The Kernel Semi-least Squares Method for Sparse Distance Approximation

Samuel Epstein and Margrit Betke

{samepst, betke}@cs.bu.edu

Department of Computer Science

Boston University

Keywords: Kernel methods, distance approximation, semi-least squares

Abstract

We extend the Semi-least Squares problem defined by Rao and Mitra (1971) to the Kernel Semi-least Squares problem. We introduce "subset projection," a technique that produces a solution to this problem. We show how the results of such a subset projection can be used to approximate a computationally expensive distance metric.

1 Introduction

We consider the problem of approximating a computationally expensive distance of a real-time observation to a set of training data. Given is a set of n samples $Q =$

$\{q_1, q_2, \ldots, q_n\}$, from a sample space $\mathcal{Q}$, with $Q \subset \mathcal{Q}$, representing training data and a new sample q, representing a real-time observation. Also given is the target distance metric d which takes $\Theta(e)$ time to compute. The goal is to approximate the distance $d(q, q_i)$ of the real-time observation q to each $q_i \in Q$ in $o(en)$ time.

Our posed problem of sparse distance approximation represents a variant in the general literature of distance metric learning (Yang and Jin (2006)). The general problem of distance metric learning is common in real-world applications such as computer vision and content retrieval.

We present a general solution to the sparse distance approximation problem, relying only on the assumption that the target distance metric d is Hilbertian. We review the Semi-least Squares problem, introduced by Rao and Mitra (1971), in Section 2. We review kernel literature in Section 3. The contribution of this paper is to show the connection between these two ideas. By extending the Semi-least Squares problem to the Kernel Semi-least Squares problem, we provide a computationally advantageous method to solving the important problem of distance approximation (Sections 5 and 6). In Section 7, we provide a method for choosing the best training subset for distance approximation. We also provide an alternate derivation of the solution (Section 8). Related works are discussed in Section 9. Experimental results of our method and a comparison to the Nyström method (Williams and Seeger (2001)) are shown in Section 10.

2 The Semi-least Squares Problem

In this section, we describe the Semi-least Squares problem introduced by Rao and Mitra (1971). In this problem, the traditional L2-norm $\|\cdot\|$, used by the well-known Least Squares problem, is replaced by a seminorm. A seminorm, $\rho(\cdot)$, differs from a norm in that it is permitted that $\rho(u) = 0$ for some non-zero vectors u.

A square matrix J is positive semidefinite if it allows a decomposition[1] $J = HH^*$

[1] The * symbol represents the transpose operation.

, for some matrix H. We define seminorm $\|z\|_J = (z^* J z)^{1/2}$, where $J = HH^*$ is a positive semidefinite matrix. It is not a proper norm because for all vectors v in the null space of H, $\|v\|_J = 0$. A matrix G is said to be a *Semi-least Squares* inverse of a matrix A if the minimum of

$$\|Az - y\|_J \tag{1}$$

is attained at $z = Gy$, for any y. Such a G exists, being of the form

$$G \;=\; (A^* J A)^{-1} A^* J + P. \tag{2}$$

The matrix P is any projection onto the null space of $H^* A$. The precise form of P is $[I - (A^* J A)^+ (A^* J A)]\, U$, for any matrix U. The $(\cdot)^+$ operation represents the Moore-Penrose pseudoinverse.

Computation efficiency was not discussed in Rao and Mitra's 1971 paper. However computational benefits emerge when the Semi-least Squares problem is extended with kernels, as shown in Section 6.

3 Kernels

We assume the distance function d is Hilbertian. A distance metric d is Hilbertian if it can be embedded into a vector space, with finite or infinite number of dimensions. We can use the inner product function (or kernel function) associated with this space to perform useful functions, such as projections. We define such a kernel function k by

$$k(q, q') \;=\; g(q) - \frac{1}{2} d^2(q, q') + g(q'), \tag{3}$$

for any function $g : \mathcal{Q} \to \mathbb{R}^+$. One common form is $g(q) = \frac{1}{2} d^2(q, q')$ for some $q' \in \mathcal{Q}$. Since d is Hilbertian, k is semi-positive definite. This implies the kernel

function k represents the inner product over Q, mapped to a Hilbert space $\mathcal{F} = \mathbb{R}^l$, with $l \in \mathbb{N} \cup \{\infty\}$. This mapping from $\mathcal{Q}$ to $\mathcal{F}$ is represented by the function $\phi(q) : \mathcal{Q} \to \mathcal{F}$, with

$$k(q, q') \quad = \quad \phi^*(q)\phi(q'). \tag{4}$$

The kernel substitution method, known also as the kernel trick, allows the mapping ϕ to be implicitly defined by kernel k (Schölkopf and Smola (2001)).

We introduce some standard definitions associated with kernels. The Gram matrix K and the design matrix Φ are defined with respect to the set $Q = \{q_1, q_2, \ldots, q_n\}$, and a mapping ϕ. The Gram matrix K contains all the inner products between mappings of elements of Q. It is an $n \times n$ matrix, whose (i, j)th element is $k(q_i, q_j)$. The Gram matrix is an explicit construct. The design matrix Φ is a listing of the mapping of the elements of Q using ϕ. The design matrix Φ is an $l \times n$ matrix whose ith column is $\phi(q_i)$.

