10-701/15-781 Recitation : Kernels

Manojit Nandi

February 27, 2014

Outline

Mathematical Theory

- Banach Space and Hilbert Spaces
- Kernels
- Commonly Used Kernels

Kernel Theory

- One Weird Kernel Trick
- Representer Theorem

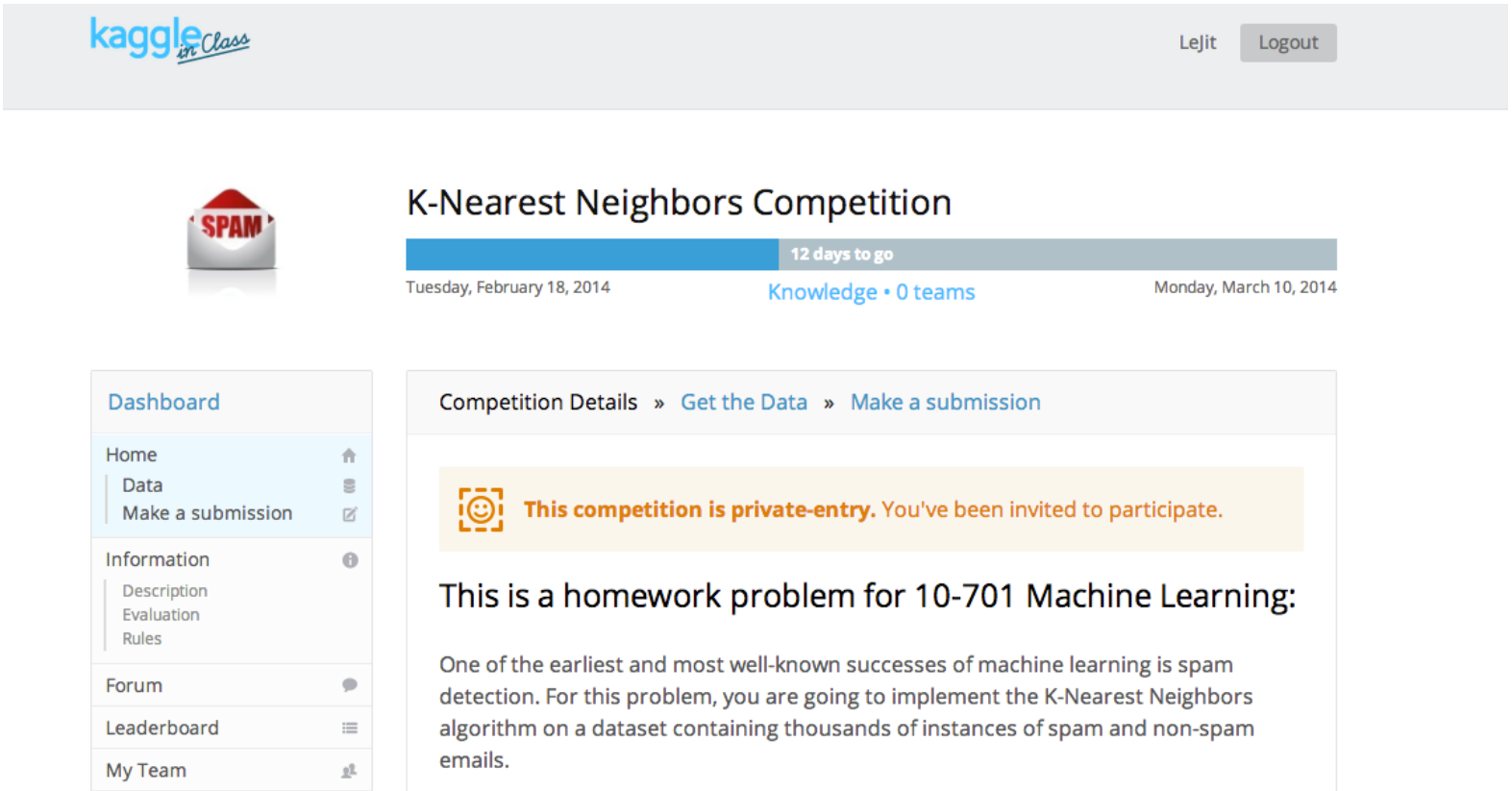
Do You want to Build a ~~*Snowman*~~ Kernel? It doesn't have to be ~~*Snowman*~~ Kernel.

