation shows the advantages of using the reproducing kernel Hilbert space formulation for information theoretic learning. In particular, we show how can we incorporate results, methods, and algorithms that have been developed developed in the machine learning community to our framework In the spirit of kernel methods, the PRI can be solved with a support vector type algorithm. The information potential RKHS formulation effectively by-passes the density estimation step that is common on previous algorithms using information theoretic learning framework. The experiments show that the method can perform different unsupervised learning tasks by setting the trade-off parameter appropriately and results are comparable with other state of the art kernel-based feature extractors.

We introduce a sequential minimal optimization algorithm for the principle of relevant information based on weighted density estimation. In order to guarantee convergence of the algorithm, we show that the Karush-Kuhn-Tucker first order optimality condition are necessary and sufficient in our formulation. In proving this, we found there exist a convex program that yields the same solution. Results show that computational complexity is manageable even for sample sizes of several tens of thousands. The very important feature is that elements of the Gram matrix are computed at request and do not need to be stored, nevertheless, speed improvements can be achieved by using a cache that temporarily stores frequently visited samples.

The results and insights obtained with the new formulations of the principle of relevant information motivate the second part of this dissertation. Here, we present an estimation framework of entropy-like quantities based on infinite divisible matrices.

By using the axiomatic characterization of Renyi's entropy, a functional on positive definite matrices is defined. We define estimators of entropy -like quantities based on gram matrices that correspond to infinite divisible kernels. The use of Hadamard products allows us to define quantities that are similar to mutual information and conditional entropy. We showed some properties of the proposed quantities and their asymptotic behavior as operators in reproducing kernel Hilbert space defining distribution dependent kernels. Numerical experiments showed the usefulness of the proposed approach with results that are competitive with the state of the art. Moreover, We show how quantities such as conditional entropy and mutual information can be employed to formulate different learning problems, and provide efficient ways to solve the optimization that arise from these formulations.

Future Work: One important subject for further research is related to extensions of the principle of relevant information that can lead to supervised or semi-supervised learning. For instance, minimization of the joint entropy of input-output target pairs along with an information preservation constraint with respect to the given marginal of the inputs is very similar to the concept of manifold regularization proposed by [9].

There is room for improvements on the sequential minimal optimization algorithm proposed to solve the RKHS PRI formulation. Namely, improvements in terms of speed by better selection heuristics and memory trade offs can be pursued in future implementations.

Another interesting question is whether the principle can be extended to deal with random variables of various domains capturing relevant information through variables that can help in problems such as visualization.

Another interesting line of work involve the relation between information theory and statistical mechanics. Adding physical understanding of the Information theoretic formulations of learning problems can be illuminating in proposing biologically plausible implementations of information theoretic objective functions.

There is plenty of room for developments in optimization and numerical analysis that lead to efficient, large scale algorithms and approximations that can be employed to find solution to the information theoretic learning objective functions based on matrices. One of the big hurdles of the proposed framework is the estimation of the spectrum, we propose to use the largest eigenvalues to approximate the gradient of the information theoretic quantities, but these approximation still requires the storage of the full Gram matrix. It seems that greedy approximation to the Gram matrix are a necessary step to large scale implementations.

APPENDIX A
BASIC DEFINITIONS AND STANDARD NOTATION

A.1 Shannon's Entropy and Mutual Information

One of the underpinning concepts in information theory is entropy, which can be understood as the average uncertainity for some given process. The first definition of entropy was introduced by Shannon [80] in the context of communication systems, and gave rise to information theory. The entropy function was defined to satisfy thefollowing requirements: continuity in the measure of uncertainity, that is on the probability arguments; for a partition where all elements (disjoint subsets) have the same probability, the function must depend on the number of subsets; and for a refinement of a certain collection of disjoint subsets the entropy should increase[1] .
Let $(\Omega, \mathcal{B}_\Omega, P)$ be a probability space and $X : (\Omega, \mathcal{B}_\Omega) \mapsto (\mathcal{X}, \mathcal{B}_\mathcal{X})$ a measurable function. For simplicity, first, consider the case where the elements x_i of $\mathcal{X}$ are countable (or finitely many) and define a partition

$$\mathcal{U} = \{X^{-1}(x_1), X^{-1}(x_2), \dots\} \tag{A–1}$$

of Ω. For a functional H representing the uncertainity about the partition $\mathcal{U}$ (induced by the measurable function X). The entropy of discrete random variable X that can take n possible values denoted by x_n

$$H(X) = \mathrm{E}_X\left[\log \frac{1}{\mathrm{P}(X)}\right] = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log \mathrm{P}(x_i) \tag{A–2}$$

Starting from a similar formulation, it is possible to define the joint entropy

$$H(X, Y) = \mathrm{E}_{X,Y}\left[\log \frac{1}{\mathrm{P}(X, Y)}\right] = -\sum_{j=1}^{m}\sum_{i=1}^{n} \mathrm{P}(x_i, y_j) \log \mathrm{P}(x_i, y_j) \tag{A–3}$$

[1] Notice we don explicitly state that the function should be strictly increasing, but but this is the case where the resulting subsets of the partition have non-zero measure since their probaility decrease and hence the uncertainity increases

and the conditional entropy

$$H(Y|X) = \mathrm{E}_{X,Y}\left[\log \frac{1}{\mathrm{P}(Y|X)}\right] = -\sum_{j=1}^{m}\sum_{i=1}^{n} \mathrm{P}(y_i, x_i) \log \mathrm{P}(y_j|X = x_i) \tag{A–4}$$

observe that the joint entropy corresponds to the expection of the uncertaintity of a variable (Y in our case) given that we have knowledge of the other (X). and in the case of independent random variables we have that

$$H(Y) = H(Y|X)$$

And when $y = g(x)$ for a deterministic mapping g we have $H(Y|X) = 0$. We can also interpret entropy not in terms of uncertainity related to a possible outcome of an experiment, but as a measure of the information we gain from observing a given outcome. Using the above definitions and the latter interpretation of entropy, we can introduce the concept of mutual information between two random variables as the amount of information that is common to the two variables. In this case the additivity property of entropy plays an important role:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X) \tag{A–5}$$

Notice that mutual information is a symmetric function and is zero if X and Y are independent A straighforward algebraic manipulation of (A–5) yields:

$$I(X;Y) = \sum_{j=1}^{m}\sum_{i=1}^{n} \mathrm{P}(y_i, x_i) \log \frac{\mathrm{P}(y_i, x_i)}{\mathrm{P}(y_j)\,\mathrm{P}(x_i)}. \tag{A–6}$$

The argument of the log function in (A–6) sugest a more general statement. The ratio between the joint and the product of the marginals can be seen as a comparison between two valid probablity mass functions. The *Kullback-Leibler divergence* is the generalization of our rather informal statement.

Definition A.1.1. (Kullback-Leibler KL divergence): *Let P and Q be two probability mass functions defined over the same domain. The relative entropy or KL divergence is is*

defined as

$$\mathrm{KL}(P\|Q) = \sum_{i=1}^{n} P(x_i) \log \frac{P(x_i)}{Q(x_i)}. \tag{A–7}$$

Notice the above above definitions have been defined for the case of discrete probability spaces (countable sets). Nevertheless, these concepts can also be extended to continuous random variables as well as other measurable functions[2] . For a continuous random variable X and probability density function $f(x)$ with support $\mathcal{X}$ the differential entropy is given by:

$$H(X) = \mathrm{E}_X\left[\log \frac{1}{f(X)}\right] = -\int_{\mathcal{X}} f(x) \log f(x) \mathrm{d}x, \tag{A–8}$$

the joint entropy

$$H(X, Y) = \mathrm{E}_{X,Y}\left[\log \frac{1}{f(X, Y)}\right] = -\iint_{\mathcal{X},\mathcal{Y}} f(x, y) \log f(x, y) \mathrm{d}x\mathrm{d}y, \tag{A–9}$$

and the conditional entropy

$$H(Y|X) = \mathrm{E}_{X,Y}\left[\log \frac{1}{f(Y|X)}\right] = -\iint_{\mathcal{X},\mathcal{Y}} f(x, y) \log f(y|x) \mathrm{d}y\mathrm{d}x. \tag{A–10}$$

Mutual information can obtained using (A–5) and the KL divergence for two valid density functions f and g

$$\mathrm{KL}(f\|g) = \int_{\mathcal{X}} f(x) \log \frac{f(x)}{g(x)} \mathrm{d}x. \tag{A–11}$$

A.2 Rényi's Measures of Information

The above introduction is rather informal and aims at providing a condensed picture of the tools we want to use throughout this work. In this section, we want to provide a

[2] As long as we can define a partition directly or indirectly [56]. However, some properties defined for discrete RVs may differ when we move to the continuous case.

more detailed description of the particular measures from which information theoretic learning has been developed [70].

Definition A.2.1. (Generalized Random Variable): *Let* $(\Omega, \mathcal{B}_\Omega, P)$ *be a probability space, and* $X(\omega)$ *a function defined on* $\Omega_1 \in \mathcal{B}_\Omega$ *where* $P(\Omega_1) < 1$ *and is measurable with respect to* $\mathcal{B}_\Omega$*.* X *is called a* generalized random variable*.*

The distribution of a generalized random variable is called a *generalized distribution*. In the case of discrete generalized random variable with finite support, the generalized distribution can be represented by a sequence of $\mathcal{P} = (p_1,\ p_2,\ \ldots,\ p_n)$ and a weight $W(\mathcal{P}) = \sum_{i=1}^{n} p_i$, where each p_i corresponds to the probability of each element of the partition (recall (A–1)). Since we are dealing with generalized distributions of subsets Ω_τ[3] of the sample space Ω. With these elements at hand, it is possible to characterize the entropy function by the following postulates[4] :

(**H1**)**:** $H(\mathcal{P})$ is a symmetric fucntion of the elements of $\mathcal{P}$.

(**H2**)**:** $H(\{p\})$ is continuous on $(0, 1]$.

(**H3**)**:** For generalized distributions $\mathcal{P}$ and $\mathcal{Q}$, $H(\mathcal{P} \times \mathcal{Q}) = H(\mathcal{P}) + H(\mathcal{Q})$.

(**H4**)**:** For generalized distributions $\mathcal{P}$ and $\mathcal{Q}$ such that $W(\mathcal{P}) + W(\mathcal{Q}) \leq 1$

$$H(\mathcal{P} \cup \mathcal{Q}) = \frac{W(\mathcal{P})H(\mathcal{P}) + W(\mathcal{Q})H(\mathcal{P})}{W(\mathcal{P}) + W(\mathcal{Q})} \tag{A–12}$$

A function satisfying such postulates is given by

$$H(\mathcal{P}) = \frac{\sum_{i=1}^{n} p_i \log \frac{1}{p_i}}{\sum_{i=1}^{n} p_i} \tag{A–13}$$

[3] According to definition A.2.1, if $(\Omega, \mathcal{B}_\Omega, P)$ is a probability space. We can define a generalized distribution for $\Omega_\tau \subset \Omega$, if $(\Omega_\tau, \Omega_\tau \cap \mathcal{B}_\Omega)$ is a measurable space.

[4] There is an extra postulate $H(\{1/2\}) = 1$ but this is only useful to define the base $\log$ function. It is an scaling constant.