The Gram matrix and the design matrix are related by $\Phi^*\Phi = K$. The empirical map is denoted by $\mathbf{k}_Q : \mathcal{Q} \to \mathbb{R}^n$, where the ith value of vector $\mathbf{k}_Q(q)$ is $k(q_i, q)$, with $q_i \in Q$. It follows that $\Phi^*\phi(q) = \mathbf{k}_Q(q)$. The Gram matrix K and empirical map $\mathbf{k}_Q(\cdot)$ can be computed, whereas the design matrix Φ and the mapping $\phi(\cdot)$ are not generally computable.

4 Distance Decomposition

In this section we show how to compute the distance $d(q, q_i)$, with $q_i \in Q$, using tangent and orthogonal components with respect to the linear span of Q. This decomposition of the distance d into orthogonal components is a convenient form to be used in Section 5 for sparse distance approximation. Let k be a kernel function whose induced distance is d, with $d^2(q, q') = k(q, q) + k(q', q') - 2k(q, q')$. The kernel k defines a kernel feature

space $\mathcal{F}$ whose distances are congruent with d, and an implicit mapping $\phi : \mathcal{Q} \to \mathcal{F}$. The linear span of the set Q in $\mathcal{F}$ is defined by $\{\sum \alpha_i \phi(q_i) \mid \alpha \in \mathbb{R}^n\}$.

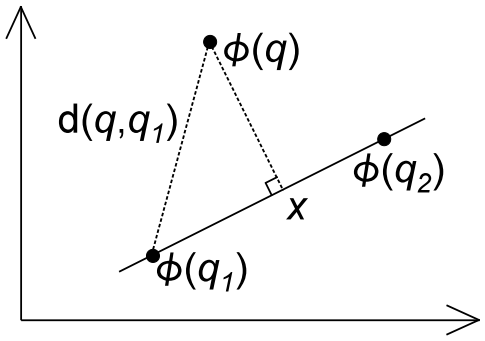

Figure 1: The point x represents the projection of the observation $\phi(q)$ onto the linear span of the set Q in kernel space $\mathcal{F}$. The vector from q to q_1 can be seen as the decomposition into two orthogonal components, the vector from $\phi(q)$ to x and the vector of x to $\phi(q_1)$. The linear span intersects the origin of $\mathcal{F}$.

Let x represent the results of a projection from the mapped observation $\phi(q)$ to the span of Q in $\mathcal{F}$, as seen in Figure 1. Since x is the result of an orthogonal projection, the distance function $d(q, q_i)$ for each $q_i \in Q$ can be defined by orthogonal components, with

$$d(q, q')^2 \quad = \quad \|\phi(q) - x\|^2 + \|x - \phi(q')\|^2. \tag{5}$$

In the rest of this section, we show how to compute the two terms of Equation 5.

Kernel Projection

The point x can be defined according to the L2 norm, with

$$x = \arg \min_{x' \in \text{Span}(Q) \text{ in } \mathcal{F}} \|x' - \phi(q)\|. \tag{6}$$

In general, the point x cannot be explicitly computed, since $\mathcal{F}$ has potentially infinite dimension. However point x can be represented as the linear combination of the observations of training set Q, mapped into Φ,

$$x \;=\; \sum_{i=1}^{n} \beta_i \phi(q_i) = \Phi\,\beta, \tag{7}$$

where β_i is the ith element of the vector $\beta \in \mathbb{R}^n$ that represents the coordinates of x using $\phi(q_i)$ as a basis. The projection used to produce point x can be formally defined using the Least Squares methodology and the coordinates β. A mapping $h(q) = \beta$ is defined to be the *Kernel Least Squares* inverse of a training set Q if

$$h(q) \;=\; \arg\min_{\beta} \sum_{q \in Q} \|\Phi\beta - \phi(q)\|.$$

Such a mapping exists and can be derived using Least Squares methodology, with

$$h(q) = \beta = \Phi^{+}\phi(q) = \left(\Phi^{*}\Phi\right)^{-1}\Phi^{*}\phi(q) = K^{-1}\mathbf{k}_Q(q), \tag{8}$$

where the term Φ^{+} represents the pseudoinverse of the design matrix (A similar form of Equation 8 was described by Schölkopf et al. (1999)). This mapping is known as the *kernel projection*. The Gram matrix K might be singular, but the limit $\lim_{\delta\to 0}(K + \delta I)^{-1}\mathbf{k}_Q(q)$ is guaranteed to exist by the definition of pseudoinverses.