- Operations that Preserve/Create Kernels

Current Research in Kernel Theory

- Flaxman Test
- Fast Food

References


kaggle in Class
LeJit
Logout
SPAM
K-Nearest Neighbors Competition
12 days to go
Tuesday, February 18, 2014
Knowledge • 0 teams
Monday, March 10, 2014
Dashboard
Home
Data
Make a submission
Information
Description
Evaluation
Rules
Forum
Leaderboard
My Team
Competition Details » Get the Data » Make a submission
This competition is private-entry. You've been invited to participate.
This is a homework problem for 10-701 Machine Learning:
One of the earliest and most well-known successes of machine learning is spam detection. For this problem, you are going to implement the K-Nearest Neighbors algorithm on a dataset containing thousands of instances of spam and non-spam emails.

IMPORTANT: The Kernels used in Support Vector Machines are different from the Kernels used in Kernel Regression.

To avoid confusion, I'll refer to the Support Vector Machines kernels as **Mercer Kernels** and the Kernel Regression kernels as **Smoothing Kernels**

Recall: A square matrix $A \in \mathbb{R}^{NxN}$ is positive semi-definite if for all vectors $u \in \mathbb{R}^n$, $u^T Au \geq 0$.

For some kernel function $\mathbf{K} : \mathcal{X}x\mathcal{X} \rightarrow \mathbb{R}$ defined over a set $\mathcal{X}; |\mathcal{X}| = n$, the Gram Matrix G is an nXn matrix such that $G_{ij} = K(x_i, x_j) =< \phi(x_i), \phi(x_j) >$

Mathematical Theory

Banach Space and Hilbert Spaces

A Banach Space is a complete Vector Space V equipped with a norm.

Examples:

- l_p^m Spaces: $\mathbf{R}^m$ equipped with the norm $\|x\| = (\sum_{i=1}^{m} |x_i|^p)^{\frac{1}{p}}$
- Function Spaces, $\|f\| := (\int_{\mathbf{X}} |f(x)|^P \, dx)^{\frac{1}{p}}$

A Hilbert Space is a complete Vector Space V equipped with an inner product $< ., . >$.
The inner product defines a norm ($\|x\| =< x, x >$), so Hilbert Spaces are a special case of Banach Spaces.

Example:

Euclidean Space with standard inner product: $x, y \in \mathbb{R}^n; < x, y >= \sum_{i=1}^{n} x_i y_i = x^T y$

Because Kernel functions are the inner product in some higher-dimensional space, each kernel function corresponds to some Hilbert Space $\mathcal{H}$.

Kernels

A Mercer Kernel is a function $\mathcal{K}$ defined over a set $\mathcal{X}$ such that $\mathcal{K} : \mathcal{X}x\mathcal{X} \rightarrow \mathbb{R}$. The Mercer Kernel represents an inner product in some higher dimensional space

A Mercer Kernel is symmetric and positive semi-definite. By Mercer's theorem, we know that any symmetric positive semi-definite kernel represents the inner product in some higher-dimensional Hilbert Space $\mathcal{H}$.

Symmetric and Positive Semi-Definite $\Leftrightarrow$ Kernel Function $\Leftrightarrow <\phi(x), \phi(x') >$ for some $\phi(.)$.

Why Symmetric?

Inner products are symmetric by definition, so therefore if the kernel function represents an inner product in some Hilbert Space, then the kernel function must be symmetric as well.

$K(x, z) =< \phi(x), \phi(z) >=< \phi(z), \phi(x) >= K(z, x)$.

Why Positive Semi-definite?

In order to be positive semi-definite, the Gram Matrix $G \in \mathbb{R}^{nXn}$ must satisfy $\mathbf{u}^T G\mathbf{u} \geq 0 \ \forall \mathbf{u} \in \mathbb{R}^n$. Using functional analysis, one can show that $\mathbf{u}^T G\mathbf{u}$ corresponds to $< h_u, h_u >$ for some h_u in some Hilbert Space $\mathcal{H}$. Because $< h_u, h_u > \geq 0$ by the definition of an inner product, then $< h_u, h_u > \geq 0$ and $< h_u, h_u >= \mathbf{u}^T G\mathbf{u} \Rightarrow \mathbf{u}^T G\mathbf{u} \geq 0$.

Why are Kernels useful?