This construction agrees with the notion of of average uncertainity. In fact, it can be extended to a more general form of a mean value using the Kolmogorov-Nagumo function $\phi(x)$, that is:

$$\phi^{-1}\left(\sum_{i=1}^{n} w_k \phi(x_k)\right) \tag{A–14}$$

where ϕ is an arbitrary strictly monotonic and continuous function. Under these conditions ϕ^{-1} is the inverse function. Postulate (**H4**) is then replaced by (**H4**′)**:** For generalized distributions $\mathcal{P}$ and $\mathcal{Q}$, there exists a strictly monotonic and increasing function ϕ, such that if $W(\mathcal{P}) + W(\mathcal{Q}) \leq 1$

$$H(\mathcal{P} \cup \mathcal{Q}) = \phi^{-1}\left(\frac{W(\mathcal{P})\phi\left(H(\mathcal{P})\right) + W(\mathcal{Q})\phi\left(H(\mathcal{P})\right)}{W(\mathcal{P}) + W(\mathcal{Q})}\right) \tag{A–15}$$

It is clear that for the case $\phi(x) = ax + b$ and $a \neq 0$ the postulate reduces to the original construction. Now, if $\phi(x) = 2^{(\alpha-1)x}$, the function

$$H_\alpha(\mathcal{P}) = \frac{1}{1-\alpha}\log\left(\frac{\sum_{i=1}^{n} p_i^\alpha}{\sum_{i=1}^{n} p_i}\right) \tag{A–16}$$

is called the entropy of order α of the generalized distribution $\mathcal{P}$, with $\lim_{a\to 1} H_\alpha(\mathcal{P}) = H(\mathcal{P})$, where $H(\mathcal{P})$ is defined in equation (A–13). As we have pointed out before the entropy not only refers to uncertainity but to information, as well. In this sense, we can go back to the more general concept related to mutual information divergence. Following a similar scheme used to define the entropy of α order, we can define the α-Divergence for a discrete random variable as:

$$D_\alpha(\mathcal{P}\|\mathcal{Q}) = \frac{1}{1-\alpha}\log\left(\frac{\sum_{i=1}^{n} \frac{p_i^\alpha}{q_i^{1-\alpha}}}{\sum_{i=1}^{n} p_i}\right) \tag{A–17}$$

For a continuous random variable X with pdf $f_X(x)$, the differential entropy of α order,

$$H_\alpha(X) = \frac{1}{1-\alpha}\log\left(\int_{\mathcal{X}} f(x)^\alpha \mathrm{d}x\right) \tag{A–18}$$

and the for two density functions f and g the α-Divergence

$$D_\alpha(f\|g) = \frac{1}{1-\alpha}\log\left(\int_{\mathcal{X}} \frac{f(x)^\alpha}{g(x)^{1-\alpha}}\mathrm{d}x\right) \tag{A–19}$$

As $\alpha \to 1$ (A–18) approximates to Shannon's entropy and (A–19) to the K-L divergence. A modified version of Renyi's definition of α-relative entropy between random variables with PDFs f and g is given in [48],

$$D_\alpha(f\|g) = \log\frac{\left(\int g^{\alpha-1}f\right)^{\frac{1}{1-\alpha}}\left(\int g^\alpha\right)^{\frac{1}{\alpha}}}{\left(\int f^\alpha\right)^{\frac{1}{\alpha(1-\alpha)}}}. \tag{A–20}$$

Similarly, Shannon's relative entropy (K-L divergence) is the limit for $\alpha \to 1$. An important component in the relative entropy is the cross-entropy term $H_\alpha(f;g)$ that quantifies the information gain from observing g with respect to the "true" density f. It turns out that for the case of $\alpha = 2$, the above quantities can be expressed, under some restrictions, as functions of inner products between PDFs. In particular, the 2-order entropy of f and cross-entropy between f and g, can be respectively expressed as,

$$\begin{aligned} H_2(f) &= -\log\int_{\mathcal{X}} f^2(x)\mathrm{d}x; \\ H_2(f;g) &= -\log\int_{\mathcal{X}} f(x)g(x)\mathrm{d}x, \end{aligned} \tag{A–21}$$

the associated relative entropy of order 2 is called the Cauchy-Schwarz divergence and is defined as follows:

$$D_{\mathrm{CS}}(f\|g) = -\frac{1}{2}\log\frac{\left(\int fg\right)^2}{\left(\int f^2\right)\left(\int g^2\right)}. \tag{A–22}$$

APPENDIX B
AUXILIARY THEOREMS PROOFS AND DERIVATIONS

B.1 Sufficient Conditions for Pseudo-Convex Programs

The following theorem is extracted from [49] Chapter 10.

Theorem B.1.1. *Let $\mathcal{X}^0$ be an open set in $\mathbb{R}^n$ an let f and g be respectively and scalar and a m-dimensional vector function both defined in $\mathcal{X}^0$. Let $x^* \in \mathcal{X}^0$, $I = \{i|\ g_i(x^*) = 0\}$, f be pseudo-convex at x^*, and g be differentiable and quasi-convex at x^*. If there exists a $\boldsymbol{\mu}^* \in \mathbb{R}^m$ such that the pair $(x^*, \boldsymbol{\mu}^*)$ satisfy the following conditions:*

$$\begin{aligned} \left[\nabla f(x^*) + \boldsymbol{\mu}^{*\mathsf{T}} \mathrm{D}g(x^*)\right](x - x^*) &\geq 0,\ \forall x \in \mathcal{X}^0;\ g(x) \leq 0 \\ \boldsymbol{\mu}^{*\mathsf{T}} g(x^*) &= 0 \\ g(x^*) &\leq 0 \\ \boldsymbol{\mu}^* &\geq 0 \end{aligned} \tag{B–1}$$

Then, x^* is a solution of the following minimization problem

$$\begin{aligned} &\min_{x \in \mathcal{X}^0} \ f(x) \\ &\text{subject to } \ g(x) \leq 0. \end{aligned} \tag{B–2}$$

Proof B.1.1. *Let $I = \{i|\ g_i(x^*) = 0\}$, $J = \{j|\ g_i(x^*) < 0\}$, thence $I \cup J = \{1, \ \ldots, \ m\}$ since $\boldsymbol{\mu}^* \geq 0$, $g(x^*) \leq 0$, and $\boldsymbol{\mu}^* \geq 0$, we have that $\{\mu_j\}_{j \in J} = 0$, and from quasi convexity of g at x^*, the gradients of g_i at x^* for $i \in I$ are orthogonal to tangent planes to the level sets defined by $g_i(x) = 0$ and therefore for any feasible point $x \in \mathcal{X}^0$ and $g(x) \leq 0$, $\mathrm{D}g_I(x^*)(x - x^*) \leq 0$, by non-negativity of $\boldsymbol{\mu}$ and since $\boldsymbol{\mu}_J = 0$ we have:*

$$\begin{aligned} \boldsymbol{\mu}_I^{*\mathsf{T}} \mathrm{D}g_I(x^*)(x - x^*) &\leq 0 \\ \boldsymbol{\mu}_J^{*\mathsf{T}} \mathrm{D}g_J(x^*)(x - x^*) &= 0 \\ \boldsymbol{\mu}^{*\mathsf{T}} \mathrm{D}g(x^*)(x - x^*) &= \left[\boldsymbol{\mu}_I^{*\mathsf{T}} \mathrm{D}g_I(x^*) + \boldsymbol{\mu}_J^{*\mathsf{T}} \mathrm{D}g_J(x^*)\right](x - x^*) \leq 0. \end{aligned} \tag{B–3}$$

Finally, since $\left[\nabla f(x^*) + \boldsymbol{\mu}^{*\mathsf{T}} \mathrm{D} g(x^*)\right](x - x^*) \geq 0$ *for all* $x \in \mathcal{X}^0$ *and* $g(x) \leq 0$*, we need that* $\nabla f(x^*)(x - x^*) \geq 0$ *and thus by pseudo-convexity of* f *implying that* $f(x) \geq f(x^*)$ *for all* $x \in \mathcal{X}^0$ *such that* $g(x) \leq 0$*.*

□

A generalization of the Kuhn-Tucker sufficient optimality criterion follows from the above theorem by replacing condition (B–1) with

$$\nabla f(x^*) + \boldsymbol{\mu}^{*\mathsf{T}} \mathrm{D} g(x^*) = 0 \tag{B–4}$$

B.2 Details of the Solution to the Minimal Subproblem

We refer to the objective in (3–29) as,

$$J(\beta_2) = (\lambda - 1) \log A(\beta_2) - 2\lambda \log B(\beta_2). \tag{B–5}$$

Taking the derivative of $J(\beta_2)$ and equating to zero yields:

$$\frac{\mathrm{d}}{\mathrm{d}\beta_2} J(\beta_2) = 0 = (\lambda - 1) B(\beta_2) \frac{\mathrm{d}}{\mathrm{d}\beta_2} A(\beta_2) - 2\lambda A(\beta_2) \frac{\mathrm{d}}{\mathrm{d}\beta_2} B(\beta_2) \tag{B–6}$$

where

$$\frac{\mathrm{d}}{\mathrm{d}\beta_2} A(\beta_2) = 2(\beta_2(V_{11} + V_{22} - V_{12}) + (w(V_{12} - V_{11}) + (v_2 - v_1))) \tag{B–7}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}\beta_2} B(\beta_2) = q_2 - q_1 \tag{B–8}$$

Expanding and rearranging yields (3–30)

APPENDIX C
APPROACHES TO UNSUPERVISED LEARNING

Among the vast amount of literature that can be found on unsupervised learning, there is customary element in their presentation that divides the methods into two main categories: discrete variable methods, and continuous variable representations. Discrete representations can be easily associated with binary representations for which a feature is either present or not; clustering and vector quantization are representatives of this category. In what concerns to continuous variable representations, we can find techniques for dimensionality reduction that look for faithful low dimensional representation of the data; methods that do not necessarily reduce dimension but give representations that decompose into independent factors and methods that exploit redundancy to build representations that are robust. Finally learning the probability distribution, which may fall into ether groups. The latter problem motivates a generative view of unsupervised learning that performs inference for a given model from observed data to give a posterior distribution and a learning step that uses the posterior to update the model.

As we mentioned before, in order to learn the representation from the observed data we need two address the questions of what look for in a the representation and how to assess the effectiveness without resorting to further stages of processing.
In here, we attempt to present some of the existing techniques for unsupervised learning based on our main objective, which is understand the role of information theoretic principles behind unsupervised learning and how they apply to learning data representation. For this reason, our review divides the set of techniques according to the way information flows in each method.

Let $\mathcal{X}$ be our input set and $\mathcal{Z}$ the code set. A mapping $f : \mathcal{X} \mapsto \mathcal{Z}$ (the mapping can be random or deterministic) will be called the *encoder*, and the mapping $g : \mathcal{Z} \mapsto \mathcal{X}$ the *decoder*. The composition $g \circ f$ is called *encoder-decoder channel* or simply *channel*.