Computation of Distance Decomposition

The coordinates β representing point x can be used to compute both distances of Equation 5. The term $\|\phi(q) - x\|^2$ can be computed using the fact that the span of Q in $\mathcal{F}$

contains the origin, and x is a orthogonal projection, with

$$\begin{aligned} \|\phi(q) - x\|^2 &= \|\phi(q)\|^2 - \|x\|^2 = \phi(q)^*\phi(q) - \beta^*\Phi^*\Phi\beta \\ &= k(q,q) - \beta^* K \beta. \end{aligned} \tag{9}$$

The term $\|x - \phi(q_i)\|^2$ can be computed by a direct substitution, with

$$\begin{aligned} \|x - \phi(q_i)\|^2 &= \|\Phi\,\beta - \phi(q_i)\|^2 \\ &= \beta^* K \beta - 2\beta^* K_i + K_{i,i}. \end{aligned} \tag{10}$$

The term K_i represents the ith column of the Gram matrix K of Q. Equations 9 and 10 can be substituted back into the distance decomposition of Equation 5, resulting in Equation 12. Given a kernel function k, a set Q with associated Gram matrix K, the distance $d(q, q_i)$ can be computed by the following steps:

1. The coordinates β are computed with the kernel projection,

$$\beta = K^{-1}\mathbf{k}_Q(q). \tag{11}$$

2. The distance $d(q, q_i)$ is computed using the coordinates, β,

$$d(q, q_i) = \left(k(q,q) - 2\beta^* K_i + K_{i,i}\right)^{1/2}. \tag{12}$$

5 Sparse Distance Approximation

Assuming a kernel of the form of Equation 3 is used, the time complexity of computing the kernel, $\Omega(e)$, is not less than the time complexity of computing the distance d. To compute the kernel projection of Equation 11, $k(q, q_i)$ needs to be computed for each $q_i \in Q$. The time complexity of this operation is on the order of $\Omega(en)$, where n is the

size of the training set and assuming $n < e$. Therefore computing the decomposition of the distance into orthogonal components in Section 4 does not provide any time complexity benefits. However, the kernel projection of Equation 11 can be approximated, which, as we show below, is computationally advantageous.

We introduce the technique of *subset projection*, which approximates the kernel projection using a secondary set of observations $R = \{r_1, r_2, \ldots, r_m\}$, where $|R| = m$, $|Q| = n$, and typically $R \subset Q$ and $m \ll n$. It is of the form

$$\hat{\beta} \quad = \quad (K_{RQ}^{+} + W)\mathbf{k}_R(q). \tag{13}$$

The term K_{RQ} is an $m \times n$ matrix whose value at position (i, j) is equal to $k(r_i, q_j)$. This matrix can be informally thought of as a "cross" Gram matrix between R and Q. The $m \times n$ matrix W represents any projection onto the null space of K_{RQ}. The term $\mathbf{k}_R$ represents the empirical function for R.

Whereas the kernel projection minimizes an L2 norm and provides a Kernel Least Squares solution, the subset projection of Equation 13 minimizes a seminorm and provides a *Kernel Semi-least Squares* solution. The Kernel Semi-least Squares problem is defined in Section 6.

Assuming the inverse cross Gram matrix is computed offline, the computational complexity of the subset projection is $\Theta(em)$, where $n = |Q|$ and $m = |R|$, and assuming $m < n < e$. The subset projection is more efficient to compute than the kernel projection. The results of the subset projection can be used to approximate the distance of q to each $q_i \in Q$.

Solution to Distance Approximation

Given is a kernel function k (derived from the distance d), a training set Q, and an observation q. A subset $R \subseteq Q$ is chosen, and the inverse cross matrix K_{RQ}^{+} and the projection W are pre-computed (the standard value of W is 0). The distance approxi-

mation, $\hat{d}$, is computed by two steps.

1. The approximate coordinates $\hat{\beta}$ are computed by the subset projection, with

$$\hat{\beta} \quad = \quad (K_{RQ}^{+} + W)\mathbf{k}_R(q). \tag{14}$$

2. The distance $\hat{d}$ is computed using the coordinates,

$$\hat{d}(q, q_i) \quad = \quad \Big(k(q,q) - 2\hat{\beta}^* K_i + K_{i,i}\Big)^{1/2}. \tag{15}$$

The approximated coordinates $\hat{\beta}$ can be reused for each $q_i \in Q$. The time complexity of the procedure is $O(em)$, assuming $m < n < e$. This implies the time complexity is $o(en)$ and thus this procedure represents a solution to the sparse distance approximation problem described in the introduction.

6 The Kernel Semi-least Squares Problem

We extend the Semi-least Squares problem of Section 2 by introducing the *Kernel Semi-least Squares* problem, for which the subset projection provides an optimal solution. Let k be a kernel, with associated mapping ϕ. Let $Q = \{q_1, q_2, \ldots, q_n\}$ be the training set and $R = \{r_1, r_2, \ldots, r_m\}$ be another set, where typically $m \ll n$ and $R \subset Q$. Sets Q and R have design matrices Φ and Φ_R, whose ith column is $\phi(q_i)$ and $\phi(r_i)$ respectively.