Let $\mathbf{x} = (x_1, x_2)$ and $\mathbf{z} = (z_1, z_2)$. Then,

$$K(x,z) =< \mathbf{x}, \mathbf{z} >^2 = (x_1 z_1 + x_2 z_2)^2 = (x_1^2 z_1^2) + (2 x_1 z_1 x_2 z_2) + (x_2^2 z_2^2)$$
$$=< (x_1^2, \sqrt{2} x_1 x_2, x_2^2), (z_1^2, \sqrt{2} z_1 z_2, z_2^2) >=< \phi(x), \phi(z) >$$

Where $\phi(\mathbf{x}) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$.

What if $\mathbf{x} = (x_1, x_2, x_3, x_4)$, $\mathbf{z} = (z_1, z_2, z_3, z_4)$, and $K(\mathbf{x}, \mathbf{z}) =< x, z >^{50}$. In this case, $\phi(x)$ and $\phi(z)$ are 23426^* dimensional vectors.

$*$: From combinatorics, 50 unlabeled balls into 4 labeled boxes $= \binom{53}{3}$.

We can either:

1. Calculate $< \mathbf{x}, \mathbf{z} >$ (Inner product of two four-dimension vectors) and raise this value (a float) to the 50^{th} power.

2. OR map $\mathbf{x}$ and $\mathbf{z}$ to $\phi(\mathbf{x})$ and $\phi(\mathbf{z})$, and calculate the inner product of two 23426 dimensional vectors.

Which is faster?

Commonly Used Kernels

Some commonly used Kernels are:

- Linear Kernel: $K(x, z) =< x, z >$
- Polynomial Kernel: $K(x, z) =< x, z >^d$ where d is the degree of the polynomial
- Gaussian RBF: $K(x, z) = exp(-\frac{1}{2\sigma^2} \|x - z\|^2) = exp(-\gamma \|x - z\|^2)$
- Sigmoid Kernel: $K(x, z) = \tanh(\alpha x^T z + \beta)$

Other Kernels:

1. Laplacian Kernel
2. ANOVA Kernel
3. Circular Kernel
4. Spherical Kernel
5. Wave Kernel
6. Power Kernel
7. Log Kernel
8. B-Spline Kernel
9. Bessel Kernel
10. Cauchy Kernel
11. χ^2 Kernel
12. Wavelet Kernel
13. Bayesian Kernel
14. Histogram Kernel

As you can see, there are a lot of kernels.

The Gaussian RBF Kernel is infinite

In class, Barnabas mentioned that the Gaussian RBF Kernel corresponds to an infinite dimensional vector space. From the Moore-Aronszajn theorem, we know there is a unique Hilbert space of functions for with the Gaussian RBF is a reproducing kernel.

One can show that this Hilbert Space has a infinite basis, and so the Gaussian RBF Kernel corresponds to an infinite-dimensional vector space. A YouTube video of Abu-Mostafa explaining this in more detail has been included in the references at the end of this presentation.

Kernel Theory

One Weird Kernel Trick

The kernel trick is one the reasons kernel methods are so popular in machine learning. With the kernel trick, we can substitute any dot product with a kernel function. This means any linear dot product in the formulation of a machine learning algorithm can be replaced with a non-linear kernel function. The non-linear kernel function may allow us to find separation or structure in the data that was not present in the linear dot product.

Kernel Trick: $< x_i, x_j >$ can be replaced with $K(x_i, x_j)$ for any valid kernel function K. Furthermore, we can swap any kernel function with any other kernel function.

Some examples of algorithms that take advantage of the kernel trick:

- Support Vector Machines (Poster Child for Kernel Methods).
- Kernel Principal Component Analysis
- Kernel Independent Component Analysis
- Kernel K-Means algorithm
- Kernel Gaussian Process Regression
- Kernel Deep Learning

