2003s-19

Spectral Clustering and Kernel PCA are Learning Eigenfunctions

Yoshua Bengio, Pascal Vincent,
Jean-François Paiement

Série Scientifique
Scientific Series

Montréal
Mai 2003

CIRANO
Centre interuniversitaire de recherche
en analyse des organisations

Les cahiers de la série scientifique (CS) visent à rendre accessibles des résultats de recherche effectuée au CIRANO afin de susciter échanges et commentaires. Ces cahiers sont écrits dans le style des publications scientifiques. Les idées et les opinions émises sont sous l'unique responsabilité des auteurs et ne représentent pas nécessairement les positions du CIRANO ou de ses partenaires.

This paper presents research carried out at CIRANO and aims at encouraging discussion and comment. The observations and viewpoints expressed are the sole responsibility of the authors. They do not necessarily represent positions of CIRANO or its partners.

ISSN 1198-8177

Spectral Clustering and Kernel PCA are Learning Eigenfunctions

Yoshua Bengio,[*] *Pascal Vincent,*[†] *Jean-François Paiement*[‡]

Résumé / *Abstract*

Dans cet article, on montre une équivalence directe entre la classification spectrale et l'ACP à noyau, et on montre que les deux sont des cas particuliers d'un problème plus général, celui d'apprendre les fonctions propres d'un noyau. Ces fonctions fournissent une base pour un espace de Hilbert dont le produit scalaire est défini par rapport à la densité des données. Les fonctions propres définissent une transformation de coordonnées naturelles pour de nouveaux points, alors que des méthodes comme la classification spectrale et les 'Laplacian eigenmaps' ne fournissaient un système de coordonnées que pour les exemples d'apprentissage. Cette analyse suggère aussi de nouvelles approches à l'apprentissage non-supervisé dans lesquelles on extrait des abstractions qui résument la densité des données, telles que des variétés et des classes naturelles.

Mots clés : Apprentissage non-supervisé, agrégation, apprentissage d'espace, fonctions propres, réduction non-linéaire de dimensionnalité.

In this paper, we show a direct equivalence between spectral clustering and kernel PCA, and how both are special cases of a more general learning problem, that of learning the principal eigenfunctions of a kernel, when the functions are from a Hilbert space whose inner product is defined with respect to a density model. This defines a natural mapping for new data points, for methods that only provided an embedding, such as spectral clustering and Laplacian eigenmaps. The analysis also suggests new approaches to unsupervised learning in which abstractions such as manifolds and clusters that represent the main features of the data density are extracted.

Keywords: *Unsupervised earning, clustering, manifold learning, eigenfunctions, non-linear dimensionality reduction.*

[*] CIRANO and Département d’informatique et recherche opérationnelle, Université de Montréal, Montréal, Québec, Canada, H3C 3J7, tel.: (514) 343-6804. Email: bengioy@iro.umontreal.ca.

[†] Département d'informatique et recherche opérationnelle, Université de Montréal, Québec, Canada, H3C 3J7. Email: vincentp@iro.umontreal.ca.

[‡] Département d'informatique et recherche opérationnelle, Université de Montréal, Québec, Canada, H3C 3J7. Email: paiemeje@iro.umontreal.ca.

1 Introduction

Clustering and manifold learning are intimately related: clusters and manifold both are zones of high density. Up to recently, both tasks have been treated quite separately with different unsupervised learning procedures, but recent work with kernel methods, as well as this paper, are changing that perspective.
Spectral clustering can give very impressive results and has attracted much interest in the last few years (Weiss, 1999; Ng, Jordan and Weiss, 2002). It is based on two main steps: first embedding the data points in a space in which clusters are more "obvious" (using the eigenvectors of a Gram matrix), a space in which the structure of the data is revealed, and then applying a classical clustering algorithm such as K-means, e.g. as in (Ng, Jordan and Weiss, 2002). What is very interesting is the way in which sets of points that are on different highly non-linear manifolds can get mapped (in the above first step) to almost linear subspaces (different for each of these manifolds), as shown below in Figure 1. The long-term goal of the research program to which this paper belongs is to better understand such mappings and take advantage of this understanding to open the door for new unsupervised learning procedures.

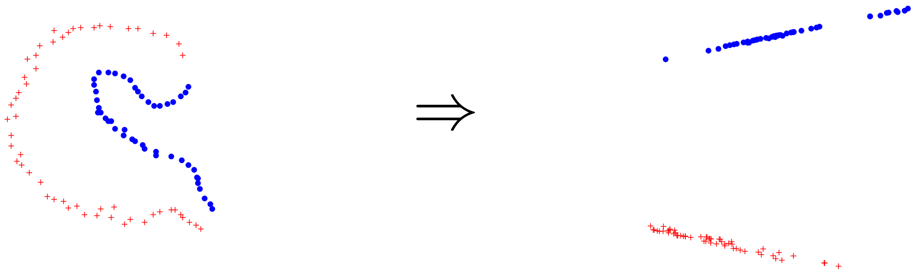

Fig. 1. Example of the transformation learned as part of spectral clustering. Input data on the left, transformed data on the right. Colors and cross/circle drawing are only used to show which points get mapped where: the mapping reveals both the clusters and the internal structure of the two manifolds.