We define the matrix J used in the seminorm $\|\cdot\|_J$ to be the scatter matrix of R, $J = \Phi_R \Phi_R^*$. A mapping h is said to be the *Kernel Semi-least Squares* inverse of Q and R if the minimum of

$$\|\Phi\beta - \phi(q)\|_J \tag{16}$$

is achieved at

$$\hat{\beta} \;=\; h(q), \tag{17}$$

for all q. From Equation 2, such a mapping h exists and is of the form

$$h(q) \;=\; G\phi(q) \tag{18}$$

with

$$\begin{aligned} G &= (\Phi^* J \Phi)^{-1} \Phi^* J + P \\ &= (\Phi^* \Phi_R \Phi_R^* \Phi)^{-1} \Phi^* \Phi_R \Phi_R^* + P \\ &= (K_{RQ}^* K_{RQ})^{-1} K_{RQ}^* \Phi_R^* + P \\ &= K_{RQ}^+ \Phi_R^* + P. \end{aligned} \tag{19}$$

The term $K_{RQ} = \Phi_R^* \Phi$ is the "cross" Gram matrix. The matrix P represents any projection onto the null space of K_{RQ}. By restricting the projection P to be of the form $W\Phi_R^*$, with W being an $m \times n$ matrix that is a projection onto the null space of K_{RQ}, we can derive a computable mapping, with

$$\begin{aligned} \hat{\beta} &= h(q) \\ \hat{\beta} &= ((K_{RQ}^+)\Phi_R^* + W\Phi_R^*)\phi(q) \\ &= ((K_{RQ}^+ + W)\Phi_R^*)\phi(q) \\ &= (K_{RQ}^+ + W)\mathbf{k}_R(q), \end{aligned} \tag{20}$$

where $\mathbf{k}_R$ representing the empirical map with respect to R. The matrix W is of the form $\left[I - K_{RQ}^+ K_{RQ}\right] U$, for any $m \times n$ matrix U. This is the same form as the subset projection introduced in Equation 13. Thus the subset projection produces a solution

to the Kernel Semi-least Squares problem. Assuming the matrices K_{RQ}^{+} and W are precomputed, the time complexity to compute $\hat{\beta}$ is $\Theta(nm + em) = \Theta(em)$, assuming $m < n < e$.

7 Subset Selection

One important open issue is how to select the subset R_o for which the corresponding approximate distance $\hat{d}_R$ is closest to the original distance d. We formalize the selection of the subset as a minimization problem, where R_o represents the optimal subset:

$$R_o \;=\; \arg\min_{R\in\mathcal{R}} \sum_{q_i\in Q} \hat{d}_R(q_i, q_i)^2. \tag{21}$$

The set of candidate subsets is given by $\mathcal{R} \subseteq 2^Q$. The measure of closeness is determined by the sum of the squared approximate distances $\hat{d}_R$ of each training element $q_i \in Q$ to itself. The optimally selected subset can be rewritten as

$$R_o \;=\; \arg\max_{R\in\mathcal{R}} \sum_{i=1}^{n} \left(\hat{\beta}_i^* K_i\right), \tag{22}$$

with K_i being the ith column of the Gram matrix K of Q, and $\hat{\beta}_i = K_{RQ}^{+}\mathbf{k}_R(q_i)$. This can be further simplified with

$$R_o \;=\; \arg\max_{R\in\mathcal{R},} \sum_{i=1}^{n} \left(K_{RQ}^{+} K_{RQ} K\right)_{ii}. \tag{23}$$

The orthogonal projection on the range of K_{RQ}^{*} is denoted by $P_{RQ} = K_{RQ}^{+}K_{RQ}$, so the final expression to compute the optimally-selected subset R_o of candidates $\mathcal{R} \subseteq 2^Q$ is

$$R_o \;=\; \arg\max_{R\in\mathcal{R}} \;\mathrm{Tr}\left(P_{RQ}K\right). \tag{24}$$

This means that the optimal subset $R_o \in \mathcal{R}$ maximizes the sum of the eigenvalues of the Gram matrix K of the training set Q, projected onto the range of $K^*_{R_oQ}$. A general way to recover R_o depends on the choice of the candidate subsets $\mathcal{R}$. There is future work in determining an efficient algorithm to recover R_o for different choices of the candidates. One natural set of candidates is all subsets of a certain size m, with $\mathcal{R} = \{R : R \in 2^Q, |R| = m\}$.

Minimizing the difference of distances

Another approach is to find the subset $R_o \in \mathcal{R}$ that minimizes the absolute difference of the squared distances $\hat{d}$ and d, sampled over the training set Q,

$$R_o = \arg\min_{R\in\mathcal{R}} \sum_{q_i,q_j\in Q} \left| \hat{d}_R(q_i,q_j)^2 - d(q_i,q_j)^2 \right|. \tag{25}$$

This formulation can be reduced further with

$$R_o = \arg\min_{R\in\mathcal{R}} \sum_{i=1}^{n}\sum_{j=1}^{n} \left| (\hat{\beta}_i - \beta_i)^* K_j \right|, \tag{26}$$

with K_j being the jth column of the Gram matrix K of Q, and $\hat{\beta}_i = K^+_{RQ}\mathbf{k}_R(q_i)$, and $\beta_i = K^{-1}\mathbf{k}_Q(q_i)$. The expression can be further converted into a term similar to Equation 24, with

$$\begin{aligned} R_o &= \arg\min_{R\in\mathcal{R}}, \sum_{i=1}^{n}\sum_{j=1}^{n} \left| (I - K^+_{RQ}K_{RQ}) * K \right|_{ij}, \\ &= \arg\min_{R\in\mathcal{R}} \left\| \|N_{RQ}K\| \right\|_1. \end{aligned} \tag{27}$$

The orthogonal projection on the null space of K_{RQ} is represented by N_{RQ}. For this formulation, the best subset R_o minimizes the entrywise 1-norm $\|A\|_1 = \sum_{i,j} \left|a_{ij}\right|$ of the Gram matrix K of the training set Q, projected onto the nullspace of the K_{R_oQ}.

