Northeastern University

Solving Interpretable Kernel Dimension Reduction

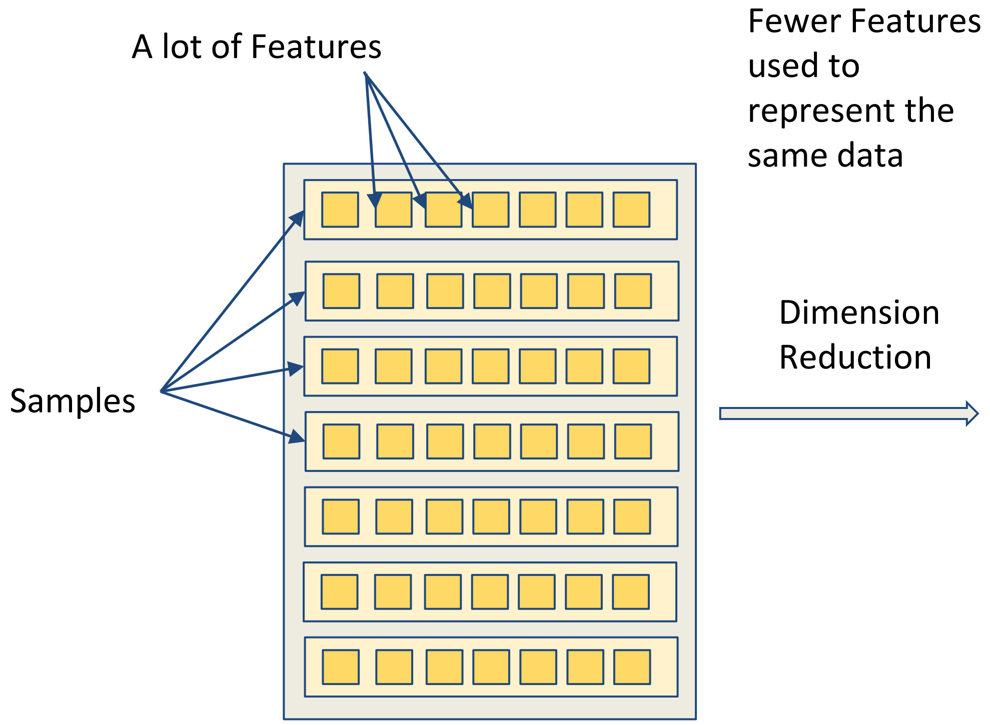
Chieh Wu, Jared Miller, Mario Sznaier, and Jennifer Dy

Electrical and Computer Engineering Dep
Northeastern University

Video Presentation For NeurIPS 2019

Source code : https://github.com/chieh-neu/ISM_supervised_DR

Dimension Reduction



Advantage of Dimension Reduction

1. Smaller size
2. Easier to handle
3. Faster to process
4. Smaller memory storage
5. Remove unimportant info

The Key is to

1. Remove unimportant data
2. Keep important data

Principal Component Analysis (PCA)

PCA is by far the most popular Dimension Reduction technique, and it is interpretable !!!

Given a sample $x \in \mathbb{R}^d$, PCA finds the $W \in \mathbb{R}^{d \times q}$ such that $W^T x$ retains the most important information.

We know exactly how the new features relate to the original features.

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 w_{11} + x_2 w_{12} + x_3 w_{13} \\ x_1 w_{21} + x_2 w_{22} + x_3 w_{23} \end{bmatrix}$$