One problem with spectral clustering is that the procedure is highly sensitive to the choice of the kernel, for example it is very sensitive to the choice of the spread (variance) of a Gaussian kernel. Another is that the procedure provides an embedding for the training points, not for new points. A very similar method for dimensionality reduction has been proposed in (Belkin and Niyogi, 2002a), based on so-called Laplacian eigenmaps. Belkin and Niyogi propose to use such transformations in a semi-supervised and transductive setting: the unlabeled test set and the input part of the training set are used to learn a mapping to a more revealing representation, and the transformed training set is used with a supervised learning algorithm.

Kernel PCA is another unsupervised learning method that was proposed earlier and that is based on the simple idea of performing Principal Components Analysis in the feature space of a kernel (Schölkopf and Müller, 1996). We will explain this approach in much more detail in the paper.

We show a direct equivalence between the embedding computed in spectral clustering and the mapping computed with kernel PCA, and how both are special cases of a more general learning problem, that of learning the principal eigenfunctions of a kernel, when the functions are from a Hilbert space whose inner product is defined with respect to a density model.

A consequence is that a natural mapping is defined, which can be applied to new points, for methods such as spectral clustering and Laplacian eigenmaps for which only an embedding of the training points was available.

2 Spectral Manifold Learning Methods

2.1 Kernels and Notation

The methods described here are based on kernels, i.e. symmetric two-argument functions. We also assume that the linear operator in $\mathcal{L}^2$ corresponding to the kernel is a compact operator, i.e. it maps functions in $\mathcal{L}^2$ to a closed and totally bounded set.

Often a kernel K is assumed to be semi-positive definite, and in that case it can be written as a dot product in a "feature space" $\phi(x)$ (see Mercer theorem and a review of learning algorithms based on kernels, e.g. (Schölkopf, Burges and Smola, 1999; Wahba, 1990)):

$$K(x,y) = \sum_i \phi_i(x)\phi_i(y) = \phi(x) \cdot \phi(y) \tag{1}$$

where both x and y are in $\mathbf{R}^d$, while $\phi(x) \in \mathbf{R}^r$, or to allow r not necessarily finite, we write $\phi(x) \in l_2$, the space of bounded sum of squares sequences.

We are given a data set $\{x_1, \ldots, x_n\}$ with examples $x_i \in \mathbf{R}^d$. We will associate a density $p(x)$ to the data generating process, either the empirical density or one obtained through a smoothing model. We will write $E[.]$ for expectations over that density, or to make it clear over which variable the integral is performed, we will write

$$E_x[f(x)] = \int f(x)p(x)dx.$$

For example, in the next two sections (on spectral clustering and on kernel PCA), we will restrict our attention to the empirical distribution associated with our data set, so we would have

$$E_x[f(x)] = \frac{1}{n}\sum_i f(x_i).$$

2.2 Spectral Clustering

Several variants of spectral clustering have been proposed (Weiss, 1999). They can yield impressively good results where traditional clustering looking for "round blobs" in the data, such as K-means, would fail miserably. Here we follow the treatment of (Ng, Jordan and Weiss, 2002) (see the figures in the same paper). The most commonly used kernel is the Gaussian kernel:

$$K(x, y) = e^{-||x-y||/\sigma^2}. \tag{2}$$

Note that the choice of σ can strongly influence the results, so this hyper-parameter has to be selected carefully.

Spectral clustering works by first embedding the data points in a space where clusters are more clearly revealed. An example of such embedding is shown in Figure 1. The embedding is obtained as follows. First form the symmetric semi-positive definite Gram matrix M with

$$M_{i,j} = K(x_i, x_j) \tag{3}$$

and then using the row sums

$$D_i = \sum_j M_{i,j}$$

normalize it as such:

$$\tilde{M}_{i,j} = \frac{M_{i,j}}{\sqrt{D_i D_j}}.$$

Note for comparison later in the paper that, equivalently, this *divisive normalization* of the Gram matrix corresponds (up to a constant) to defining a normalized kernel $\tilde{K}$ as follows:

$$\tilde{K}(x, y) = \frac{K(x, y)}{\sqrt{E_x[K(x, y)]E_y[K(x, y)]}} \tag{4}$$

(where the expectations are over the empirical distribution). Finally compute the $m < n$ principal eigenvectors of $\tilde{M}$, satisfying

$$\tilde{M}\alpha_k = \lambda_k \alpha_k.$$

Let A be the $m \times n$ matrix of these eigenvectors. The lower-dimensional embedding associates the point x_i in $\mathbf{R}^d$ to the i-th column of A, $A_i \in \mathbf{R}^m$:

$$A_i = (\alpha_{1i}, \alpha_{2i}, \ldots, \alpha_{mi}). \tag{5}$$

The coordinates of the examples within the eigenvectors represent an embedding that has very interesting properties (see Figure 1). Clustering is obtained from these coordinates. In the illustration of Figure 1, the two clusters correspond to groups of points that have an approximately **constant angle**, i.e. they are near one of two lines that start at the origin. Thus, in (Ng, Jordan and Weiss, 2002) it is proposed to first project these coordinates onto the unit sphere before performing K-means clustering. Projection onto the unit sphere maps A_i into $A_i/||A_i||$. See (Ng, Jordan and Weiss, 2002; Weiss, 1999) for further justification of this procedure and its relation to the graph Laplacian and the min-cut problem.