Figure C-1 depicts a block diagram of the above elements. The encoder-decoder

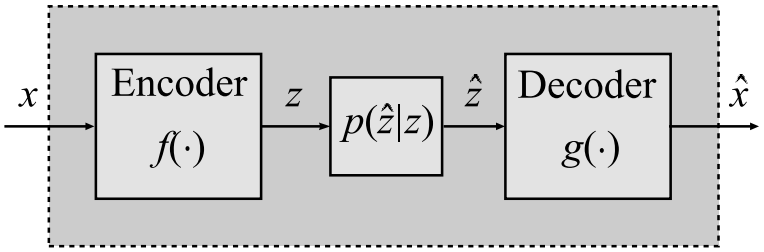


Figure C-1. Block diagram encoder decoder scheme

structure allows us to classify the existing unsupervised learning methods into three main groups: Encoder learning, decoder learning, and encoder-decoder channel learning.

C.1 Encoder Learning Methods

In this case our main concern is to learn how to code x into z. Recovering x or an approximation to it by using z might be possible but it is not required for these methods. The relation between $\mathcal{X}$ and $\mathcal{Z}$ is established through a mapping f that can be learned explicitly or implicitly.

k-Means Clustering is simple algorithm that partitions the input space in k disjoint portions according to the distribution of the data. Let $\mathcal{X}$ be $\mathbb{R}^d$, $C = \{\boldsymbol{\mu}_\ell\}_{\ell=1}^k \subset \mathbb{R}^d$ a set of centers. The objective is to adjust the locations of the set of centers C such that the objective is minimized:

$$\mathrm{E}\left[\left\| X - \arg\min_{\boldsymbol{\mu}_\ell \in C} \{\|X - \boldsymbol{\mu}_\ell\|^2\} \right\|^2\right]. \tag{C–1}$$

The sample estimator of the cost leads to an iterative procedure for which the sample $S = \{\mathbf{x}_i\}_{i=1}^N$ is partitioned into k disjoint subsets according to C, using the elements of each subset S_ℓ the values of $\boldsymbol{\mu}_\ell$ are updated according to the sample mean. The algorithm iterates until the partitions do not change. The k-means algorithm is a simple method and has been widely applied in different tasks that require quantization or a prototype based representation of a very large sample. As an information theoretic objective the k-means clustering correspond to the a minimization of a divergence

between the data distribution and a mixture of k isotropic Gaussian distribution. One caveat of the cost function is the presence of multiple local minimum, which makes the algorithm sensitive to initial conditions.

Principal component analysis (PCA) seeks for a linear mapping $\mathbf{A} : \mathbb{R}^d \mapsto \mathbb{R}^p; \mathbf{x} \mapsto \mathbf{A}\mathbf{x}$, where $p < d$, $\mathbf{A}\mathbf{A}^T = \mathbf{I}$, and $\mathrm{E}[X] = \mathbf{0}$, such that $\mathrm{tr}\,\mathrm{E}[ZZ^T]$ is maximized. In other words, we look for a linear transformation for which outputs have maximum variance [38] and become uncorrelated. The second condition can be rather thought of as a consequence of the solution, the rows of $\mathbf{A}$ correspond to the transpose of p leading eigenvectors of the covariance matrix of X. Due to its simplicity PCA is widely applied in practice, but may gifitted to representve poor results since only works with the second order statistics of the data. Assuming that X Gaussian distributed, PCA can be though as finding the linear subspace for which Z has maximum differential entropy, since any linear transformation of a Gaussian variable results in another Gaussian the maximum entropy objective becomes a variance maximization objective.

Kernel PCA [77] is nonlinear generalization of PCA based on the theory of positive definite kernels. Let $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a positive definite kernel, where $\mathcal{X}$ is an arbitrary set. For $x, y \in \mathcal{X}$, $\kappa(x, y) = \langle \phi(x), \phi(y) \rangle$ represents an inner product on a reproducing kernel Hilbert space $\mathcal{H}$, where $\phi : \mathcal{X} \mapsto \mathcal{H}$ is an underlying mapping induced by the kernel function. In this generalization, linear PCA is carried out in $\mathcal{H}$ by a dual representation of the eigenvectors of the covariance operator in $\mathcal{H}$. For the sample $S = \{x_i\}_{i=1}^N$, let $\mathbf{K}$ be the Gram matrix with entries $K_{ij} = \kappa(x_i, x_j)$ for $i, j = 1, \dots, N$, and for ease of exposition let us assume the kernel is such that the sample is centralized in the feature space. The solution is found by solving $\mathbf{K}\boldsymbol{\alpha}_\ell = \lambda_\ell \boldsymbol{\alpha}_\ell$ for the p eigenvectors with largest eigenvalues. The encoder is a nonlinear mapping from $\mathcal{X}$ to $\mathbb{R}^p$ such that for each dimension of $\mathbf{z}$ there is a function $f_\ell : \mathcal{X} \mapsto \mathbb{R}$. For $x \in \mathcal{X}$ the evaluation of f_ℓ on x is

expressed in term of kernel evaluation as follows:

$$f_\ell(x) = \frac{1}{\sqrt{\lambda_\ell}} \sum_{i=1} \alpha_{\ell i} \kappa(x_i, x), \tag{C–2}$$

where $\alpha_{\ell i}$ is the i-th entry of the ℓ-th eigenvector of $\mathbf{K}$.

Kernel entropy component analysis (KECA) [36] is a method very similar to kernel PCA in the way the eigenvectors in $\mathcal{H}$ are computed using the dual formulation. However, the solution is motivated by the relation between estimators of Renyi's second order entropy based on Parzen windows and the Gram matrix $\mathbf{K}$ [1] , and thus this matrix does not require centering as in kernel PCA. In KECA the mapping is computed using the p eigenvectors that have the highest scores $\gamma_\ell = \sqrt{\lambda_\ell} \left| \sum_{i=1}^{N} \alpha_{\ell i} \right|$. Although, both kernel PCA and KECA are nonlinear algorithms with a very elegant mathematical formulation, the dual representation employed for the eigenvectors comes at the price of increased computational burden for larger N. Moreover, the information theoretic interpretation of KECA using the Renyi's entropy relation is not shown to remain valid for every positive semidefinite kernel; thus, the choice of the eigenvalues extracted by KECA may be misleading in some cases.

Noiseless independent component analysis (ICA) [10, 35] is a paradigm motivated by the earlier work of Barlow on minimun entropy codes and redundancy reduction [6–8]. ICA computes a linear transformation of the data $\mathbf{z} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} : \mathbb{R}^d \mapsto \mathbb{R}^d$, such that the components of Z are statistically independent. Because independence implies zero linear correlation between variables, ICA is usually seen as the composition of two linear operations $\mathbf{A} = \mathbf{U}\mathbf{B}$ where $\mathbf{B}$ is a whitening operation and $\mathbf{U}$ an orthonormal matrix (*i.e.* a rotation). As a result, $\mathrm{E}[ZZ^T] = \mathbf{I}$, this explains why there is no dimensionality reduction as in PCA since there is no explicit reason to select a subset of the components. One of the early algorithms for ICA was proposed

[1] The topic of Renyi's entropy is developed in detail in the Appendix

in [10]. The idea is to maximize the mutual information between the input $X \in \mathbb{R}^d$ and the output of a nor linear mapping $\mathbf{y} = (h \circ \mathbf{A})h(\mathbf{A}x)$. This objective is fully achieved if the nonlinearity h matches the shape of the cumulative distribution of each component of $Z = \mathbf{A}X$. Since the input output map is deterministic, the maximization of the mutual information only depends on the entropy of the outputs. The maximum entropy distribution over a finite interval is the uniform distribution and thus the property of the network to maximize mutual information if the nonlinearity matches the cumulative distribution. Another assumption about independent components is that they tend to be non-Gaussian, this property is exploited in [35], where each component of Z is optimized to maximize or minimize its kurtosis, in particular, values of this statistics that are close to 3 since this is the value of kurtosis for Gaussian distributed random variables. Notice that this algorithm does not employ the nonlinearity h to find the linear map. Although ICA is well-motivated from the physiological point of view more recent approaches that incorporate robust coding seem to be more plausible and expose better features.

Self organizing maps (SOM) [22] seek for a mapping from $\mathcal{X}$ to a low-dimensional space such that proximity relations are preserved as much as possible. The mapping is computed to a topologically constrained network. The units of the network are arranged in a lattice that forces neighboring units to learn at the same time. For instance, consider a set of $k \ll N$ units, where N is the size of the data set, the j-th unit computes a map $f_i : \mathcal{X} \mapsto \mathbb{R}$ of the form $f_j(x_i) = \langle \phi(x_i), w_j \rangle$. By competitive learning, the unit parameters are updated according to the unit with the highest activation value i^*, that is, $w_j \leftarrow w_j + \eta(t)P_{i^*}(j)\phi(x_i)$, where $\eta(t)$ is the step size, which may vary over time, and P_ℓ is a strength function centered at the ℓ-th point in the lattice that weights the neighboring units to ℓ. This technique is very simple to compute, and has been subject of a significant amount of research. Several connections to vector quantization and manifold learning have been also established. Some know problems of the SOM are

local minimum and model selection. Picking the right topology is not trivial and since there is no particular cost function, it is difficult to assess the success of learning.

As mentioned above, the encoder can be explicit, *e.g.* the methods we just presented, or the mapping can be learned implicitly. In the latter case, we do not have direct access to the encoder function f, but we use the values of z to imply a transformation $z = f(x)$. Instances of the implicit encoder learning approach are **spectral clustering** [90] and **multidimensional scaling (MDS)** [22]. Spectral clustering partitions the input data set by looking at the spectrum of different normalizations of the graph Laplacian of the data (for an introductory treatment see [90]). The method has some appealing properties such as: convexity of the cost, ability to unveil nonlinear structures in data, and for some cases, theoretical guarantees such as consistency [89]. MDS uses a distance function $D : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ to construct a distance matrix $\mathbf{D}$ with entries $D_{ij} = D(x_i, x_j)$, and a distance function (usually euclidean) $d : \mathcal{Z} \times \mathcal{Z} \mapsto \mathbb{R}$ to compute a distance matrix $\mathbf{\Delta}$. Then, the locations of a set $S_z = \{\mathbf{z}_i\}_{i=1}^N$, usually in $\mathbb{R}^p$, are optimized to minimize the cost

$$J(S_z) = \sum_{i<j}^{N} \frac{(d_{ij} - D_{ij})^2}{w_{ij}}, \tag{C–3}$$

where w_{ij} is a weight function to emphasize the preservation of large or small distances in the mapping.

Some of the manifold learning methods also fall into this category. **Locally linear embedding (LLE)** [74] is a two-stage method that looks for a global coordinate system that can represent well the local geometry extracted from local neighborhoods. In the first stage, a set of weights $\mathbf{W}_i$ that minimizes $\sum_{j\in\mathcal{N}_i} \|x_i - W_{ij}x_j\|2$ is obtained by solving a least squares problem under the translation invariance constraint $\sum_{j\in\mathcal{N}_i} W_{ij} = 1$ for all i. In the second stage, similar to MDS, we optimize the location of a set $S_z = \{z_i\}_{i=1}^N \subset \mathbb{R}^p$ by minimizing $\sum_{i=1}^N \sum_{j\in\mathcal{N}_i} \|x_i - W_{ij}x_j\|2$ subject to $\mathrm{E}[ZZ^T] = \mathbf{I}$. The free parameter is the choice of the neighborhoods $\mathcal{N}_i$. The method is very appealing since it can solve a

highly non-linear problem by solving linear problems. However, the method may suffer degeneracies for lattice organization of points as well as not being good at preserving global structure since very far away points in the input space may be mapped to close location in the output space.