Representer Theorem

Theorem

Let $\mathcal{X}$ be a non-empty set and k a positive-definite real-valued kernel on on $\mathcal{X}x\mathcal{X}$ with the corresponding reproducing kernel Hilbert Space H_k. Given a training sample $[(x_1, y_1), ..., (x_n, y_n)] \in \mathcal{X}x\mathbb{R}$, a strictly monotonic increasing real-valued function $g : [0, \infty) \to \mathbb{R}$, and an arbitrary emprical loss function $E : (\mathcal{X}X\mathbb{R}^2)^m \to \mathbb{R} \cup \infty$, then for any f^* satisfying

$$\hat{f} = \underset{f \in \mathcal{H}_k}{argmin}\{E[(x_1, y_1, f(x_1), ..., (x_n, y_n, f(x_n))]\} + g(\|f\|)$$

f^* can be represented in the form $f^*(.) = \sum_{i=1}^{n} \alpha_i k(\cdot, x_i)$,

where $\alpha_i \in \mathbb{R}$ for all α_i

Example: Let H_k be some RKHS function space with a kernel $k(.,.)$. Let $[(x_1, y_1), ..., (x_m, y_m)]$ be a training input-output set, and our task is to find f^* that minimizes the following regularized kernel.

$$\hat{f} = \underset{f \in \mathcal{H}_k}{argmin}(\prod_{i=1}^{n} |f(x_i)|^6) \sum_{i=1}^{n} [\left|\sin(\|x_i\|^{y_i - f(x_i)})\right|^{25} + y_i \, |f(x_i)|^{249}] + \exp(\|f\|_{\mathcal{F}})$$

$$\hat{f} = \underset{f \in \mathcal{H}_k}{argmin}(\prod_{i=1}^{n} |f(x_i)|^6) \sum_{i=1}^{n} [\left|\sin(\|x_i\|^{y_i - f(x_i)})\right|^{25} + y_i \, |f(x_i)|^{249}] + \exp(\|f\|_{\mathcal{F}})$$

Well $exp(.)$ is a strictly monotonic increasing function, so $exp(\|f\|_{\mathcal{F}})$ fits the $g(\|f\|_{\mathcal{F}})$ part.

$(\prod_{i=1}^{n} |f(x_i)|^6) \sum_{i=1}^{n} [\left|\sin(\|x_i\|^{y_i - f(x_i)})\right|^{25} + y_i \, |f(x_i)|^{249}]$ is some arbitrary emperical loss function of the $[x_i, y_i, f(x_i)]$. So this optimization can be expressed as

$$\hat{f} = \underset{f \in \mathcal{H}_k}{argmin}\{E[(x_1, y_1, f(x_1), ..., (x_n, y_n, f(x_n))]\} + g(\|f\|)$$

Therefore, by the Representer Theorem, $f^*(.) = \sum_{i=1}^{n} \alpha_i K(x_i, .)$

Do You want to Build a ~~*Snowman*~~ Kernel? It doesn't have to be ~~*Snowman*~~ Kernel.

Operations that Preserve/Create Kernels

Let $k_1, ...k_m$ be valid kernel functions defined over some set $\mathcal{X}$ such that $|\mathcal{X}| = n$, and let $\alpha_1, ..., \alpha_m$ be non-negative coefficents. The following are valid kernels.

- K(x,z) $= \sum_{i=1}^{m} \alpha_i k_i(x, z)$ (Closed under non-negative linear multiplication)
- K(x,z) $= \prod_{i=1}^{m} k_i(x, z)$ (Closed under multiplication)
- K(x,z) $= k_1(f(x), f(z))$ for any function $f : \mathcal{X} \to \mathcal{X}$
- K(x,z) $= g(x)g(z)$ for any function $g : \mathcal{X} \to \mathbb{R}$
- K(x,z) $= x^T A^T A z$ for any matrix $A \in \mathbb{R}^{mXn}$

Proof: K(x,z) $= \sum_{i=1}^{m} \alpha_i k_i(x, z)$ is a valid kernel

Symmetry: $K(z, x) = \sum_{i=1}^{m} \alpha_i k_i(z, x)$. Because each k_i is a kernel function, it is symmetric, so $k_i(z, x) = k_i(x, z)$. Therefore,

$$K(z, x) = \sum_{i=1}^{m} \alpha_i k_i(z, x) = \sum_{i=1}^{m} \alpha_i k_i(x, z) = K(x, z) \Rightarrow K(z, x) = K(x, z)$$

Positive Semi-definite: Let $\mathbf{u} \in \mathbb{R}^n$ be arbitrary. The Gram matrix of K, denoted by G has the property, $G_{i,j} = K(x_i, x_j) = \sum_{i=1}^{m} \alpha_i k_i(z, x) \Rightarrow G = \alpha_1 G_1 + ... + \alpha_m G_m$. Now $\mathbf{u}^T G \mathbf{u} = \mathbf{u}^T (\alpha_1 G_1 + ... + \alpha_m G_m)\mathbf{u}$

$= \alpha_1 \mathbf{u}^T G_1 \mathbf{u} + ... + \alpha_m \mathbf{u}^T G_m \mathbf{u} = \sum_{i=1}^{m} \alpha_i \mathbf{u}^T G_i \mathbf{u}$.