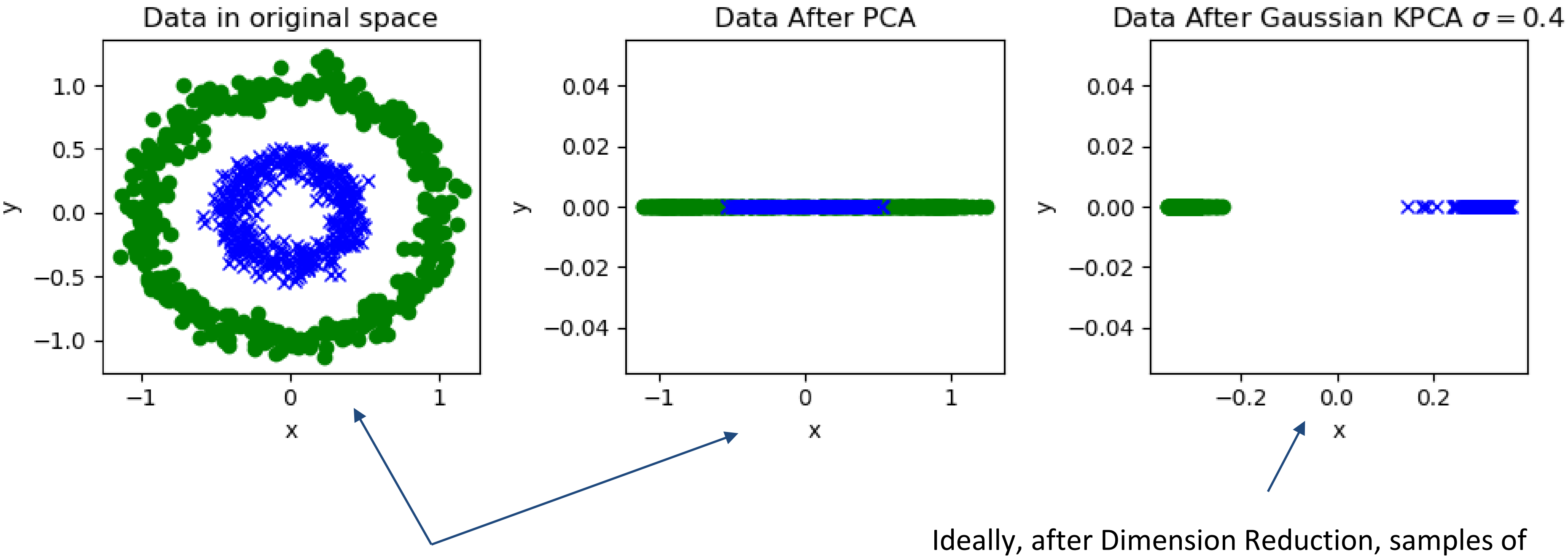
For example :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}$$

Here, we know exactly how the new features relate to the old features.

Principal Component Analysis (PCA)

But PCA only captures Linear Relationships!!!!

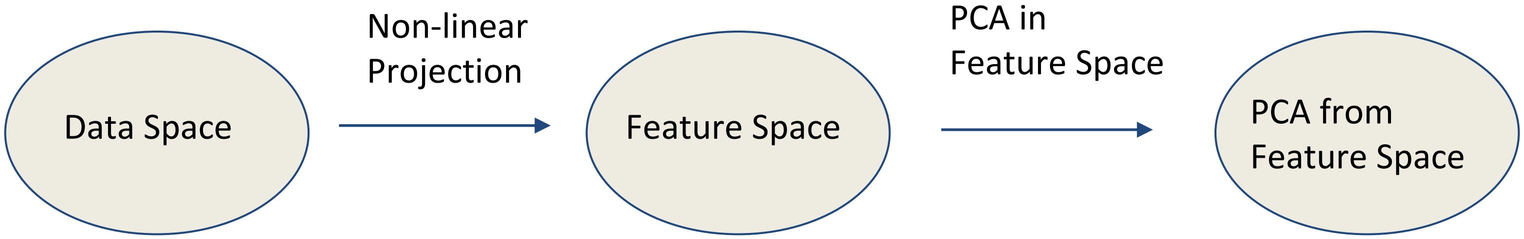


After Dimension Reduction, you can't tell that Blue and Green are actually separated.

Ideally, after Dimension Reduction, samples of the same group should stay close together.

Note : This requires us to also capture non-linear relationships!!!

However, we know from the kernel community, if you first project the data into a higher dimensional feature space, non-linear relationships can become linearly separable.