Isomap [85] is also a two-stage method. First, it computes an approximation to the geodesic distance on a manifold based on the shortest path distance in a graph constructed by local neighborhoods of each data point. The obtained distance is mapped to points in $\mathbb{R}^p$ using MDS. Unlike LLE, the isomap algorithm has convergence guarantees for some special classes of manifolds. Nevertheless, the algorithm cannot compute a good map if the intrinsic parameter space is not convex, or the manifold is not uniformly sampled, or if the curvature is invariant under local isometry. Also the complexity increases quadratically with the number of points, and sub-sampling methods are not good alternatives since they jeopardize the accuracy of the approximation of the geodesic distance. More recent algorithms, **semidefinite embedding** [92], **minimum volume embedding** [81], are also instances of two-stage methods. They estimate a centered similarity matrix $\mathbf{K}$ under local constraints and positive definiteness, the objective function operates on the spectrum of $\mathbf{K}$ and can be solved via semidefinite programming. For semidefinite embedding, the cost function is $\mathrm{rm}(\mathbf{K})$ which is equivalent to the sum of the eigenvalues of $\mathbf{K}$. Since the matrix is centered, this is equivalent to maximizing the variance of the underlying set of points generating $\mathbf{K}$. Minimum volume embedding goes one step beyond by maximizing the gap between the sum of p largest eigenvalues of $\mathbf{K}$ and the sum of the $N - p$ remaining eigenvalues. In both algorithms, once the similarity matrix is learned, a map to $\mathbb{R}^p$ is computed as in kernel PCA. It has been argued that these techniques are not guaranteed to preserve local and global structure at the same time. Moreover, the assumption data lying on a manifold is not clear in the presence of disjoint classes. The computational complexity of these methods is cubic and their robustness against noise has not been addressed.

The t-distributed version **stochastic neighbor embedding (SNE)** [88] is a powerful algorithm to visualize high-dimensional data. Instead of preserving the distances directly SNE [32] converts the distances between $x_i \in \mathcal{X}$ and the other points into similarities based on conditional probabilities $p_{j|i}$ that x_i would pick x_j as a neighbor, and then optimize the locations of points in $z_i \in \mathbb{R}^p$ such that the conversion of their distance to conditional probabilities $q_{j|i}$ are a good match to $p_{j|i}$ using $\mathrm{KL}(P\|Q)$ as the cost function. The t-SNE is a modification of the original method to alleviate the crowding problem. First the function that models probabilities in $\mathcal{Z}$ is heavy-tailed forcing large distances in $\mathcal{X}$ to remain relatively large in $\mathcal{Z}$. Also, the conditional probabilities are replaced by joint probabilities to make the cost symmetric. The results for visualization are comparable with other state of art methods; however, the usefulness of the technique has not been evaluated for other subsequent tasks and on target spaces of more than three dimensions. Besides, in some cases the intrinsic dimensionality of the manifold can be very high and a lower dimensional embedding may not be possible [2] .

C.2 Decoder Learning Methods

In this case, we assume there is a generative model for X for which Z is a latent variable. Since the latent variables are considered unobserved they are estimated from the decoder. The model assumes a class of decoders along with a code distribution.

Gaussian mixture models (GMM) are a simple type of latent variable models. In this generative model, Z is a discrete variable taking k possible values each one of them associated with a Gaussian component of the mixture. The EM algorithm is the most popular method for learning the mixture model. The expectation step assign *responsibilities* to each observed data point x_i. The maximization step update the parameters of the model by taking into account the responsibilities via maximum

[2] An embedding is an injective map by definition

likelihood. Once the model has been learned, we can code new inputs by partitioning the space based on maximum a posteriori using Bayes theorem.

A unifying view of **Linear generative models** is presented in [69]. In this view, a pair of linear state space and observation models are used as a generative model for the observed data. The model is written as [3] :

$$\begin{aligned} s_{t+1} &= Bs_t + z_t \\ x_t &= Cs_t + \epsilon_t, \end{aligned} \tag{C–4}$$

This model comprises various static latent variable models such as **factor analysis**, **probabilistic PCA**, particular cases of the GMM, and even ICA models by including nonlinearities in the state equation of the model. This generalization also consider dynamic models such Hidden Markov models and Kalman filters. In factor analysis, $B = 0$ and $z_t \in \mathbb{R}^p$ is time independent and Gaussian distributed with zero mean and covariance **I**; the observation model $C : \mathbb{R}^p \mapsto \mathbb{R}^d$ is a linear map and ϵ_t is a time independent Gaussian vector with zero mean and diagonal covariance matrix **R**. Probabilistic PCA has a similar model to factor analysis but ϵ is isotropic, that is, the covariance matrix is $\sigma_\epsilon^2\mathbf{I}$. In the case of ICA, the state equation is $s = g(z)$, with $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Notice the distinction between learning the encoder model as presented above and the decoder model. The encoder model is the limiting case for $\epsilon \to 0$ and thus the learning can be done in terms of $\mathbf{A} = \mathbf{C}^{-1}$, the decoder scheme can be estimated in the presence of noise via EM. However computation of the posterior usually becomes intractable and thus sampling methods should be considered.

The **generative topographic map (GTM)** [86] is a topologically constrained model that solves many of the issues of the SOM. This a generative model for $\mathcal{X}$, with a prior

[3] Note the changes in notation with respect top literature on control systems theory where it is customary to use x to denote the states and y for the outputs of the system.

$p(z)$ on $\mathcal{Z}$ a parametric class of continuous functions $\mathcal{G} = \{g_w : \mathcal{Z} \mapsto \mathcal{X} |\ w \in \Omega\}$ and a conditional probability $p(x|z, w, \theta)$ where θ are the parameters of noise model that follows the map $g(z)$. Bayesian treatment of the GTM may include a prior on $\mathcal{G}$. In the original formulation, parameters w and θ are adjusted by maximizing the log likelihood of the observed x's by integrating over $\mathcal{Z}$, that is, $p(x|w, \theta) = \int_{\mathcal{Z}} p(x|z, w, \theta)p(z)\mathrm{d}z$. In order to make computations tractable, the GTM assumes a prior of the form $p(x) = \frac{1}{p}\sum_{i=1}^{p} \delta(z - z_i)$ where locations of z_i are usually taken to form a uniform grid in $\mathcal{Z}$.

Gaussian process latent variable models (GP-LVM) [43] is a Bayesian latent variable model based on a dual interpretation of probabilistic PCA. Recall the generative model for probabilistic PCA, $x = \mathbf{C}z + \epsilon$ where $z \sim \mathcal{N}(0, \mathbf{I})$ and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2\mathbf{I})$, assume $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Z} = \mathbb{R}^p$. In a Bayesian framework, we can include a prior on $\mathbf{C}$ of the form $p(\mathbf{W}) = \prod_{i=1}^{d} \mathcal{N}(\mathbf{w}_i|0, \mathbf{I})$. Since simultaneous marginalization over $\mathcal{Z}$ and $\mathbf{W}$ is intractable, the dual formulation marginalizes over $\mathbf{W}$, therefore we obtain a likelihood function of the form $p(\mathbf{X}|\mathbf{Z}, \sigma_\epsilon^2) = \prod_{i=1}^{d} p(\mathbf{x}^{(i)}|\mathbf{Z}, \sigma_\epsilon^2)$, where $\mathbf{Z}$ is the matrix representation of a set of N latent points associated with the observed set $S = \{x_i\}_{i=1}^{N}$ also arranged in the matrix $\mathbf{X}$, and $\mathbf{x}^{(i)}$ is the i-th column of $\mathbf{X}$. It turns out that $p(\mathbf{x}^{(i)}|\mathbf{Z}, \sigma_\epsilon^2) = \mathcal{N}(\mathbf{x}^{(i)}|0, \mathbf{K})$, where $\mathbf{K} = \mathbf{Z}\mathbf{Z}^T + \sigma_\epsilon^2\mathbf{I}$. The optimal $\mathbf{Z}$ is calculated according to the log likelihood:

$$-\frac{dN}{2}\log 2\pi - \frac{d}{2}\log|\mathbf{K}| - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T) \tag{C–5}$$

This formulation allows a straightforward nonlinear extension by constructing $\mathbf{K}$ using positive definite kernels, then $K_{ij} = \kappa(z_i, z_j) + \sigma_\epsilon^2\delta_{ij}$ and thus the relation between z and x can be nonlinear. Moreover, the probabilistic formalism allows the use of other noise models rather than Gaussian, but this comes at a price, approximate inference must be employed. An algorithm based on direct optimization of (C–5) does not scale well with $\mathcal{O}(N^3)$, to circumvent this problem sparse approximations of $\mathbf{K}$ are needed. The optimization procedure is also prone to local minimum and initialization procedures must be considered.

The work on **Sparse coding** [54] and **Sparse coding with over-complete basis** [53] describe plausible mechanisms that can explain the properties of receptive fields of simple cells in mammalian primary visual cortex. Here, decoding process is linear generative model $x = Cz + \epsilon$ with a sparsity constraint on the values the variable z can take. The sparse representation forces a large portion of the p dimensions of z to take zero values. This allows cases where $p > d$, that is, C can contain an over-complete basis. Given the basis through C, the code $z \in \mathbb{R}^p$ for a particular $x \in \mathbb{R}^d$ is found by solving the objective:

$$\min_{z \in \mathbb{R}^p} \|x - Cz\|^2 + \lambda \sum_{i=1}^{p} S(z^{(i)}) \tag{C–6}$$

where $z^{(i)}$ denotes the i-th dimension of z. This objective corresponds to a maximum likelihood estimation of z assuming ϵ is isotropic uncorrelated Gaussian and the prior on z is a product of marginals of the form $\frac{1}{Z_{\text{part}}} \exp -\beta S(z^{(i)})$ that correspond to high kurtosis distributions (see [52] for details). An important observation is that the encoding procedure associated with this decoding learning scheme is non-linear. In practice, the basis also needs to be estimated. For such purpose interleaved optimization of codes and basis is carried out. In short, given a fixed C, we solve (C–6) for all x_i's in the training set; using the code vectors z_i we minimize $\sum_{i=1}^{N} \|x - Cz\|^2$ with respect to C. As a consequence, the overall optimization is prone to local minimum if both C and z_i are to be learned. Under the maximum likelihood assumption these models try to minimize the KL divergence between the output distribution of teh generative model and the true distribution of the data. **Non-negative matrix factorization** [44] is a representation of the observed data in matrix form. The set of N observed points in $\mathbb{R}^d$ is represented by a $N \times d$ matrix $\mathbf{X}$. For a non-negative matrix $\mathbf{X}$, we look for decomposition $\mathbf{ZW}^T$, where size of $\mathbf{W}$ and $\mathbf{Z}$ are $(d \times p)$ and $(N \times p)$, respectively, such that the norm $\|\mathbf{X} - \mathbf{ZW}^T\|$

is minimized [4] . In [44] two types of non-negative factorizations are described: convex and conic. In both, $0 \leq W_{ij} \leq 1$ and $0 \leq Z_{ij}$, but in the convex case the extra constraint $\mathbf{Z1} = \mathbf{1}$ is also imposed.