2.3 Kernel Principal Components Analysis

Kernel PCA is another unsupervised learning technique that maps data points to a new space. It generalizes the Principal Components Analysis approach to non-linear transformations using the kernel trick (Schölkopf and Müller, 1996; Schölkopf, Smola and Müller, 1998; Schölkopf, Burges and Smola, 1999). The algorithm implicitly finds the leading eigenvectors and eigenvalues of the covariance of the projection $\phi(x)$ of the data in "feature space" (see eq. 1):

$$C = E_x[(\phi(x) - E_x[\phi(x)])(\phi(x) - E_x[\phi(x)])'] = E_x[\tilde{\phi}(x)\tilde{\phi}(x)'] \tag{6}$$

where

$$\tilde{\phi}(x) \overset{\text{def}}{=} \phi(x) - E_x[\phi(x)].$$

Let us define the eigenvectors of the covariance matrix:

$$C v_k = \lambda_k v_k.$$

Using the notation of the previous section, the kernel PCA algorithm has the following steps.

- **Training:**
  1. Centering: the kernel K is first "normalized" into $\tilde{K}$ such that the corresponding feature space points $\tilde{\phi}(x_i)$ have zero expected value (under the data empirical distribution):

     $$\tilde{K}(x,y) = K(x,y) - E_x[K(x,y)] - E_y[K(x,y)] + E_x[E_y[K(x,y)]]. \tag{7}$$

     See derivation of this expression in the next subsection. A corresponding normalized Gram matrix $\tilde{M}$ is formed. Note this *additive normalization* is different from normalization 4.
  2. Eigen-decomposition: find the principal eigenvectors α_k and eigenvalues a_k of the Gram matrix $\tilde{M}$ ($\tilde{M}_{i,j} = \tilde{K}(x_i, x_j)$), i.e. solving

     $$\tilde{M}\alpha_k = a_k \alpha_k.$$

- **Test points projection:** to project a test point x on the k-th eigenvector v_k of the (properly centered) covariance matrix, compute

  $$\pi_k(x) = v_k \cdot \tilde{\phi}(x) = \sum_{i=1}^{n} \alpha_{ki} \tilde{K}(x_i, x). \tag{8}$$

  Note that $v_k = \sum_{i=1}^{n} \alpha_{ki} \tilde{\phi}(x_i)$, as shown in (Schölkopf, Smola and Müller, 1998).

2.4 Normalization of the Kernel by Centering

It can be shown that the above normalization of the kernel indeed yields to centering of $\tilde{\phi}(x)$, as follows. The normalized kernel

$$\tilde{K}(x,y) = \tilde{\phi}(x) \cdot \tilde{\phi}(y)$$

is expanded as follows

$$\begin{aligned}\tilde{K}(x,y) &= (\phi(x) - E_x[\phi(x)]) \cdot (\phi(y) - E_y[\phi(y)]) \\ &= K(x,y) - E_x[K(x,y)] - E_y[K(x,y)] + E_x[E_y[K(x,y)]] \qquad (9)\end{aligned}$$

2.5 Other Spectral Dimensionality Reduction Methods

Several other dimensionality reduction and manifold discovery methods rely on the solution of an eigen-decomposition problem. For example, Local Linear Embedding (Roweis and Saul, 2000) and Isomap (Tenenbaum, de Silva and Langford, 2000) try to discover a non-linear manifold, while Multidimensional Scaling (Cox and Cox, 1994) looks for a linear manifold (but starting from a matrix of similarities between pairs of points, whereas Principal Components Analysis starts from a set of points and the definition of a dot product). An interesting link between multidimensional scaling and kernel PCA is discussed in (Williams, 2001).

A non-linear manifold discovery method very close to the mapping procedure used in spectral clustering is that of Laplacian eigenmaps (Belkin and Niyogi, 2002a; Belkin and Niyogi, 2002b; He and Niyogi, 2002; Belkin and Niyogi, 2002c), which have been proposed to perform semi-supervised learning: the mapping is obtained through the eigen-decomposition of an affinity matrix, on the input part of both labeled and unlabeled data. The mapped inputs from the labeled data set can then be used to perform supervised learning from a representation that is hoped to be more meaningful, since only the types of variations of the data that are relevant to the input distribution would be represented in the transformed data.