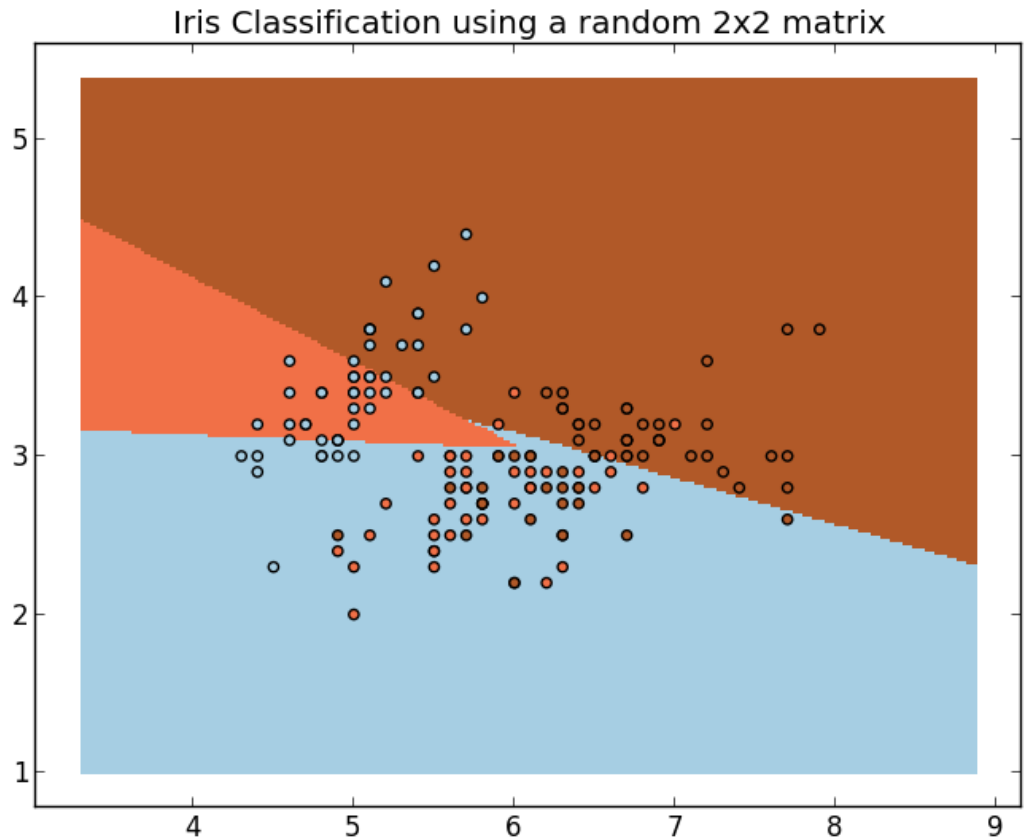
$\mathbf{u}^T G_i \mathbf{u} \geq 0$, and $\alpha_i \geq 0$, so $\alpha_i \mathbf{u}^T G_i \mathbf{u} \geq 0$.