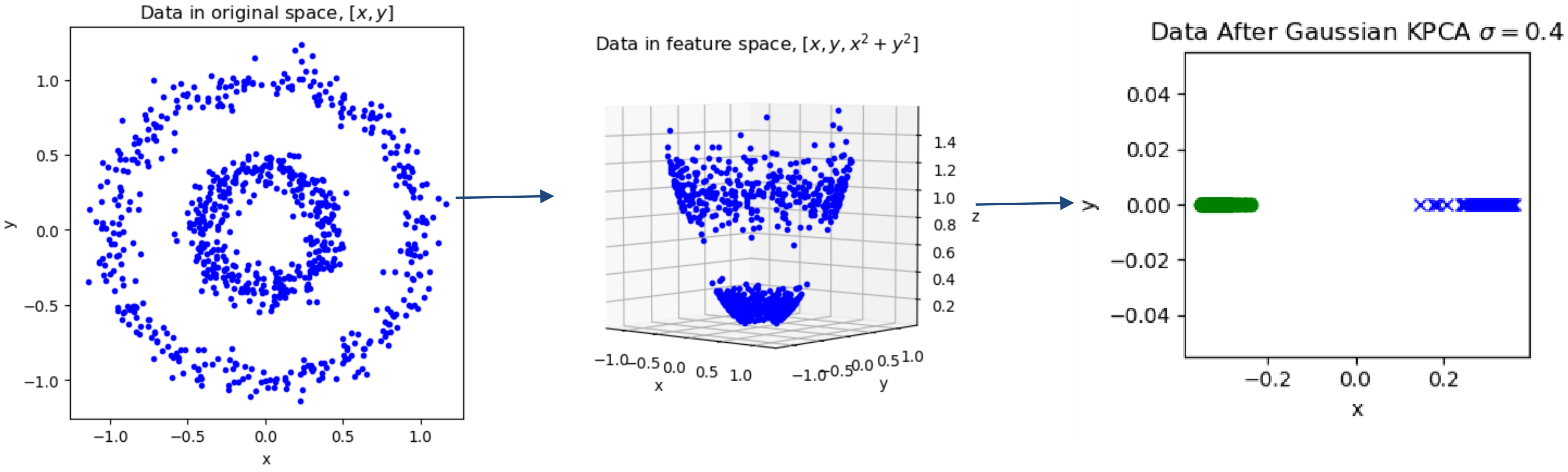
Data Space

Non-linear Projection

Feature Space

PCA in Feature Space

PCA from Feature Space



This is called the Kernel PCA, or KPCA.

KPCA is very powerful, but …….

Problem 1: You cannot use labels to guide the dimension reduction

Problem 2: Since KPCA is PCA in the feature space, it's not obvious what they mean.

Here is the Gaussian Kernel feature map:

$$\phi(x) = e^{-x^2/2\sigma^2}\left[1, \sqrt{\frac{1}{1!\sigma^2}}x, \sqrt{\frac{1}{2!\sigma^4}}x^2, \sqrt{\frac{1}{3!\sigma^6}}x^3, \ldots\right]^T$$

Not too obvious what running PCA on these features mean.

Interpretable Kernel Dimension Reduction solves both problems...

KPCA → $\Phi(X)\,W$

Finds the W matrix in Feature Space that keeps the most information.

Interpretable KDR → $\Phi(XW)$

Finds the W matrix in **original data space** that keeps the most information in the Feature Space.

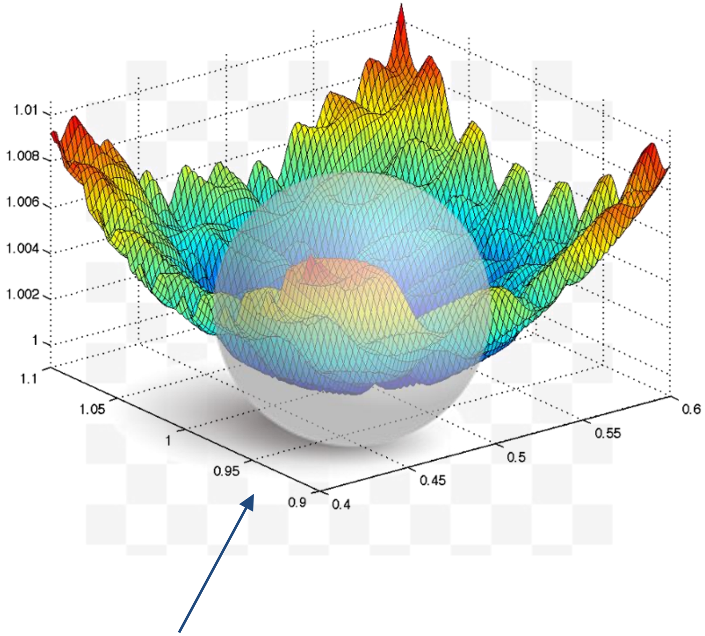
HSIC is a general objective for capturing non-linear dependence to achieve IKDR.

$$\max_{W} HSIC(XW, Y) \quad \text{s.t } W^T W = I$$

HSIC(X,Y) measures the non-linear dependence between X and Y in Feature Space.

Although this make the solution interpretable, it is very difficult to solve.

Unfortunately, IKDR is very difficult to solve !!



This is a highly non-linear, non-convex shape where the solution must intersect with a hypersphere.