C.3 Encoder-Decoder or Channel Learning Methods

We have seen that some of the unsupervised learning methods have encoder and decoder formulation, namely PCA vs probabilistic PCA, ICA infomax vs ICA maximum likelihood, and k-means vs GMM. The encoder methods are effective in computing a map from the observed input space $\mathcal{X}$ to the code space $\mathcal{Z}$, however they are not concerned about recovering the input x from the code z. On the other hand, decoder methods are good at recovering x from an inferred code, since they are generative models for the observed data. In the absence of additional information, how can a system recognize the important features? This bring us back informally to the principle of information preservation [47]. A coding scheme effectively preserves the information about the inputs if we can recover them from their coded versions. The encoder-decoder or channel learning scheme is an intuitive way to specify the desired properties of the features, and at the same time, guarantee that such features preserve the information about the observed data. In the encoder-decoder learning, the input data undergoes a series of transformations aimed at extracting the structure conveyed by the redundancy in the input. The preservation of information is measured by a fidelity criterion. The composition encoder-decoder, or the channel, is a map from $\mathcal{X}$ to $\mathcal{X}$, where x denotes the input and $\hat{x}$ the reconstructed version (output of the channel).

The work on **generalized Hebbian learning** presented in [72] and his extension to nonlinear units in [73], are early examples that make explicit the encoder-decoder

[4] Extensions to other measures of discrepancy between $\mathbf{X}$ and $\mathbf{ZW}^T$ can be considered. In [19] algorithms for non-negative matrix factorization based on Bregman divergences are proposed

learning method for unsupervised learning. In the linear case, it was shown that the algorithm has the optimality properties of PCA (Karhunen-Loeve tranform). Instead of measuring the mutual information between X and $\hat{X}$, the objective function is the expected squared Euclidean distance between x and its after-channel version $\hat{x}$. The assumption is that the structure of the input in $\mathbb{R}^d$ can be captured by a variable taking values on $\mathbb{R}^p$. In the linear case $p \ll d$ but in the nonlinear case this need not be the case.

Within the sparse coding literature the **robust coding** point of view presented in [20] is another example of the encoder-decoder method. The robustness in the encoding is enforced by introducing additive noise to the coded inputs. Consider a linear model $\mathbf{z} = \mathbf{A}\mathbf{x}$ for which want to decode $\hat{\mathbf{z}} = \mathbf{z} + \boldsymbol{\epsilon}$. The decoded version of $\mathbf{x}$, is also a linear procedure $\hat{\mathbf{x}} = \mathbf{C}\hat{\mathbf{z}}$. Adding the noise has a biological motivation of limiting the information capacity of the processing units. As we mentioned before, sparseness involves a nonlinear operation and needs to be explicitly stated in the cost function. The objective function for sparse robust coding is,

$$\mathrm{E}[\|\hat{X} - X\|] - \lambda_1 \sum_{i=1}^{p} \mathrm{E}[\log p(u^{(i)})] + \lambda_2 \sum_{i=1}^{p} \left[\log \frac{\mathrm{E}[u^{(i)}]}{\sigma_\epsilon^2}\right]^2 \tag{C–7}$$

The first term is the information preservation by fidelity of the recovered input. The second term is a sparseness constraint; this is an entropy term and this does not enforce sparseness per se. The third term, that constrains the power of the code, can pull the peak of the distribution of Z towards the origin. Important remark is that the noise term will produce encoding procedures that range from noiseless ICA to PCA as the power of the noise term increases with respect to the code power.

In **energy-based model** learning, energy functions are parametrized families of functions with low energy values on certain regions of their domains. In unsupervised learning, the energy functions are optimized such that low energies will be associated with regions where training data lies and high energy values will be given to regions

where no data has been observed. Once these energy-models have been learned, inference can be done by returning the point closest local energy minimum to an input x. Common examples of energy functions are norms, log likelihood. The second one is associated with Gibbs distribution of the form $\frac{1}{Z_E} \exp -\beta E(X)$. One of the advantages of the energy-model based learning, compared to probabilistic models, is that it eliminates normalization of the functions, which can be computationally expensive or intractable. When combining energy terms, it is important to consider the scale at which each of the terms work since improper trade-off may lead to poor results. The energy-based model framework was adopted for unsupervised learning in [65] that allows the formulation of encoder-decoder schemes as a combination of energy terms for inference (encoding), code cost (sparseness), and reconstruction cost (decoding). Namely,

- Code prediction energy $E_C[X, Z, \mathbf{A}]$
- Reconstruction energy (decoder) $E_D[X, \hat{Z}, \mathbf{C}]$
- Code cost $E_Z[c(\hat{Z})]$ this penalizes undesired forms for the code, properties such as sparseness will then have low energy values.

The energy function is the sum of the above terms:

$$E_C[X, Z, \mathbf{A}] + E_D[X, \hat{Z}, \mathbf{C}] + E_Z[c(\hat{Z})] \tag{C–8}$$

Even after the energy-model has been learned, inference for coding is still problematic since it involves the code cost. To allow fast inference, [66] removes the code cost term and trains a soft max sparsifying function. By using a fast inference encoder, it is possible to build deep architectures that compute features efficiently.

Encoder-decoder methods we have reviewed so far, have explicit encoder and decoder functions. We can also, combine them into one mapping from $\mathcal{X}$ to $\mathcal{X}$. In this case, the intermediate variable z is implicit; the information provided by the output variable $\hat{x}$ about the input x serves as a lower bound on the information the implicit variable z contains about x. It is common across all the above presented methods

that unsupervised learning has the goal of preserving the information conveyed by the observed inputs. At the same time, it assumes that all the uncertainty carried by the input data need not be preserved. Finally, the encoder-decoder learning scheme also suggests that $\hat{x}$ should not carry more uncertainty than necessary to carry the information about x.

APPENDIX D
AN EFFICIENT RANK-DEFICIENT COMPUTATION OF THE PRINCIPLE OF RELEVANT INFORMATION

A major issue, which we address in this work, is that the amount of computation associated to the PRI grows quadratically with the size of the available sample. This limits the scale of the applications if one were to apply the formulas directly. The problem of polynomial growth on complexity has also received attention within the machine learning community working on kernel methods. Consequently, approaches to compute approximations to positive semidefinite matrices based on kernels have been proposed [23, 94]. The goal of these methods is to accurately estimate large Gram matrices without computing their n^2 elements, directly. It has been observed that in practice the eigenvalues of the Gram matrix drop rapidly and therefore replacing the original matrix by a low rank approximation seems reasonable[5, 23]. In our work, we derive an algorithm for the principle of relevant information based on rank deficient approximations of a Gram matrix. We also propose a simple modified version of the Nyström method particularly suited for estimation in ITL.

The chapter starts with a brief introduction to Renyi's Entropy and the associated information quantities with their corresponding rank deficient approximations. Then, the objective function for the principle of relevant information (PRI) is presented. Following, we propose an implementation of the optimization problem based on rank deficient approximations. The algorithm is tested on simulated data for various accuracy regimes (different ranks) followed by some results on realistic scenarios. Finally, we provide some conclusions along with future work directions.

D.1 Rank Deficient Approximation for ITL

D.1.1 Renyi's α-Order Entropy and Related Functions

In information theory, a natural extension of the commonly used Shannon's entropy is α-order entropy proposed by Renyi [70]. For a random variable X with probability

density function (PDF) $f(x)$ and support $\mathcal{X}$, the α-entropy $H_\alpha(X)$ is defined as;

$$H_\alpha(f) = \frac{1}{1-\alpha} \log \int_{\mathcal{X}} f^\alpha(x) \mathrm{d}x. \tag{D–1}$$

The case $\alpha \to 1$ gives Shannon's entropy. Similarly, a modified version of Renyi's definition of α-relative entropy between random variables with PDFs f and g is given in [48],

$$D_\alpha(f\|g) = \log \frac{\left(\int g^{\alpha-1} f\right)^{\frac{1}{1-\alpha}} \left(\int g^\alpha\right)^{\frac{1}{\alpha}}}{\left(\int f^\alpha\right)^{\frac{1}{\alpha(1-\alpha)}}}. \tag{D–2}$$

likewise, $\alpha \to 1$ yields Shannon's relative entropy (KL divergence). An important component of the relative entropy is the cross-entropy term $H_\alpha(f; g)$ that quantifies the information gain from observing g with respect to the "true" density f. It turns out that for the case of $\alpha = 2$, the above quantities can be expressed, under some restrictions, as a function of the inner product between PDFs. In particular, the 2-order entropy of f and cross-entropy between f and g, are:

$$H_2(f) = -\log \int_{\mathcal{X}} f^2(x) \mathrm{d}x, \tag{D–3}$$

$$H_2(f; g) = -\log \int_{\mathcal{X}} f(x) g(x) \mathrm{d}x. \tag{D–4}$$

The associated relative entropy of order 2 is called the Cauchy-Schwarz divergence and is defined as follows:

$$D_{\mathrm{CS}}(f\|g) = -\frac{1}{2} \log \frac{\left(\int fg\right)^2}{\left(\int f^2\right)\left(\int g^2\right)}. \tag{D–5}$$

The above operations assume that f and g are know, which is almost never the when learning from data. Plug in estimators of the second order Renyi's entropy and cross-entropy (Cauchy-Schwarz (D–5)) can be derived using Parzen denstity estimators. For an *i.i.d.* sample $S = \{\mathbf{x}_i\}_{i=1}^n \subseteq \mathbb{R}^p$ drawn from g, the Parzen density estimator $\hat{g}$ at x is given by $\hat{g}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \kappa(\mathbf{x}, \mathbf{x}_i)$, where κ is an admissible kernel [59]. Consider two samples $S_1 = \{\mathbf{x}_i\}_{i=1}^n$ and $S_2 = \{\mathbf{y}_i\}_{i=1}^m$, both in $\mathbb{R}^p$, drawn *i.i.d.* from g and f, respectively.

Let $\mathbf{K}_1$ be a matrix with all pairwise evaluations of κ on S_1, that is $\mathbf{K}_1(i,j) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$; the estimate of the entropy is $\hat{H}(g) = -\log \frac{1}{n^2}^\top \mathbf{K}_1$, $\hat{H}(f)$ can be derived similar fashion from matrix $\mathbf{K}_2$. Cross-entropy is $\hat{H}(f;g) = -\log \frac{1}{nm}^\top \mathbf{K}_{12}$, where $\mathbf{K}_{12}(i,j) = \kappa(\mathbf{x}_i, \mathbf{y}_j)$. Note that we are basically estimating the arguments of the $\log$ functions in (D–3) and (D–4); we will refer to them as information potential (IP) and cross-information potential (CIP) [62]. For positive semidefinite kernels that are also Parzen, $\mathbf{K}$'s are Gram matrices.

D.1.2 Rank Deficient Approximation

Any symmetric positive semidefinite matrix $\mathbf{A}$ can be written as the product $\mathbf{G}\mathbf{G}^\top$. Note the decomposition $\mathbf{A}$ need not be unique.