8 Alternate Derivation of the Subset Projection

The subset projection can also be derived as the concatenation of two sequential projections (Figure 2). However this derivation is more restrictive than that of Section 6, since it produces only one form of the subset projection with the matrix W set to zero. The first projection is from the position $\phi(q)$ to a point α in the linear span of R in feature space $\mathcal{F}$ with

$$\alpha \quad = \quad K_R^{-1}\mathbf{k}_R(q). \tag{28}$$

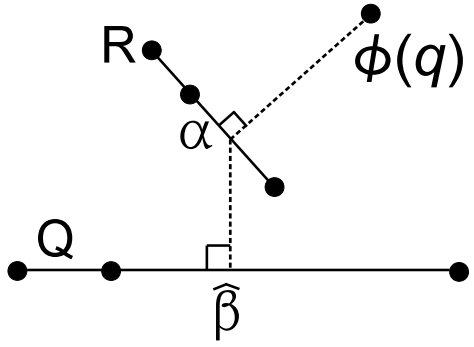

Figure 2: The subset projection can be interpreted as two projections. The first projection is from $\phi(q)$ to α on the span of R. The second projection is from α to $\hat{\beta}$ on the span of Q.

The second projection is from point α to the position $\hat{\beta}$ in span of Q in $\mathcal{F}$. Thus the second projection finds a position $\hat{\beta}$ that minimizes the term $\|\Phi\hat{\beta} - \Phi_R\alpha\|$. The solution to this term is of the form

$$\hat{\beta} \quad = \quad \left(K_Q^{-1}K_{QR}\right)\alpha. \tag{29}$$

Equation 29 appears in the solution of the *Reduced Subset Problem*, described by Schölkopf et al. (1999), which is to find a small set of examples Q and coefficients β to represent a point α in span of a larger set R. However, we use Equation 29 to project

onto a larger set. Whereas the *Reduced Subset Problem* assumes that $|Q| \ll |R|$ and $Q \subset R$, we assume $|Q| \gg |R|$ and $R \subset Q$. Concatenating the two projections together results in

$$\hat{\beta} \;=\; K_Q^{-1} K_{QR} K_R^{-1} \mathbf{k}_R(q) = K_{RQ}^{+} \mathbf{k}_R(q). \tag{30}$$

Equation 30 has the same form as the subset projection of Equation 13 with $W = 0$, and thus can be computed in the same fashion.

If the approximate distance $\hat{d}$ uses a subset where $R \subseteq Q$ and a subset projection with $W = 0$, then for any $q \in \mathcal{Q}$ and $q_i \in R$, $\hat{d}(q, q_i) = d(q, q_i)$. Although the approximation $\hat{d}$ of the distance is accurate, the value $\hat{d}(q, q_i)$ can be quite large. Without further assumptions, there are no general inequalities between distances $\hat{d}(q, q_j)$ and $\hat{d}(q, q_i)$, with $q_i \in R$, $q_j \in Q \setminus R$. Given the inequality $d(q, q_i) > d(q, q_j)$ on actual distances, the inequality $\hat{d}(q, q_i) > \hat{d}(q, q_j)$ on approximate distances would be desirable, but there are examples where the inequality on approximate distances does not hold.

9 Related Works

Our sparse distance approximation problem is related to the out-of-sample extension to the dimensionality reduction problem (Yang and Jin (2006); Bengio et al. (2003)). For the dimensionality reduction problem, a data set $X = \{x_1, \ldots, x_n\}$ and a metric $d(\cdot, \cdot)$ are given. The goal is to find a set of points $Y = \{y_1, \ldots, y_n\} \in \mathbb{R}^m$, such that each y_i "represents" its counterpart x_i. This "representation" is defined by either local or global constraints with regard to X and Y. The goal of the out-of-sample extension to dimensionality reduction is to compute a new point y given a real-time point x, without recomputing the mapping of X to Y.

The out-of-sample extension is compatible with algorithms that solve variants of the dimensionality reduction problem such as Multi-Dimensional Scaling (Cox and Cox

(2000)), Spectral Clustering (Weiss (1999)), Laplacian Eigenmaps (Betkin and Niyogi (2003)), Isomaps (Tenenbaum et al. (2000)), and Locally Linear Embedding (Roweis and Saul (2000)). These algorithms construct a (problem specific) kernel function $k_D(\cdot,\cdot)$ dependent on the sample data D and the distance function $d(\cdot,\cdot)$ (Bengio et al. (2003)). With k_D, a Gram matrix K of the sample data D is computed. Given an out-of-sample element x, its corresponding value y is computed from a kernel projection onto the eigenvectors of K using k_D (Schölkopf and Smola (2001)).