Note also that it has already been proposed to use kernel PCA as a preprocessing step before doing clustering, in (Christianini, Shawe-Taylor and Kandola, 2002).

Note that very interesting links have already been established between kernel PCA and learning eigenfunctions in (Williams and Seeger, 2000). In particular, the eigenvalues and eigenvectors obtained from the eigen-decomposition of the Gram matrix converge to the eigenfunctions of the linear operator defined by the Kernel K with respect to the data density p, as in equation 12 below, as the number of data points increases.

3 Similarity Kernel Eigenfunctions

In this section we introduce the notion of **eigenfunctions of a kernel**, which we will find later to generalize both spectral clustering and kernel PCA. Eigenfunctions thus defined have already been introduced in (Williams and Seeger, 2000), where the convergence of the Gram matrix eigenvalues to the kernel operator eigenvalues is shown. In section 3.2, we discuss how one might learn the eigenfunctions when the reference density $p(x)$ below is *not necessarily the empirical density.*

3.1 Hilbert Space and Kernel Decomposition

Consider a Hilbert space $\mathcal{H}$, a set of real-valued functions in $\mathbf{R}^d$ accompanied by an inner product defined with a density $p(x)$:

$$\langle f, g \rangle \overset{\text{def}}{=} \int f(x)g(x)p(x)dx. \tag{10}$$

This also defines a norm over functions:

$$||f||^2 \overset{\text{def}}{=} \langle f, f \rangle.$$

As discussed in (Williams and Seeger, 2000), the eigenfunctions of the linear operator corresponding to a given semi-positive kernel function $K(x, y)$ are thus defined by the solutions of

$$K f_k = \lambda_k f_k \tag{11}$$

where $f \in \mathcal{H}$, $\lambda_k \in \mathbf{R}$, and we denote Kf the application of the linear operator K to the function f,

$$(Kf)(x) \overset{\text{def}}{=} \int K(x, y)f(y)p(y)dy. \tag{12}$$

We assume that K and p are such that K has a discrete spectrum (e.g. obtained with bounded values on a compact support). The kernel K can thus be seen as a linear operator, and expanded in terms of a basis formed by its eigenfunctions (Mercer):

$$K = \sum_k \lambda_k f_k f'_k$$

where by convention $|\lambda_1| \geq |\lambda_2| \geq \ldots$ This can also be written as follows:

$$K(x, y) = \sum_{k=1}^{\infty} \lambda_k f_k(x) f_k(y).$$

Because we choose the eigenfunctions to form an orthonormal basis, we have

$$\langle f_k, f_l \rangle = \delta_{k,l}.$$

Section 3.2 shows a criterion which can be minimized in order to learn the principal eigenfunctions.

(Williams and Seeger, 2000) have shown the following when $p(x)$ is the true data generating density and the unknown function f is estimated with an approximation g that is a finite linear combination of basis functions: if f is assumed to come from a zero-mean Gaussian process prior with covariance $E_f[f(x)f(y)] = K(x, y)$, then the best choices of basis functions, in terms of expected squared error, are (up to rotation/scaling) the leading eigenfunctions of the linear operator K as defined above.

3.2 Learning the Leading Eigenfunctions

Using the Fourier decomposition property, the best approximation of $K(x, y)$ w.r.t. $\mathcal{H}$'s norm using only m terms is the expansion that uses the *first* m terms (with largest eigenvalues):

$$\sum_{k=1}^{m} \lambda_k f_k(x) f_k(y) \approx K(x, y),$$

in the sense that it minimizes the $\mathcal{H}$-norm of the approximation error. In particular, let us consider the principal eigenfunction. It is the norm 1 function f which minimizes

$$J_K(f, \lambda) = \int (K(x, y) - \lambda f(x) f(y))^2 p(x) p(y) dx dy$$

i.e.

$$(f_1, \lambda_1) = \mathrm{argmin_{f,\lambda} J_K(f, \lambda)} \tag{13}$$

under the constraint $||f|| = 1$. This is only a generalization to functional spaces of the results already obtained for principal component analysis.

Proposition 1 *The principal eigenfunction of the linear operator (eq. 11) corresponding to kernel K is the norm-1 function f that minimizes the reconstruction error*

$$J_K(f, \lambda) = \int (K(x, y) - \lambda f(x) f(y))^2 p(x) p(y) dx dy.$$

The proof can be found in (Bengio, Vincent and Paiement, 2003).
Note that the proof also gives us a criterion in which the norm 1 constraint is eliminated:

$$J_K(g) = \int (K(x, y) - g(x) g(y))^2 p(x) p(y) dx dy \tag{14}$$

which gives a solution g from which we can recover λ and f through $\lambda = ||g||^2$ and $f = g/\sqrt{\lambda}$.
Note that the function g that we obtain is actually a component of a "feature space" ϕ for K. Indeed, if

$$K(x, y) = \sum_i \lambda_i f_i(x) f_i(y)$$

then writing $\phi_i(x) = \sqrt{\lambda_i} f_i(x)$ gives rise to a dot product decomposition of K,

$$K(x, y) = \phi(x) \cdot \phi(y).$$

Let us now consider learning not only the first but also the leading m eigenfunctions.