Incomplete Cholesky Decomposition: This decomposition is a special case of the LU factorization is know as the *Cholesky decompositon* [28]. Here, $\mathbf{G}$ is a lower triangular matrix with positive diagonal. The advantage of this decomposition is that we can approximate our Gram matrix $\mathbf{K}$ with arbitrary accuracy ϵ by choosing a lower triangular matrix $\tilde{\mathbf{G}}$ with d columns such that $\|\mathbf{K} - \tilde{\mathbf{G}}\tilde{\mathbf{G}}^\top\| \leq \epsilon$ (For a suitable matrix norm). This incomplete Cholesky decomposition (ICD) can be computed by a greedy approach that minimizes the trace of the residual of $\mathbf{K} - \tilde{\mathbf{G}}\tilde{\mathbf{G}}^\top$ [5, 23, 78]. For an $n \times n$ matrix the complexity of this method is $O(nd)$ and the time complexity is $O(nd^2)$. Therefore, this algorithm is preferable only when $d^2 \ll n$. The error of the ICD based estimators can be easily bounded. For a positive semidefinite matrix we have that $\|\mathbf{A}\|_2 \leq \mathbf{A}$, which for error matrix $(\mathbf{K} - ^\top) \leq \epsilon$. the estimators treated in this paper are mostly of the form $\mathbf{a}^\top\mathbf{b}$ and so the error is bounded by $\epsilon\|\mathbf{a}\|\|\mathbf{b}\|$.

Nyström Aproximation: This is a well know rank deficient approximation to $\mathbf{K}$ in machine learning [93]. The approximate Gram matrix $\tilde{\mathbf{K}}$ is computed by projecting all the data points onto a subspace spanned by a random subsample of size d in the feature space. Consequently $\tilde{\mathbf{K}} = \mathbf{K}_d\mathbf{K}_{dd}^{-1}\mathbf{K}_d^\top$, where $\mathbf{K}_d$ is the kernel evaluation between all data points and the subsample of size d, and $\mathbf{K}_{dd}$ is the kernel evaluation between all the points in the subsample. The price we pay for this simplicity is that the accuracy of

the approximation cannot be simply determined. An improved version of this method with error guarantee can be found in [21]. One important remark on the Nyström method relates to the computation of $\mathbf{K}_{dd}^{-1}$, for which we can employ the eigen-decomposition of $\mathbf{K}_{dd} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$.

Nyström-KECA: Assume we want to reduce even further the size of $\mathbf{K}_{dd}$ based on its eigenvalues, we will have a good projection in terms of the squared norm in the reproducing kernel Hilbert space associated to κ if we pick the columns of $\mathbf{U}$ corresponding to the largest eigenvalues on the diagonal of $\mathbf{\Lambda}$. However, we are more interested on the projection of the mean of the mapped data points $\boldsymbol{\mu}_\Phi$. This idea resembles the approach followed by [26] on the stopping criterion for the orthogonal series density estimation based on Kernel PCA. The matrix $\mathbf{K}_d\mathbf{K}_{dd}^{-1}\mathbf{K}_d^\top$ represents an Ansatz product in the feature space $\mathcal{H}$ as $\langle \Phi(\mathbf{x}), \mathrm{P}\Phi(\mathbf{y})\rangle$, where P is a projection operator. In particular $\langle \boldsymbol{\mu}_\Phi, \mathrm{P}\boldsymbol{\mu}_\Phi\rangle \leq \langle \boldsymbol{\mu}_\Phi, \boldsymbol{\mu}_\Phi\rangle$. We can find a faster convergence series than the one obtained by ordering the eigenvalues of $\mathbf{K}_{dd}^{-1}$ in a non-increasing way. Such series can be created by ordering the columns of $\mathbf{U}$ and their respective eigenvalues according to the score $s_i = \lambda_i^{-1/2}[\mathbf{u}_i^\top \mathbf{K}_d^\top]^2$. We call this decomposition Nyström-KECA because it resembles the kernel entropy component analysis proposed in [36].

The computation of the estimators of the IP and CIP can be easily carried out by computing a low rank decomposition of the $(n+m)\times(n+m)$ Gram matrix $\mathbf{K}$ of the augmented sample $S = \{x_i\}_1^n \cup \{y_i\}_1^m$,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_{12} \\ \mathbf{K}_{12}^\top & \mathbf{K}_2 \end{bmatrix} \tag{D–6}$$

Since $\mathbf{K} \approx \tilde{\mathbf{G}}\tilde{\mathbf{G}}^\top$ with $d \ll n+m$. The block array in (D–6) can be expressed in sub-blocks of $\tilde{\mathbf{G}}$ as[1] :

$$\begin{bmatrix} \tilde{\mathbf{G}}_1 \\ \tilde{\mathbf{G}}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{G}}_1^\top, & \tilde{\mathbf{G}}_2^\top \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{G}}_1\tilde{\mathbf{G}}_1^\top & \tilde{\mathbf{G}}_1\tilde{\mathbf{G}}_2^\top \\ \tilde{\mathbf{G}}_2\tilde{\mathbf{G}}_1^\top & \tilde{\mathbf{G}}_2\tilde{\mathbf{G}}_2^\top \end{bmatrix} \tag{D–7}$$

Then for the IP $\frac{1}{n^2}^\top \mathbf{K}_1 \approx \frac{1}{n^2}^\top \mathbf{G}_1\tilde{\mathbf{G}}_1^\top$, and for the CIP $\frac{1}{nm}^\top \mathbf{K}_1 \approx \frac{1}{nm}^\top \mathbf{G}_1\tilde{\mathbf{G}}_2^\top$. Note that computing CIP needs roughly $\mathcal{O}((n+m)d^2)$ operations rather than $\mathcal{O}(nm)$. However, it seems redundant to indirectly work with a $(n+m)\times(n+m)$ matrix while we are only interested in a $n \times m$ block. However, our problem requires both IP and CIP.

D.2 The Principle of Relevant Information

Regularities on the data can be attributed to structure in the underlying generating process. These regularities can be quantified by the entropy estimated from data, hence, we can think of the minimization of entropy as a means for finding such regularities. Suppose we are given a random variable X with PDF g, for which we want to find a description in terms of a PDF f with reduced entropy, that is, a variable Y that captures the underlying structure of X. The principle of relevant information (PRI) casts this problem as a trade-off between the entropy of Y $H_2(f)$ and its descriptive power about X in terms of their relative entropy $D_{\mathrm{CS}}(f\|g)$. The principle can be briefly understood as a trade-off between the minimization of redundancy preserving most of the original structure of a given probability density function. For a fixed pdf $g \in \mathcal{F}$ the objective is given by:

$$\arg\min_{f\in\mathcal{F}}[H_2(f) + \lambda D_{\mathrm{CS}}(f\|g)] \text{ for } \lambda \geq 1. \tag{D–8}$$

The trade off parameter λ defines various regimes for this cost functions ranging from clustering ($\lambda = 1$) to a reminiscent principal curves ($\lambda \approx p$) to vector quantization ($\lambda \to \infty$).

[1] Sub-blocks cannot be computed individually from $\mathbf{K}_1$ and $\mathbf{K}_2$

PRI as a Self Organization Mechanism: A solution to the above search problem was proposed in [68]. The method combines Parzen density estimation with a self organization of a sample to match the desired density f that minimizes (D–8). The optimization problem becomes:

$$\min_{Y \in (\mathbb{R}^p)^m} [\hat{H}_2(Y) + \lambda \hat{D}_{\mathrm{CS}}(Y \| X)] \tag{D–9}$$

where $X \in (\mathbb{R}^p)^n$ is a set of p-dimensional points with cardinality n, and Y a set of p-dimensional points with cardinality m. Problem (D–9) is equivalent to:

$$\min_{Y \in (\mathbb{R}^p)^m} [(1-\lambda)\hat{H}_2(Y) + 2\lambda \hat{H}_2(Y;X)] = \min_{Y \in (\mathbb{R}^p)^m} J_\lambda(Y) \tag{D–10}$$

For the Gaussian kernel $\kappa_\sigma(\mathbf{x},\mathbf{y}) = \exp -\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{y}\|^2$ we can evaluate the cost (D–10) as [2]:

$$-(1-\lambda)\log \frac{1}{m^2}^\top \mathbf{K}_2 - 2\lambda \log \frac{1}{mn}^\top \mathbf{K}_{12} \tag{D–11}$$

The self organization principle moves each particle $\mathbf{y}_i$ according to the forces exerted by the samples X and Y. The entropy minimization creates attractive forces among the $\mathbf{y}_i$'s and the sample X induces a force field that restrict the movement of each $\mathbf{y}_i$. Computing the partial derivatives of (D–11) with respect to each point $\mathbf{y}_r \in Y$ yields:

$$\frac{\partial}{\partial \mathbf{y}_r} \log \frac{1}{m^2}^\top \mathbf{K}_2 = \frac{2\sum_{i=1}^{m} \kappa_\sigma(\mathbf{y}_r,\mathbf{y}_i)\left(\frac{\mathrm{y}_i-\mathrm{y}_r}{\sigma^2}\right)^\top}{^\top \mathbf{K}_2} \tag{D–12}$$

$$\frac{\partial}{\partial \mathbf{y}_r} \log \frac{1}{mn}^\top \mathbf{K}_{12} = \frac{\sum_{i=1}^{n} \kappa_\sigma(\mathbf{y}_r,\mathbf{x}_i)\left(\frac{\mathrm{x}_i-\mathrm{y}_r}{\sigma^2}\right)^\top}{^\top \mathbf{K}_{12}} \tag{D–13}$$

[2] Note that the employed kernel does not integrate to one. This is not a problem since a normalization factor becomes an additive constant in the objective function.

A direct optimization of the cost in (D–11) is computationally burdensome and only feasible for samples sizes up to a few thousands, this limits the applicability of the principle. Here, we want to overcome this limitation by allowing computation on larger samples commonly encountered in signal processing applications. Below, we develop a way of incorporating the rank deficient approximations to the gradient of the PRI cost function. It is important to remind the reader that this solution can be easily adapted to other ITL objectives.

Consider the following identity: For $\mathbf{A} = \mathbf{a}\mathbf{b}^\top$, where $\mathbf{a}$ and $\mathbf{b}$ are column vectors,

$$\mathbf{C} \circ \mathbf{A} = \mathrm{diag}(\mathbf{a})\mathbf{C}\mathrm{diag}(\mathbf{b}) \tag{D–14}$$

where $\mathrm{diag}(\mathbf{z})$ denotes a diagonal matrix with the elements of $\mathbf{z}$ on the main diagonal, for simplicity we denote $\mathrm{diag}(\cdot)$ as $\mathrm{d}(\cdot)$.