Nyström Method

Our proposed solution to the distance approximation problem is also comparable in performance to solutions which employ matrix approximation techniques such as the Nyström method. The Nyström method has been used to speed up the computation of kernel machines (Williams and Seeger (2001)) and has been used for improved performance in applications such as clustering (Chitta et al. (2011)) and manifold learning (Talwalkar et al. (2008)). Given an $n \times n$ positive definite matrix G and a parameter $m \ll n$, one can use the Nyström method to produce an $n \times n$ matrix $\tilde{G}^+$ of rank m that approximates G^{-1}. The runtime complexity of producing $\tilde{G}^+$ with the Nyström method is $O(m^2 n)$. This is asymptotically more computationally efficient than the $\Theta(n^3)$ runtime complexity of a standard matrix inversion algorithm.

The input to this calculation is an $n \times n$ matrix G and $m \ll n$ columns sampled from G, represented as an $n \times m$ matrix $G_{n,m}$. The $m \times m$ matrix $G_{m,m}$ consists of the intersection of these m columns with the corresponding m rows of G. The matrix $\tilde{G} \approx G_{n,m} G^+_{m,m} G^*_{n,m}$ is an approximation of G. Analogously, the matrix $\tilde{G}^+$ is an approximation of G^{-1}, and can be computed from $G_{n,m}$ and the singular value decomposition of $G_{m,m} = U_m \Sigma_m U^*_m$. The approximate eigenvalues $\tilde{\Sigma}$ and eigenvectors $\tilde{U}$ of G are $\tilde{\Sigma} = \left(\frac{n}{m}\right) \Sigma_m$ and $\tilde{U} = \sqrt{\frac{m}{n}} G_{n,m} U_m \Sigma^+_m$, respectively. From $\tilde{\Sigma}$ and $\tilde{U}$, the approximation $\tilde{G}^+ = \tilde{U}_m \tilde{\Sigma}^+_m \tilde{U}^*_m$ of G^{-1} is computed.

Kernel Projection Approximation Performance		
Method	Offline Computation	Online Computation
Nyström	$\theta(m^2n + mne)$	$\theta(ne + mn)$
Kernel Semi-least Squares	$\theta(mn^2 + mne)$	$\theta(me + mn)$

Table 1: The variable e is the runtime complexity of the kernel function k and n is the size of the dataset Q. For the Nyström method, m is the rank of the approximate Gram matrix inverse $\tilde{K}^+$. For the Kernel Semi-least Squares method, m is the size of the subset $R \subseteq Q$. Typically $m \ll n$.

The Nyström method can be used to create the approximation, $\hat{\beta} = \tilde{K}^+ \mathbf{k}_Q(q)$, of the standard kernel projection, $\beta = K^{-1}\mathbf{k}_Q(q)$, where K is the Gram matrix with respect to the kernel function k and the training set Q. The complexity analysis of the kernel projection approximation derived from the Nyström and Kernel Semi-least Squares methods can be seen in Table 1. For the analysis of the offline computations, we assume the kernel function k is only computed on entries in the Gram matrix K. The Nyström method provides computational savings in the offline computation of K^{-1} whereas the Kernel Semi-least Squares method provides computational savings in both the offline computation of K^+_{RQ} and the online computation of the kernel empircal map $\mathbf{k}_R(q)$ of the subset $R \subseteq Q$.

10 Experiments

We tested the Kernel Semi-least Squares method on the Sheffield Face Database (Graham and Allinson (1998)). The face database consists of 564 images of 20 individuals, covering a mixed range of race, sex, and age. The images of the faces were in range of poses including profile and frontal views. There were an average of 28 images per individual. Each picture is approximately 220 $\times$ 220 greyscale pixels represented with 256-bits. Example pictures of the individuals in different poses can be seen in Figure 3.

Figure 3: Example images from two individuals in the Sheffield Database

We used a type of Hamming distance, $h(\cdot, \cdot)$, as the target distance. To compute this distance, two input images are first converted from greyscale to binary. The greyscale value at every position in the image is set to 1 if it is above a pre-determined threshold, and it is set to 0 otherwise. The Hamming distance is computed by counting the number of positions where the two converted binary images have different values. Each individual has a unique threshold, determined by averaging all the greyscale values of all the individual's images in the database. This experiment represents the envisioned application of the Kernel Semi-least Squares method: the distances are expensive to compute, but, as our experiment shows, they can be isometrically-embedded into a low-dimensional vector space.

For each individual, we tested the accuracy of the approximate Hamming distance $\hat{h}(\cdot, \cdot)$ using leave-one-out cross-validation. Each image q from the individual's set of images Q was removed in turn and from the remaining group, Q/q, the optimal subsets R_m of sizes $m = 1 \ldots 10$ were computed. Each R_m minimized the cost function of Equation 21 over all subsets of size m, with the set of candidate subsets being of the form $\mathcal{R} = \{R : R \subset Q, |R| = m\}$. For each subset R_m of size m, we constructed an approximate Hamming distance $\hat{h}_m(\cdot, \cdot)$. We used the kernel of Equation 3, with