Proposition 2 *Given the principal $m - 1$ eigenfunctions f_i of the linear operator associated with a symmetric function $K(x, y)$ (eq. 11), the m-th one can be obtained by minimizing w.r.t. g the expected value of $(K(x, y) - g(x)g(y) - \sum_{i=1}^{k-1} \lambda_i f_i(x) f_i(y))^2$ over $p(x, y) = p(x)p(y)$. Then we get the m-th eigenvalue*

$\lambda_m = ||g||^2$ and the m-th eigenfunction $f_m = g/\sqrt{\lambda_m}$.

Proof

Approximate the kernel with $g(x)g(y) + \sum_{i=1}^{m-1} \lambda_i f_i(x) f_i(y)$:

$$J_m = \int (K(x,y) - g(x)g(y) - \sum_{i=1}^{m-1} \lambda_i f_i(x) f_i(y))^2 p(x)p(y)dxdy.$$

where $g(x)$ can be decomposed into $g(x) \overset{\text{def}}{=} \lambda' f'(x)$ with $||f'|| = 1$, and (f_i, λ_i) are the first $m-1$ (eigenfunction,eigenvalue) pairs in order of decreasing absolute value of λ_i. We want to prove that g that minimizes J_m is f_m.
The minimization of J_m with respect to λ' gives

$$\frac{\partial J_m}{\partial \lambda'} = 2\int (K(x,y) - \lambda' f'(x) f'(y) - \sum_{i=1}^{m-1} \lambda_i f_i(x) f_i(y)) f'(x) f'(y) p(x)p(y)dxdy = 0$$

which gives rise to

$$\lambda' = \langle f', Kf' \rangle - \sum_{i=1}^{m-1} \int \lambda_i f_i(x) f_i(y) f'(x) f'(y) p(x)p(y)dxdy \qquad (15)$$

We have

$$\begin{aligned} J_m = J_{m-1} - 2\textstyle\int \lambda' f'(x) f'(y)(K(x,y) - \sum_{i=1}^{m-1} \lambda_i f_i(x) f_i(y))p(x)p(y)dxdy \\ + \textstyle\int (\lambda' f'(x) f'(y))^2 p(x)p(y)dxdy \end{aligned}$$

which, using eq. 15, gives

$$J_m = J_{m-1} - \lambda'^2.$$

λ'^2 should be maximized for J_m to be minimized (giving rise to the ordering of the eigenfunctions). Take the derivative of J_m w.r.t. the value of f' at z (under regularity conditions to bring derivatives inside integrals):

$$\frac{\partial J_m}{\partial f'(z)} = 2\int (K(z,y) - \lambda' f'(z) f'(y) \sum_{i=1}^{m-1} \lambda_i f_i(z) f_i(y)) \lambda' f'(y) p(y)dy$$

and set it equal to zero:

$$\int K(z,y) f'(y) p(y)dy = \sum_{i=1}^{m-1} \int \lambda_i f_i(z) f_i(y) f'(y) p(y)dy.$$

Using the constraint $||f'||^2 = \langle f', f' \rangle = \int f'(y)^2 p(y)dy = 1$, we obtain:

$$Kf' = \lambda' f' + \sum_{i=1}^{m-1} \int \lambda_i f_i(z) f_i(y) f'(y) p(y)dy. \qquad (16)$$

Using the assumption that f_i are orthogonal for $i < m$, rewrite eq. 16 as

$$Kf' = \lambda' f' + \sum_{i=1}^{m-1} \lambda_i f_i \langle f', f_i \rangle.$$

Since we can write the application of K in terms of the eigenfunctions,

$$Kf' = \sum_{i=1}^{\infty} \lambda_i f_i \langle f', f_i \rangle,$$

we obtain

$$\lambda' f' = \lambda_m f_m \langle f', f_m \rangle + \sum_{i=m+1}^{\infty} \lambda_i f_i \langle f', f_i \rangle.$$

Applying Perceval's theorem to obtain the norm on both sides, we get

$$\lambda'^2 = {\lambda_m}^2 \langle f', f_m \rangle^2 + \sum_{i=m+1}^{\infty} {\lambda_i}^2 \langle f', f_i \rangle^2.$$

If the eigenvalues are distinct, we have $\lambda_m > \lambda_i$ for $i > m$, and the last expression is maximized when $\langle f', f_m \rangle = 1$ and $\langle f', f_i \rangle = 0$ for $i > m$, which proves that $f_m = f'_m$ is in fact the m-th eigenfunction of the kernel K and thereby $\lambda_m = \lambda'_m$.
If the eigenvalues are not distinct, then the result can be generalized in the sense that the choice of eigenfunctions is not anymore unique but the eigenfunctions sharing the same eigenvalue form an orthogonal basis for a subspace.
Then since we have assumed $g = \lambda' f'$, after obtaining g through the minimization of J_m, since this minimization yields $\lambda' = \lambda_m$ and $f' = f_m$, and since $||f_m|| = 1$ by definition, we get $\lambda_m = ||g||^2$ and $f_m = g/\sqrt{\lambda_m}$.
Q.E.D.