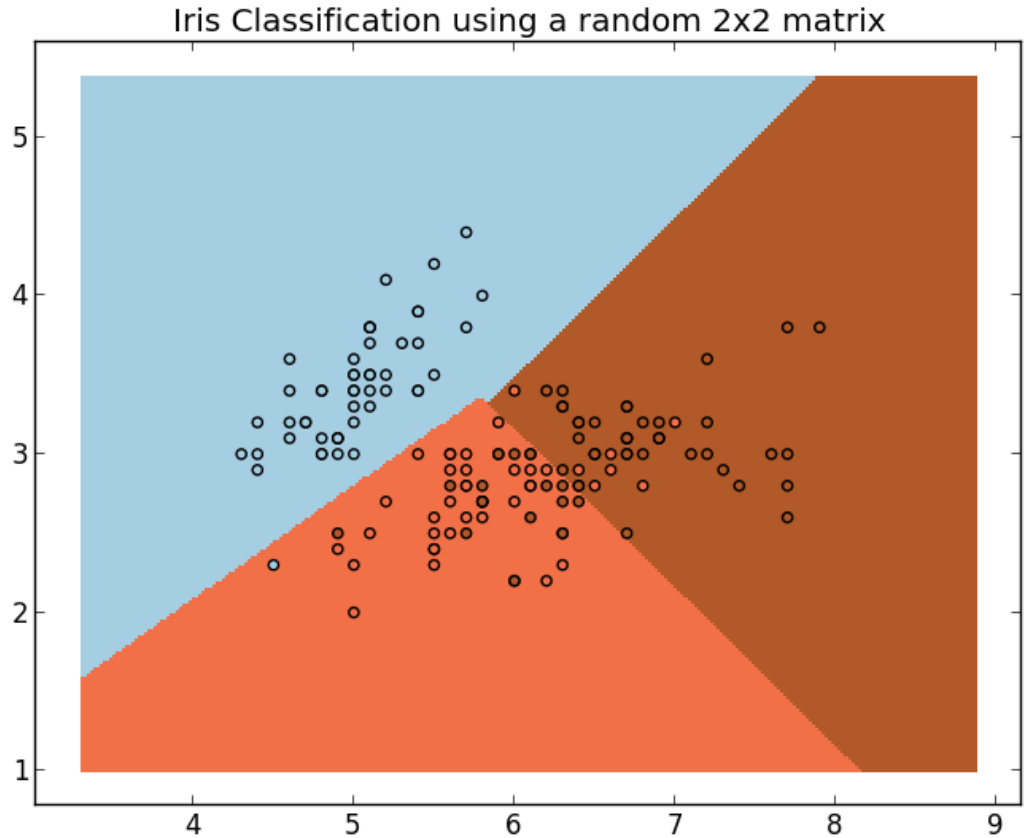
Proof: K(x,z) $= x^T A^T Az$ for any matrix $A \in \mathbb{R}^{mXn}$ is a valid Kernel.

For this proof, we are going to show $K(x, z)$ is an inner product on some Hilbert Space. Let $\phi(x) = Ax$, then $< \phi(x), \phi(z) >= \phi(x)^T \phi(z) = (Ax)^T (Az) = x^T A^T Az = K(x, z) \Rightarrow < \phi(x), \phi(z) >= K(x, z)$.

Therefore, $K(x, z)$ is an inner product on some Hilbert Space.

Just because it is a valid kernel does not mean it is a good kernel




Iris Classification using a random 2x2 matrix
5
4
3
2
1
4
5
6
7
8
9

Some Exercises Prove the following are not valid kernels:

1. $K(x, z) = k_1(x, z) - k_2(x, z)$ for k_1, k_2 valid kernels

2. $K(x, z) = \exp(\gamma \|x - z\|^2)$ for some $\gamma > 0$

Current Research in Kernel Theory

Flaxman Test

Correlates of homicide: New space/time interaction tests for spatiotemporal point processes

Goal: Develop a statistical test that can test for the strength of interaction between time and space for different types of homicides.

Result: Using Reproducing Kernel Hilbert spaces, one can project the space-time data into a higher dimensional space that can test for a significant interaction of a kernalized distance of space and time using the Hilbert-Schmidt Independence Criterion.

Fast Food

Fastfood - Approximating Kernel Expansions in Loglinear time

Problem: Kernel methods do not scale well to large datasets, especially during prediction time, because the algorithm has to compute the kernel distance between the new vector and all of the data in the training set.

Solution: Using advanced mathematical black magic, dense random Gaussian matrices can be approximated using Hadamard matrices and diagonal Gaussian matrices.

$$V = \frac{1}{\sigma\sqrt{d}} SHG \prod HB$$

Where $\prod \in \{0,1\}^n$ is a permutation matrix, H is a Hadamard matrix, S is a random scaling matrix with elements along the diagonal, B has random $\{\pm 1\}$ along the diagonal, and G has random Gaussian entries.

This algorithm performs 100x faster than the previous leading algorithm (Random Kitchen Sinks) and uses 1000x less memory.

References

Representer Theorem and Kernel Methods
Predicting Structured Data by Alex Smola
Kernel Machines
String and Tree Kernels
Why Gaussian Kernel is Infinite

Questions?