We will restrict the analysis to the Gaussian kernel but a similar treatment for other kernels with similar properties on their derivative can be adopted. Let $\mathbf{\Delta Y}^{(k)}$ and $\mathbf{\Delta X}^{(k)}$ be matrices with entries $\Delta Y_{ij}^{(k)} = \mathbf{y}_i^{(k)} - \mathbf{y}_j^{(k)}$ and $\Delta X_{ij}^{(k)} = \mathbf{x}_i^{(k)} - \mathbf{y}_j^{(k)}$, respectively. By $\mathbf{z}_i^{(k)}$, we mean the k-th component of the vector $\mathbf{z}_i$. Equations (D–12) and (D–13) can be re-expressed and combined for all $r = 1, \ldots, m$ using matrix operations as:

$$\frac{\partial}{\partial \mathbf{y}^{(k)}} \hat{H}_2(Y) = -\frac{2}{\sigma^2} \frac{^\top\left(\mathbf{K}_2 \circ \mathbf{\Delta Y}^{(k)}\right)}{^\top\mathbf{K}_2} \tag{D–15}$$

$$\frac{\partial}{\partial \mathbf{y}^{(k)}} \hat{H}_2(Y;X) = -\frac{1}{\sigma^2} \frac{^\top\left(\mathbf{K}_{12} \circ \mathbf{\Delta X}^{(k)}\right)}{^\top\mathbf{K}_{12}} \tag{D–16}$$

decomposing $\mathbf{\Delta Y}^{(k)} = \mathbf{y}^{(k)\top} - \mathbf{y}^{(k)^\top}$ and $\mathbf{\Delta X}^{(k)} = \mathbf{x}^{(k)}\mathbf{1}^\top - \mathbf{y}^{(k)^\top}$, where $\mathbf{y}^{(k)} = (\mathbf{y}_1^{(k)}, \mathbf{y}_2^{(k)}, \ldots, \mathbf{y}_m^{(k)})^\top$ and $\mathbf{x}^{(k)} = (\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \ldots, \mathbf{x}_m^{(k)})^\top$, and applying the identity (D–14) on (D–15) and (D–16), after some algebra, yields:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{y}^{(k)}} J_\lambda(Y) = a\left(\mathbf{y}^{(k)^\top}\mathbf{K}_2 - ^\top\mathbf{K}_2\mathrm{d}(\mathbf{y}^{(k)})\right) + \\ + b\left(\mathbf{x}^{(k)^\top}\mathbf{K}_{12} - ^\top\mathbf{K}_{12}\mathrm{d}(\mathbf{y}^{(k)})\right),\end{aligned} \tag{D–17}$$

where $a = -2(1-\lambda)/(\sigma^{2\top}\mathbf{K}_2)$ and $b = -2\lambda/(\sigma^{2\top}\mathbf{K}_{12})$. Finally, it is easy to verify that for the rank deficient approximation $\mathbf{K} \approx \tilde{\mathbf{G}}\tilde{\mathbf{G}}^\top$:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{y}^{(k)}} J_\lambda(Y) \approx \left(a\mathbf{y}^{(k)\top}\tilde{\mathbf{G}}_2 + b\mathbf{x}^{(k)\top}\tilde{\mathbf{G}}_1\right)\tilde{\mathbf{G}}_2^\top + \\ - \left(a^\top\tilde{\mathbf{G}}_2 + b^\top\tilde{\mathbf{G}}_1\right)\tilde{\mathbf{G}}_2^\top \mathrm{d}(\mathbf{y}^{(k)}),\end{aligned} \tag{D–18}$$

The last equation can be computed in $\mathcal{O}(\max\{n, m\}d)$, d is the rank of $\tilde{\mathbf{K}}$, instead of $\mathcal{O}(\max\{nm, m^2\})$ which is the complexity of the fixed point algorithm presented in [68].

D.3 Experiments

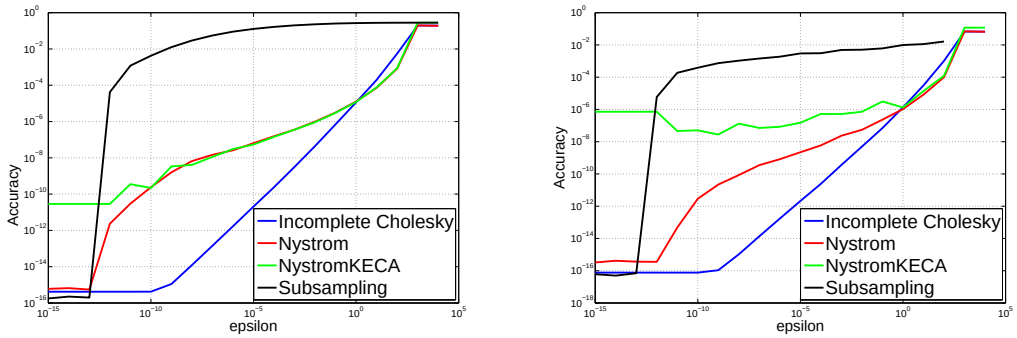
The experimental setup for the above methods is divided into two stages; in order to assess the performance of the methodology in terms of accuracy we test the rank deficient estimation algorithms on simulated data. Then, we apply our implementation methodology for the PRI in two real scenarios, namely, automatic image segmentation and signal denoising.

D.3.1 Simulated Data

For the simulated data we compute the information potential and the cross-information potential of a mixture of two unitary isotropic Gaussian distributions in a four-dimensional space. The mean vectors are $(1, 1, 1, 1)^\top$ and $-(1, 1, 1, 1)^\top$. The kernel size is set to be $\sigma = 2$ and the size of the set is $n = 1000$. Figure D-1 displays the performance of the estimators for different ranks that are related to the accuracy level set for the incomplete Cholesky decomposition. Notice that the performance of the Nyström-KECA remains constant, albeit slightly worse than pure Nyström. This is because we drop vectors with low scores as described in section D.1.2. Therefore we sacrifice more accuracy to lower the rank even further.

D.3.2 Image Segmentation Signal Denoising with the PRI

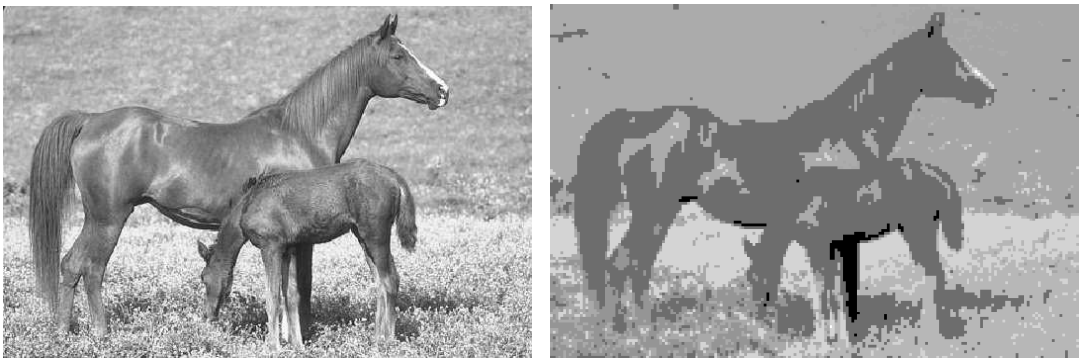
Automatic image segmentation is usually seen as a clustering problem, where spatial and intensity features are combined to discern objects from background. A well established procedure for image segmentation and for clustering in general is the



A IP B CIP

Figure D-1. Accuracies for the IP and CIP estimators

Gaussian mean shift (GMS). It has been shown that GMS is a especial case of the PRI objective when the trade off parameter λ is set to 1. Treating images as collection of pixels is a fairly challenging task due to the amount of points to be processed. Figure D-2 shows the segementation results using the PRI optimization described on Section D.2 and the Nyström-KECA rank deficient factorization. The image resolution is 130×194 pixels for a total of 25220 points.



A Original Image B Segmented image using PRI

Figure D-2. Image segmentation using PRI

Setting $\lambda = 1$, as we already mentioned, defines a mode seeking regime for the PRI. For larger values of λ we obtained a solution, for which points concentrate on highly dense regions and are nicely scattered on patterns resembling principal curves of data. The key interpretation is that the estimated distribution gets pulled towards regions of

higher entropy in the manifold of PDFs where we are looking for an optimum of the PRI objective. Figure D-3 shows the resulting denoised signal that was embedded in a two dimensional space along with the contour plots of the estimated PDF. The number of points is 15000.

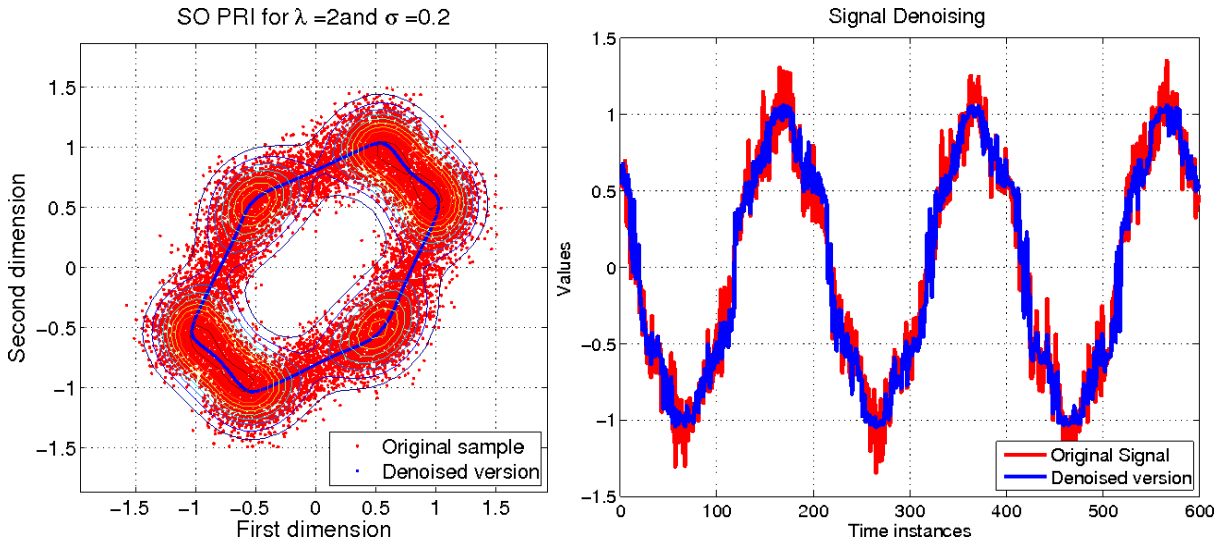


Figure D-3. Noisy and Denoised versions of a periodic signal embedded in a 2 dimensional space

D.4 Remarks

In this chapter we suggest the use of rank deficient approximation to Gram matrices involved in the estimation of ITL objective functions. In particular we focus on the Principle of Relevant Information that requires estimation of second-order entropies and cross-entropies (equivalently information and cross-information potentials) along with their respective derivatives, which are employed during the optimization. We developed a methodological approach to factorize the elements involved in the gradient calculation and these results can be extended to other methods that involve similar forms. Along the lines of rank deficient approximation, we propose a simple modification to the Nyström method that is motivated by the nature of the quantities we want to estimate. The presented methodology allows the application of the PRI on much larger datasets, we

expect this improvement opens new directions for which we can apply this principle. Some of the analysis that lead to the modification of the Nyström-based decomposition raises the question about what kernels can be more useful in the context of Information theoretic learning based on their convergence properties for certain vectors.

REFERENCES

[1] Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. (1964). The method of the potential functions for the problem of restoring the characteristic of a function converter from randomly observed points. *Avtomatika i Telemekhanika*, *25*(12), 1705–1714.

[2] Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Avtomatika i Telemekhanika*, *25*(6), 917–936.

[3] Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*(3), 337–404.

[4] Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, *3*, 1–48.

[5] Bach, F. R., & Jordan, M. I. (2002). Kernel independent component analysis. *JMLR*, *3*, 1–48.

[6] Barlow, H. (1961). Possible principles underlying the transformations of sensory messages. *Sensory Communication*, (pp. 217–234).

[7] Barlow, H. (1989). Unsupervised learning. *Neural Computation*, *1*, 295–311.

[8] Barlow, H., Kaushal, T., & Mitchison, G. (1989). Finding minimum entropy codes. *Neural Computation*, *1*, 412–423.