To simplify notation, let us define the "residual kernel"

$$K_k(x, y) = K(x, y) - \sum_{i=1}^{k} \lambda_k f_k(x) f_k(y), \tag{17}$$

with $K_0 = K$.
Justified by Proposition 2 above, a general algorithm for learning the first m eigenfunctions (and corresponding eigenvalues) of a linear operator K can thus be written as follows:

- For $k = 1$ to m
  $$(f_k, \lambda_k) = \underset{||f||=1}{\text{argmin}_{\text{f},\lambda}} J_{K_{k-1}}(f, \lambda).$$

In practice the minimization would have to be performed on a large class of functions or non-parametrically, i.e. we impose some restrictions on the class of functions. A special case of interest is that in which the density $p(x)$ is the empirical

density. In that case the minimization of J can be done with numerical analysis methods for finding the eigenvectors of a matrix. However, it might be interesting to consider smooth or otherwise constrained classes of functions (which can only approximate the above minimization).
Following the reasoning exposed for online learning of the principal components (Diamantras and Kung, 1996), a simpler implementation would not have to wait for the first $m-1$ eigenfunctions before beginning to learn the m-th one. They can all be learned in parallel, using the algorithm to learn the m-th one that assumes that the first $m-1$ are learned: convergence will simply be faster for the leading eigenfunctions. Note that convergence also depends on the ratios of eigenvalues, as usual for PCA and iterative eigen-decomposition algorithms (Diamantras and Kung, 1996).

4 Links between the Methods

In this section we show that finding the eigenfunctions of the kernel function includes as a special case both the embedding found in spectral clustering and that found by Kernel PCA.

Proposition 3 *If we choose for $p(x)$ (the weighing function in the Hilbert space inner product of eq. 10) the empirical distribution of the data, then the embedding A_{ik} obtained with spectral clustering (see eq. 5) is equivalent to values of the eigenfunctions: $A_{ik} = f_k(x_i)$ where f_k is the k-th principal eigenfunction of the kernel.*

Proof

As shown in Proposition 1, finding function f and scalar λ minimizing

$$\int (\tilde{K}(x,y) - \lambda f(x)f(y))^2 p(x)p(y)dxdy$$

such that $||f|| = 1$ yields a solution that satisfies the eigenfunction equation

$$\int \tilde{K}(x,y)f(y)p(y)dy = \lambda f(x)$$

with λ the (possibly repeated) maximum norm eigenvalue, i.e. we obtain $f = f_1$ and $\lambda = \lambda_1$ respectively the principal eigenfunction (or one of them if the maximum eigenvalue is repeated) and its corresponding eigenvalue.
Here $\tilde{K}$ refers to a possibly normalized kernel, e.g. such as may be defined in eq. 4 for spectral clustering.
Using the empirical density and considering the values of x at the data points x_i, the above equation becomes (for all x_i):

$$\frac{1}{n}\sum_j \tilde{K}(x_i, x_j)f(x_j) = \lambda f(x_i).$$

Let us write $u_j = f(x_j)$ and $\tilde{M}_{ij} = \tilde{K}(x_i, x_j)$, then the above can be written

$$\tilde{M}u = n\lambda u.$$

The spectral clustering method thus solves the same eigenvalue problem (up to scaling the eigenvalue by n) and we obtain for the principal eigenvector:

$$A_{i1} = f_1(x_i).$$

To obtain the result for the other coordinates, i.e. other eigenvalues, simply consider the "residual kernel" K_k as in eq. 17 and recursively apply the same reasoning to obtain that $A_{i2} = f_2(x_i)$, $A_{i3} = f_3(x_i)$, etc... Q.E.D.

Discussion

What do we learn from this proposition? Firstly, there is an equivalence between the principal eigenvectors of the Gram matrix and the principal eigenfunctions of a the kernel, when the Hilbert space is defined with an inner product of the form of eq. 10, and the density in the inner product is the empirical density. Why is this interesting? This suggests generalizations of the transformation performed for spectral clustering in which one uses a smoother density $p(x)$, e.g. obtained through a parametric or non-parametric model.

Proposition 4 *Let $\pi_k(x)$ be the test point projection (eq. 8) on the k-th principal component obtained by kernel PCA with normalized kernel $\tilde{K}(x, y)$. Then*

$$\pi_k(x) = \lambda_k f_k(x)$$

where λ_k and $f_k(x)$ are respectively the k-th leading eigenvalue and eigenfunction of $\tilde{K}$, and the Hilbert space inner product weighing function $p(x)$ is the empirical density.