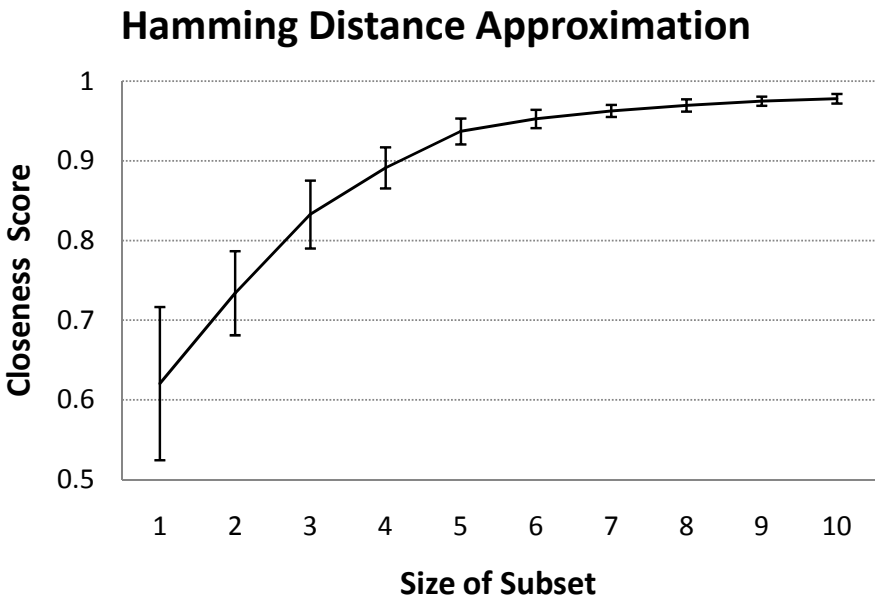

Figure 4: The performance of our Kernel Semi-least Squares method for distance approximation for each subset size. The closeness score was determined using leave-one-out cross-validation over 20 individuals. There was an average of 28 images per individual. The error bars represent the standard deviation of the closeness score.

d being the target Hamming distance and with $g(\cdot)$ set to a constant function. We chose this kernel because it can be seen as a variant of an intersection kernel, which we have seen has good discriminatory properties. The projection matrix W of the subset projection used in the distance approximation (Equation 14) was set to 0. The *difference score* for each image q removed from the set of images Q, using the best subset of size m, was

$$D(m,q,Q) \;=\; \frac{1}{|Q/q|}\sum_{q'\in Q/q}\frac{\left|\hat{h}_m(q,q')-h(q,q')\right|}{h(q,q')}. \tag{31}$$

The *closeness score* for each subset size m was computed from the average of the difference score over all individuals $\mathcal{I}$, with

$$S(m) \;=\; 1-\frac{1}{|\mathcal{I}|}\sum_{Q\in\mathcal{I}}\frac{1}{|Q|}\sum_{q\in Q}D(m,q,Q). \tag{32}$$

We computed closeness score for subset sizes 1 to 10. Our results show that a small subset can be used to approximate the Hamming distance with a high degree of accuracy, as seen in Figure 4.

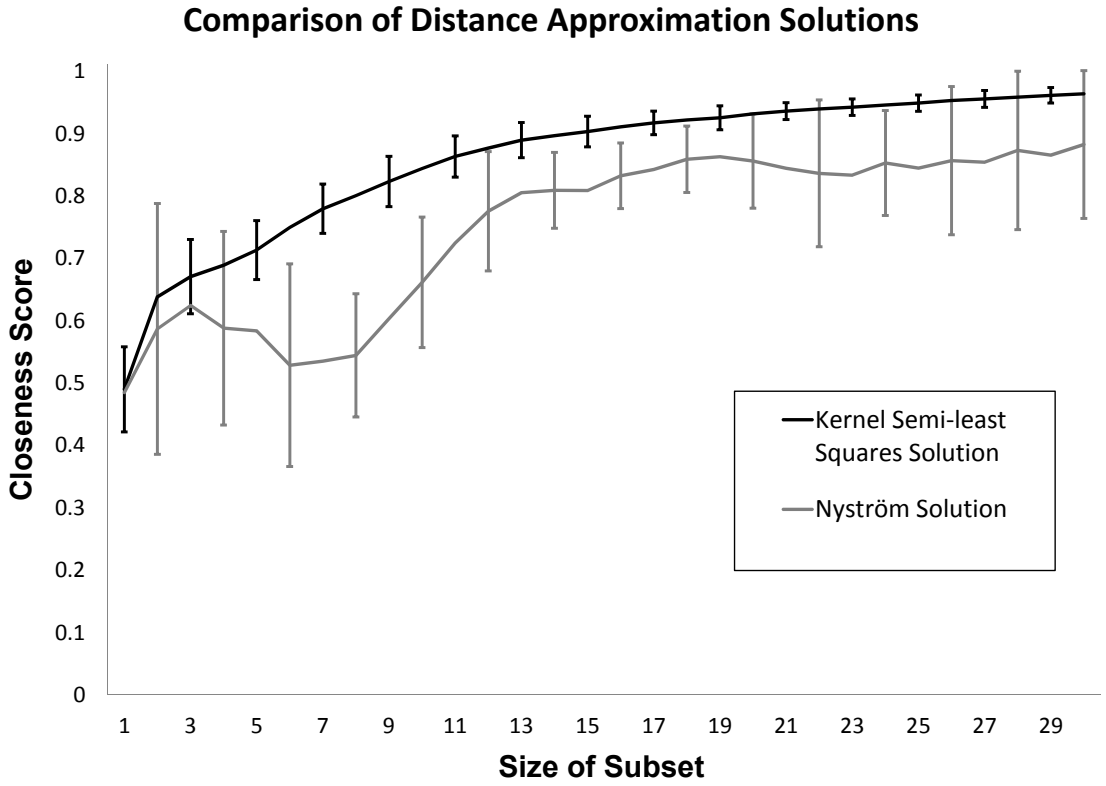

Figure 5: The performance of our Kernel Semi-least Squares method in relation to the Nyström method in the task of distance approximation. The closeness score was computed using leave-one-out cross validation over 20 individuals. There were 78 images for each individual. The error bars represent the standard deviation of the closeness score.