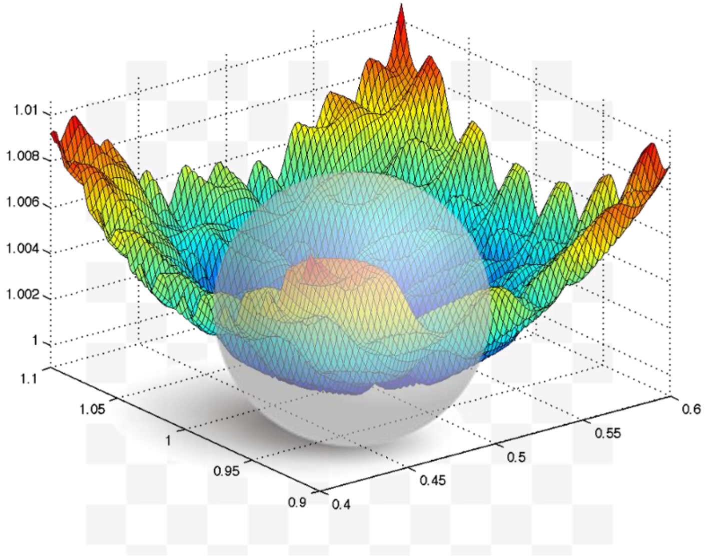
A generic IKDR problem :

$$\max_{W} \sum_{i,j} \Gamma_{i,j} K_{XW_{i,j}} \text{ s.t } W^T W = I$$

Using a Gaussian Kernel :

$$\max_{W} \sum_{i,j} \Gamma_{i,j} e^{-\frac{(W^T x_i - W^T x_j)^2}{2\sigma^2}} \text{ s.t } W^T W = I$$

There are many existing ways to solve this !!!



Dimension Growth:

- very slow
- Difficult to implement
- stuck at saddle point
- poor results

Optimization Via Stiefel Manifold:

- slow
- stuck at saddle point
- Difficult to implement
- Decent results

Solve Via SGD :

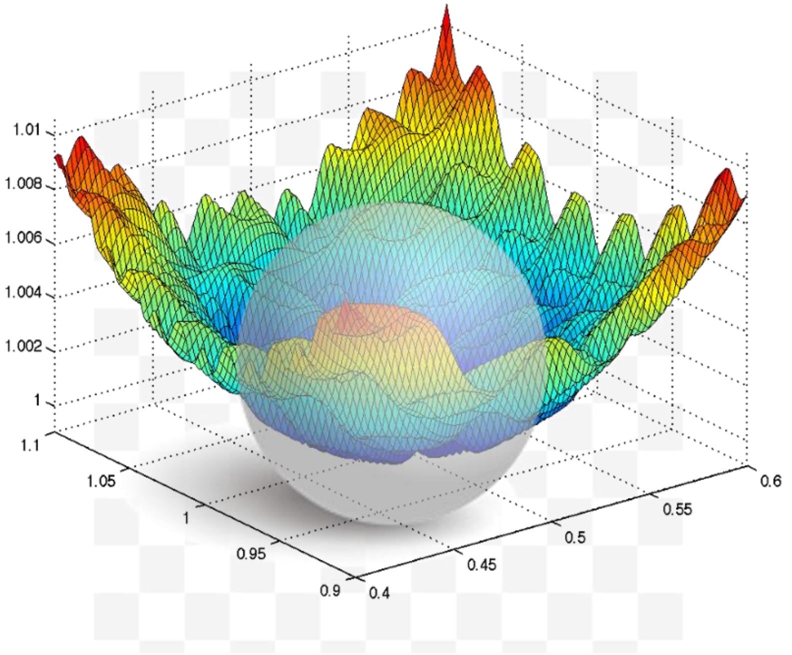
- Slow
- stuck at saddle point
- Easy to implement
- Not good results

We propose the Iterative Spectral Method (ISM) :

- very fast
- doesn’t get stuck at saddle point (not gradient based)
- Easy to implement
- Very good results

We discovered a solution for the IKDR problem for a family of kernels.

The family of kernels is called the ISM Family.