Proof

Let us start from the eigenfunction equation 11 on kernel $\tilde{K}$ and apply the linear operator $\tilde{K}$ on both sides:

$$\tilde{K}\tilde{K}f_k = \lambda_k \tilde{K} f_k.$$

which can be written

$$\int \tilde{K}(x, y) \int \tilde{K}(y, z) f_k(z) p(z) p(y) dz dy = \lambda_k \int \tilde{K}(x, y) f_k(y) p(y) dy$$

or changing the order of integrals on the left-hand side:

$$\int f_k(z) \left(\int \tilde{K}(x, y) \tilde{K}(y, z) p(y) dy \right) p(z) dz = \lambda_k \int \tilde{K}(x, y) f_k(y) p(y) dy$$

Let us now plug-in the definition of $\tilde{K}(x, y) = \sum_i \tilde{\phi}_i(x)\tilde{\phi}_i(y)$:

$$\int f_k(z) \left(\int \sum_i \tilde{\phi}_i(x)\tilde{\phi}_i(y) \sum_j \tilde{\phi}_j(y)\tilde{\phi}_j(z) p(y) dy \right) p(z) dz = \lambda_k \int \sum_i \tilde{\phi}_i(x)\tilde{\phi}_i(y) f_k(y) p(y) dy.$$

In this expression we can see the element (i,j) of the feature space covariance matrix C (eq. 6):

$$C_{ij} = \int \tilde{\phi}_i(y)\tilde{\phi}_j(y)p(y)dy$$

and we obtain (plugging this definition on the left hand side and pulling sums out of integrals)

$$\sum_i \tilde{\phi}_i(x) \sum_j C_{ij} \int \tilde{\phi}_j(z) f_k(z) p(z) dz = \lambda_k \sum_i \tilde{\phi}_i(x) \int f_k(y)\tilde{\phi}_i(y)p(y)dy$$

or

$$\tilde{\phi}(x) \cdot (C\langle f_k, \tilde{\phi}\rangle) = \tilde{\phi}(x) \cdot (\lambda_k \langle f_k, \tilde{\phi}\rangle)$$

where $\langle f_k, \tilde{\phi}\rangle$ is the feature space vector with elements $\int f_k(y)\tilde{\phi}_i(y)p(y)dy$. Since this is true for all x, it must be that in the region where $\tilde{\phi}(x)$ takes its values,

$$Cv_k = \lambda_k v_k$$

where $v_k = \langle f_k, \tilde{\phi}\rangle$ and it is also the k-th eigenvector of the covariance matrix C. Finally, the kernel PCA test projection on that eigenvector is

$$\begin{aligned} \pi_k(x) &= v_k \cdot \tilde{\phi}(x) \\ &= (\int f_k(y)\tilde{\phi}(y)p(y)dy) \cdot \tilde{\phi}(x) \\ &= \int f_k(y)\tilde{\phi}(y) \cdot \tilde{\phi}(x)p(y)dy \\ &= \int f_k(y)K(x,y)p(y)dy \\ &= \lambda_k f_k(x) \qquad (18) \end{aligned}$$

Q.E.D.

Discussion

What do we learn from this second proposition? We find an equivalence between the eigenfunctions of the kernel (in an appropriate Hilbert space) and the mapping computed through kernel PCA. By combining this with the first proposition, we trivially obtain an equivalence between the mappings computed for spectral clustering and for kernel PCA, up to *scaling by the eigenvalues* and to a *different normalization of the kernel.*

A nice fallout of this analysis is that it provides for methods such as spectral clustering and Laplacian eigenmaps a simple way to **generalize the embedding to a mapping**: whereas these methods only give the transformed coordinates of training points (i.e. an embedding of the training points), it is easy to obtain the transformed coordinates of a new point, once it is realized that the transformed coordinates are simply the values of the principal eigenfunctions. Let us first consider the easiest case, where $p(x)$ is the empirical distribution. Then Proposition 4 allows us to write

$$f_k(x) = \sum_i \alpha_{ki}\tilde{K}(x_i, x)$$

where α_k is the k-th principal eigenvector of the normalized Gram matrix $\tilde{M}$, with $\tilde{M}_{ij} = \tilde{K}(x_i, x_j)$. When $p(x)$ is not the empirical distribution, Propositions 1 and 2 provide a criterion that can be minimized in order to learn the principal eigenfunctions f_k.

In addition, we again find the possibility of generalizing from kernel PCA to the case when the density defining the inner product of the Hilbert space is not the empirical density but a smoother density. Finally, by stating the problem in terms of eigenfunctions, we open the door to other generalizations which future work will investigate, in which the eigenfunctions are only approximately estimated (allowing to impose further smoothness constraints or other domain-specific constraints), and the possibility of estimating the eigenfunctions through a stochastic minimization process that does not require to explicitly compute and store the Gram matrix.