Nyström Method Comparison

We also compared the performances of two solutions to our posed problem of distance approximation. The first solution applied the Kernel Semi-least Squares method as described in this chapter. The second solution is identical to the "Kernel Semi-least Squares solution" except the Nyström method was used to approximate the kernel

projection, as described in section 9. For the experiments, we used images of faces from the CMU Pose, Illumination, and Expression (PIE) Database (Sim et al. (2002)). The database consists of color pictures of faces of individals under different illumination conditions, poses and expressions. We selected 20 individuals randomly from this dataset. For each individual selected, a dataset Q was constructed, consisting of 78 manually cropped 150×250 color images of the person's face under different illumination conditions and poses.

For each individual dataset Q, we tested the "Kernel Semi-least Squares solution" and the "Nyström solution" on the hamming distance $h(\cdot, \cdot)$ using leave-one-out cross validation in the same manner as the previous experiment. For each individual dataset, the optimal subsets of sizes $n = 1 \ldots 30$ were selected using the criteria of section 7.

The results were measured and aggregated over the individuals using the distance $D(m, q, Q)$ and closeness $S(m)$ scores described earlier in this section. The results, as shown in Figure 5, indicate the Kernel Semi-least Squares method produced a more accurate and precise approximation of the target distance than the Nyström method.

11 Discussion

We presented a kernel based solution to the sparse distance approximation problem, where the given distance metric d is too computationally expensive to compute exhaustively. Our method uses the subset projection to map an observation to the span of a training set in the kernel feature space. The derivation of the subset projection is derived by extending Rao and Mitra's Semi-least Squares problem with kernel methods.

The kernel projection requires the computation of $d(x, x_i)$ for each $x_i \in X$. In practice, this distance computation can be prohibitively expensive. Our posed problem takes this into consideration, with the aim to have less than n computations of $d(\cdot, \cdot)$. The goal of our formulation is to compute the distances from a real-time element to each element in the training set, instead of a general mapping to a low dimensional

Euclidean space.

A benefit of our method is its simplicity. The distance approximation method can be described with two equations (Equations 14 and 15). The pre-computing requirements are light, consisting of evaluation of the kernel function and matrix (pseudo)inverses.

Future work consists of developing a method for training the arbitrary projection matrix W used in the subset projection. It is an open question how to compute the optimal subset R_o efficiently. We provide several optimization criteria in Equations 21, 24 and 27, which can be computed via enumeration of all possible subsets of size m. The more general question of how closely the coefficient vector β in Equation 11 is approximated by $\hat{\beta}$ in Equation 14 remains open.

Acknowledgments

The authors gratefully acknowledge NSF funding (HCC grants IIS-0713229 and IIS-0910908) and are grateful to Stan Sclaroff for insightful discussions.

References

Belkin, M., and Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396.

Bengio, Y., Paiement J.-F., and Vincent, P., 2003. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems*, pp. 177–184.

Chitta, R., Jin, R., Havens, T., and Jain, A., 2011. Approximate Kernel k-means: Solution to Large Scale Kernel Clustering. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 895–903.

Cox, T. and Cox, M., 2000. *Multidimensional Scaling, Second Edition*. Chapman & Hall/CRC.

Graham, D. and Allinson, N., 1998. Characterizing virtual eigensignatures for general purpose face recognition. In *Face Recognition: From Theory to Applications*, pages 446–456. The Sheffield (previously UMIST) Face Database, `http://www.shef.ac.uk/eee/research/iel/research/face.html`.

Rao, C. R. and Mitra, S. K., 1971. Further contributions to the theory of generalized inverse of matrices and its applications. Sankhya-: The Indian Journal of Statistics, Series A 33:(3):289–300. `http://www.jstor.org/stable/25049740`.

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

Schölkopf, B., 2000. The kernel trick for distances. In: Advances in Neural Information Processing Systems. pp. 301–307.

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K.-R., Rätsch, G., and Smola, A. J., 1999. Input space versus feature space in kernel-based methods. IEEE Transactions on Neural Networks 10 (5), 1000–1017.

Schölkopf, B. and Smola, A., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press.

Sim, T., Baker, S., and Bsat, M., 2002. The CMU Pose, Illumination, and Expression (PIE) Database. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*.

Talwalkar, A., Kumar, S., and Rowley, H., 2008. Large-Scale Manifold Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008*, 8 pp.

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

Weiss, Y., 1999. Segmentation using eigenvectors: A unifying view. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2, pp. 975–982.

Williams, C. and Seeger, M., 2001. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems*, pp. 682–688.

Yang, L. and Jin, R., 2006. Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University.