[9] Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labelled and unlabelled examples. *Journal of machine Learning Research*, *7*, 2399–2434.

[10] Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, *7*, 1129–1159.

[11] Bhatia, R. (1996). *Matrix Analysis*. Graduate Texts in Mathematics. Springer.

[12] Bhatia, R. (2006). Infinite divisible matrices. *The American Mathematical Monthly*, *113*(3), 221–235.

[13] Candocia, F. M., & Principe, J. C. (1999). Super-resolution of images based on local correlations. *IEEE Transactions on Neural Networks*, *10*(2), 372–380.

[14] Chong, E. K. P., & Zak, S. H. (2001). *An Introduction to Optimization*. Discrete Mathematics and Optimization. Wiley Interscience, second ed.

[15] Costa, J. A., & III, A. O. H. (2004). Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transacions on Signal Processing*, *52*, 2210–2221.

[16] Cover, T. M., & Thomas, J. A. (2006). *Elements of Information Theory*. Wiley-Interscience, second ed.

[17] Cucker, F., & Zhou, D. X. (2007). *Learning Theory: An Approximation Theory Viewpoint*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press.

[18] Davis, J. V., Kulis, B., Jain, P., Sra, S., & Dhillon, I. S. (2007). Information-theoretic metric learning. In *ICML*, (pp. 209–216). Corvalis, Oregon, USA.

[19] Dhillon, I. S., & Sra, S. (2000). Generalized non-negative matrix approximations with bregman divergences. In *NIPS*, (pp. 556–562).

[20] Doi, E., & Lewicki, M. S. (2004). Sparse coding of natural images using an overcomplete set of limited capacity units. In *NIPS*, (pp. 377–384).

[21] Drineas, P., & Mahoney, M. W. (2005). On the nyström method for approximating a gram matrix for improved kernel-based learning. *JMLR*, *6*, 2153–2175.

[22] Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification*. John Wiley and Sons, second ed.

[23] Fine, S., & Scheinberg, K. (2001). Efficient svm training using low-rank kernel representations. *JMLR*, *2*, 243–264.

[24] Friedland, S. (1981). Convex spectral functions. *Linear and Multilinear Algebra*, *9*, 299–316.

[25] Gammerman, A., Vovk, V., & Vapnik, V. (1998). Learning by transduction. In *Uncertainty in Artificial Intelligence*, (pp. 148–155).

[26] Girolami, M. (2002). Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, *14*(3), 669 – 688.

[27] Globerson, A., & Roweis, S. (2005). Metric learning by collapsing classes. In *NIPS*.

[28] Golub, G. H., & Van Loan, C. F. (1996). *Matrix Computation*. Baltimore, Maryland: The Johns Hopkins University Press, third ed.

[29] Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In S. Jain, H. Simon, & E. Tomita (Eds.) *Proceedings of Algorithmic Learning Theory*, (pp. 63–77).

[30] Gretton, A., & Györfi, L. (2010). Consistent nonparametric test of independence. *Journal of Machine Learning Research*, *11*, 1391–1423.

[31] Hero, A., & Michel, O. (1998). Robust entropy estimation strategies based on edge weighted random graphs. In *Proceedings of the Meeting of the International Society for Optical Engineering (SPIE)*.

[32] Hinton, G., & Roweis, S. (2003). Stochastic neighbor embedding. In *NIPS 2002*, (pp. 857–864).

[33] Horn, R. A. (1969). The theory of infinitely divisible matrices and kernels. *Transactions of the American Mathematical Society*, *136*, 269–286.

[34] Horn, R. A., & Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.

[35] Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. John Wiley and Sons.

[36] Jensen, R. (2009). Kernel entropy component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[37] Jenssen, R., Eltoft, T., Girolami, M., & Erdogmus, D. (2006). Kernel maximum entropy data transformation and an enhanced spectral clustering algorithm. In *NIPS*, (pp. 633–640).

[38] Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, first ed.

[39] Kato, T. (1987). Variation of discrete spectra. *Communications in Mathematical Physics*, *111*, 501–504.

[40] Kimeldorf, G. S., & Wahba, G. (1971). Some results on the tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, *33*(1), 82–95.

[41] Kreyszig, E. (1978). *Introductory Functional Analysis with Applications*. John Wiley & Sons.

[42] Kwok, J. T., & Tsang, I. W. (2003). The preimage problem in kernel methods. In *Proceeedings of the 20th International Conference on Machine Learning*, (pp. 408–415).

[43] Lawrence, N. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, *6*, 1783–1816.

[44] Lee, D. D., & Seung, H. S. (1997). Unsupervised learning by convex and conic coding. In *NIPS*, (pp. 515–521).

[45] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

[46] Lewis, A. S. (1996). Derivatives of spectral functions. *Mathematics of Operations Research*, *21*(3), 576–588.

[47] Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, *21*, 105–117.

[48] Lutwak, E., Yang, D., & Zhang, G. (2005). Cramér-rao and moment-entropy inequalities for renyi entropy and generalized fisher information. *IEEE Transactions on Information Theory*, *51*(2), 473–478.

[49] Mangasarian, O. (1969). *Nonlinear Programming*. Systems and Science. McGraw-Hill.

[50] Mika, S., Rätch, G., Weston, J., Schölkopf, B., & Müller, K.-R. (1999). Fisher discriminant analysis with kernels. In *IX IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing*, (pp. 41–48).

[51] Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability and its Applications*, *9*(1), 141–142.

[52] Olshausen, B. A. (1996). Learning linear, sparse, factorial codes. Tech. Rep. AI memo 1580, CBCL and AI Lab MIT.

[53] Olshausen, B. A., & Field, D. J. (1997). Sparse coding with and overcomplete basis set: A startegy employed by v1? *Vision Research*, *37*(23), 3311–3325.

[54] Olshausen, B. A., & J.Field, D. (1996). Emergence of simple-cell receptive field property by learning a sparse code for natural images. *Nature*, *381*, 607–609.

[55] Osuna, E., Freund, R., & Girosi, F. (1997). An improved training algorithm for support vector machines. In *IEEE Workshop on neural Networks for Signal Processing*, (pp. 276–285).

[56] Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, third ed.

[57] Park, M. K., & Kang, M. G. (2003). Super-resolution image reconstruction: a technical overview. *IEEE Signal Processing Magazine*, *20*(3), 21–36.

[58] Parzen, E. (1959). Statistical inference on time series by hilbert space methods, i. Tech. Rep. 23, Stanford University.

[59] Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, *33*(3), 1065–1076.

[60] Platt, J. C. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. rep., Microsoft Research.

[61] Principe, J. C. (2010). *Information Theoretic Learning: Renyi's Entropy and Kernel Perspective*. Information Science and Statistics. Springer.

[62] Principe, J. C. (2010). *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Series in Information Science and Statistics. Springer.

[63] Principe, J. C., Xu, D., & III, J. W. F. (2000). *Unsupervised Adaptive Filtering*, chap. Information-Theoretic Learning, (pp. 265–319). John Wiley and Sons.

[64] Pál, D., Póczos, B., & Szepesvári, C. (2010). Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs. In *NIPS*.

[65] Ranzato, M. (2009). *Unsupervised Learning of Feature Hierarchies*. Ph.D. thesis, New York University.

[66] Ranzato, M., Chopra, C. P. S., & LeCun, Y. (2006). Efficient learning of sparse representations with an energy based model. In *NIPS*.

[67] Rao, S., de Medeiros Martins, A., & Principe, J. C. (2008). Mean shift: An information theoretic perspective. *Pattern Recognition Letters*, *30*, 222–230.

[68] Rao, S. M. (2008). *Unsupervised Learning: An Information Theoretic Framework*. Ph.D. thesis, University of Florida.

[69] Roweis, S., & Ghahramani, Z. (1999). A unifying review of linear gaussian models. *Neural Computation*, *11*, 305–345.

[70] Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, (pp. 547–561). Berkeley: University of California Press.

[71] Sanchez-Giraldo, L. G., & Principe, J. C. (2011). A reproducing kernel hilbert space formulation of the principle of relevant information. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*.

[72] Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer feedforward neural network. *Neural Networks*, *2*, 459–473.

[73] Sanger, T. D. (1989). An optimality principle for unsupervised learning. In *NIPS*, (pp. 11–19).

[74] Saul, L. K., & Roweis, S. T. (2003). Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, *4*, 119–155.

[75] Schoenberg, I. J. (1938). Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, *44*(3), 522–536.

[76] Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. MIT Press.

[77] Schölkpf, B., Smola, A. J., & Müller, K.-R. (1996). Nonlinear component analysis as a kernel eigenvalue problem. Tech. Rep. 44, Max-Planck-Institut für biologische Kybernetik.

[78] Seth, S., & Principe, J. (2009). On speeding up computation in information theoretic learning. In *IJCNN*.

[79] Seth, S., Rao, M., Park, I., & Príncipe, J. C. (2011). A unified framework for quadratic measures of independence. *IEEE Transactions on Signal Processing*, *59*(8), 3624–3635.

[80] Shannon, C. E. (1948). A mathematical theory of communication. *The bell System technical Journal*, *27*, 379–423, 623–656.

[81] Shaw, B., & Jebara, T. (2007). Minimum volume embedding. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, (pp. 460–467).

[82] Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

[83] Slonim, N. (2002). *The Information Bottleneck: Theory and Applications*. Ph.D. thesis, Hebrew University.

[84] Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Lanckriet, G., & Schölkopf, B. (2008). Injective hilbert space embeddings of probability measures. In *Proceedings of the 21st Annual Conference on Learning Theory*, (pp. 111–122).

[85] Tenenbaum, J. B., de Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.

[86] Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, *61*(3), 611–622.

[87] Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*, (pp. 368–377).

[88] van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*, 2579–2605.

[89] von Luxburg, U. (2004). *Statistical Learning with Similarity and Dissimilarity Functions*. Ph.D. thesis, Technical University of Berlin.

[90] von Luxburg, U. (2007). A tutorial on spectral clustering. Tech. Rep. 149, Max Plank Institute for Biological Cybernetics.

[91] Weinberger, K. Q., Blitzer, J., & Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *NIPS*.

[92] Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceeding of the Twenty-first International Conference on Machine Learning*, (pp. 106–113).

[93] Williams, C. K., & Seeger, M. (2000). Using the nyström method to speed up kernel machines. In *NIPS*, (pp. 682–688).

[94] Williams, C. K. I., & Seeger, M. (2000). The effect of the input density distribution on kernel-based classifiers. In *ICML*, (pp. 1159–1166).

[95] Xu, J.-W., Paiva, A. R. C., Park, I., & Principe, J. C. (2008). A reproducing kernel hilbert space framework for information theoretic learning. *IEEE Transactions on Signal Processing*, *56*(12), 5891–5902.

BIOGRAPHICAL SKETCH

Luis Gonzalo Sánchez Giraldo was born in 1983 in Manizales, Colombia. He received the B.S. in electronics engineering and M.Eng. in industrial automation from Universidad Nacional de Colombia in 2005 and 2008, respectively, and his Ph.D. in electrical and computer engineering from University of Florida in 2012. Between 2004 and 2008, he was appointed as a research assistant at the Control and Digital Signal Processing Group (GCPDS) at Universidad Nacional de Colombia. During his Ph.D. studies he worked as a research assistant at the Computational Neuro-Engineering Laboratory (CNEL) at University of Florida. His main research interests are in machine learning and signal processing.