The links thus discovered leave open other questions. For example, is there a "geometric" meaning to the divisive normalization of the kernel used with spectral clustering (equation 4)? This normalization comes out of the justification of spectral clustering as a relaxed statement of the min-cut problem (Chung, 1997; Spielman and Teng, 1996) (to divide the examples into two groups such as to minimize the sum of the "similarities" between pairs of points straddling the two groups). The additive normalization performed with kernel PCA (equation 7) makes sense geometrically as a centering in feature space. Both normalization procedures make use of the kernel row/column average $E_x[K(x, y)]$.

5 Conclusion

Spectral methods discussed in this paper provide a data-dependent mapping that can be applied not only to training points but also to new points. They empirically appear to allow capturing such salient features of a data set as its main clusters and submanifolds. This is unlike previous manifold learning methods like LLE and Isomap, which compute an embedding for the training points and which assume a single manifold (but may work with more).

However, there is much that remains to be understood about these methods. For example, what is the role of normalization? How should the kernel be chosen? More fundamentally, why are these algorithms doing what they are doing? To this question there are already partial answers, and this paper may have contributed a little bit to this understanding, but the picture is far from clear.

Future work will investigate specific algorithms for performing the minimization of the quadratic criterion of eq. 14. In particular, importance sampling and stochastic gradient descent could be used to iteratively minimize it, even in an on-line setting or where it is not feasible to store a Gram matrix or compute the leading eigenvectors of a huge (even if sparse) matrix.

Finally, a better understanding of these methods opens the door to new and potentially much more powerful unsupervised learning algorithms. Several directions remain to be explored:

1. Using a smoother distribution than the empirical distribution to define the inner product. But why, fundamentally, might this be helpful?

2. Learning a density function from the mapping in order to compute likelihoods. This could be achieved by learning a reconstruction from the eigenfunction space back to the original space, and using a noise model to obtain (through a convolution) the mathematical form of the input density.
3. Learning higher-level abstractions on top of lower-level abstractions by iterating the unsupervised learning process in multiple "layers". Preliminary experiments on toy data suggests that this idea works, but why should it work? that remains to be shown.
4. Using the data to define the kernel. Another paper is in preparation that attempts to answer that question.

Acknowledgments The authors would like to thank Yann Le Cun and Yves Grandvalet for helpful discussions, and the following funding organizations: NSERC, MITACS, IRIS, and the Canada Research Chairs.

References

Belkin, M. and Niyogi, P. (2002a). Laplacian eigenmaps and spectral techniques for embedding and clustering. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press. 1, 5

Belkin, M. and Niyogi, P. (2002b). Laplacian eigenmaps for dimensionality reduction and data representation. Technical Report TR-2002-01, University of Chicago, Computer Science. 5

Belkin, M. and Niyogi, P. (2002c). Semi-supervised learning on manifolds. Technical Report TR-2002-12, University of Chicago, Computer Science. 5

Bengio, Y., Vincent, P., and Paiement, J. (2003). Learning eigenfunctions of similarity: Linking spectral clustering and kernel pca. Technical Report 1232, Département d'informatique et recherche opérationnelle, Université de Montréal. 7

Christianini, N., Shawe-Taylor, J., and Kandola, J. (2002). Spectral kernel methods for clustering. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems*, volume 14. The MIT Press. 5

Chung, F. (1997). Spectral graph theory. In *CBMS Regional Conference Series*, volume 92. American Mathematical Society. 13

Cox, T. and Cox, M. (1994). *Multidimensional Scaling*. Chapman & Hall, London. 5

Diamantras, K. and Kung, S. (1996). *Principal Components Neural Networks: theory and applications*. Wiley. 10

He, X. and Niyogi, P. (2002). Locality preserving projections (lpp). Technical Report TR-2002-09, University of Chicago, Computer Science. 5

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press. 1, 3

Roweis, S. and Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326. 5

Schölkopf, B., A. S. and Müller, K.-R. (1996). Nonlinear component analysis as a kernel eigenvalue problem. Technical Report 44, Max Planck Institute for Biological Cybernetics, Tübingen, Germany. 2, 4

Schölkopf, B., Burges, C. J. C., and Smola, A. J. (1999). *Advances in Kernel Methods — Support Vector Learning*. MIT Press, Cambridge, MA. 2, 4

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319. 4

Spielman, D. and Teng, S. (1996). Spectral partitionning works: planar graphs and finite element meshes. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*. 13

Tenenbaum, J., de Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323. 5

Wahba, G. (1990). Spline models for observational data. In *CBMS-NSF Regional Conference Series in Applied Mathematics*, volume 59, Philadelphia, PA. Society for Industrial and Applied Mathematics (SIAM). 2

Weiss, Y. (1999). Segmentation using eigenvectors: a unifying view. In *Proceedings IEEE International Conference on Computer Vision*, pages 975–982. 1, 3

Williams, C. (2001). On a connection between kernel pca and metric multidimensional scaling. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13, pages 675–681. MIT Press. 5

Williams, C. and Seeger, M. (2000). The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan Kaufmann. 5, 6