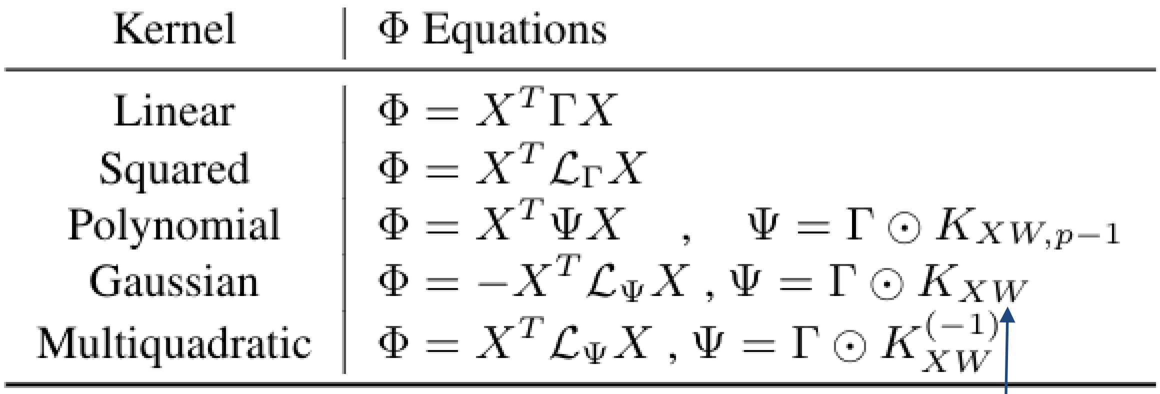


$$\max_{W} \sum_{i,j} \Gamma_{i,j} K_{XW_{i,j}} \text{ s.t } W^T W = I$$

We discovered that if a kernel is within the ISM Family, then the kernel has an associated Scaled Covariance Matrix Φ.

Just like PCA, the optimal solution W is the most dominant eigenvectors of Φ.

Here are some examples of kernels in the family and how the Scaled Covariance Matrix can be computed.

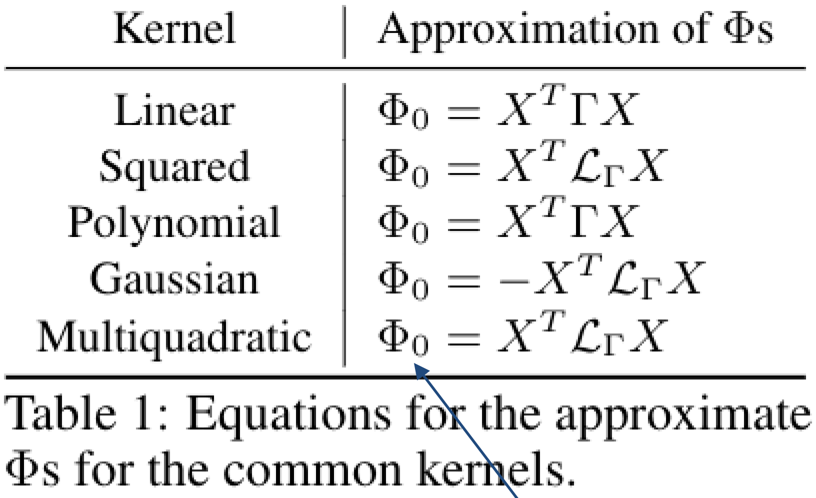


Kernel	Φ Equations
Linear	$\Phi = X^T \Gamma X$
Squared	$\Phi = X^T \mathcal{L}_\Gamma X$
Polynomial	$\Phi = X^T \Psi X \quad , \quad \Psi = \Gamma \odot K_{XW,p-1}$
Gaussian	$\Phi = -X^T \mathcal{L}_\Psi X \ , \Psi = \Gamma \odot K_{XW}$
Multiquadratic	$\Phi = X^T \mathcal{L}_\Psi X \ , \Psi = \Gamma \odot K_{XW}^{(-1)}$

Table 2: Equations for Φs for the common kernels.

Sometimes the Scaled Covariance Matrix is a function of W itself. For these cases, we use the 2nd order Taylor Expansion to approximate the Scaled Covariance Matrix.

Here, notice that none of the Φ are functions of W.



Kernel	Approximation of Φs
Linear	$\Phi_0 = X^T \Gamma X$
Squared	$\Phi_0 = X^T \mathcal{L}_\Gamma X$
Polynomial	$\Phi_0 = X^T \Gamma X$
Gaussian	$\Phi_0 = -X^T \mathcal{L}_\Gamma X$
Multiquadratic	$\Phi_0 = X^T \mathcal{L}_\Gamma X$

Table 1: Equations for the approximate Φs for the common kernels.

Approximated Φ with Taylor Expansion.

By approximating Φ, we can initialize W and use this W to compute the next Φ. We can repeat this process until W converges.

This is the ISM algorithm.

$$\Phi_0 \rightarrow W_0 \rightarrow \Phi_1 \rightarrow W_1 \rightarrow \Phi_2 \rightarrow W_2 \rightarrow \ldots$$

We simply repeat this until W converges.

Although ISM look simple, the analysis required to guarantee its effectiveness was not simple !!!

Thm 1 :

Guarantees that the dominant eigenvector of Φ satisfies 1st and 2nd order conditions.

Thm 2 :

Guarantees that ISM algorithm converges to a subsequence.

Proposition 1 :

Any linear combination of ISM kernels is still a ISM kernel.

Thm 3 :

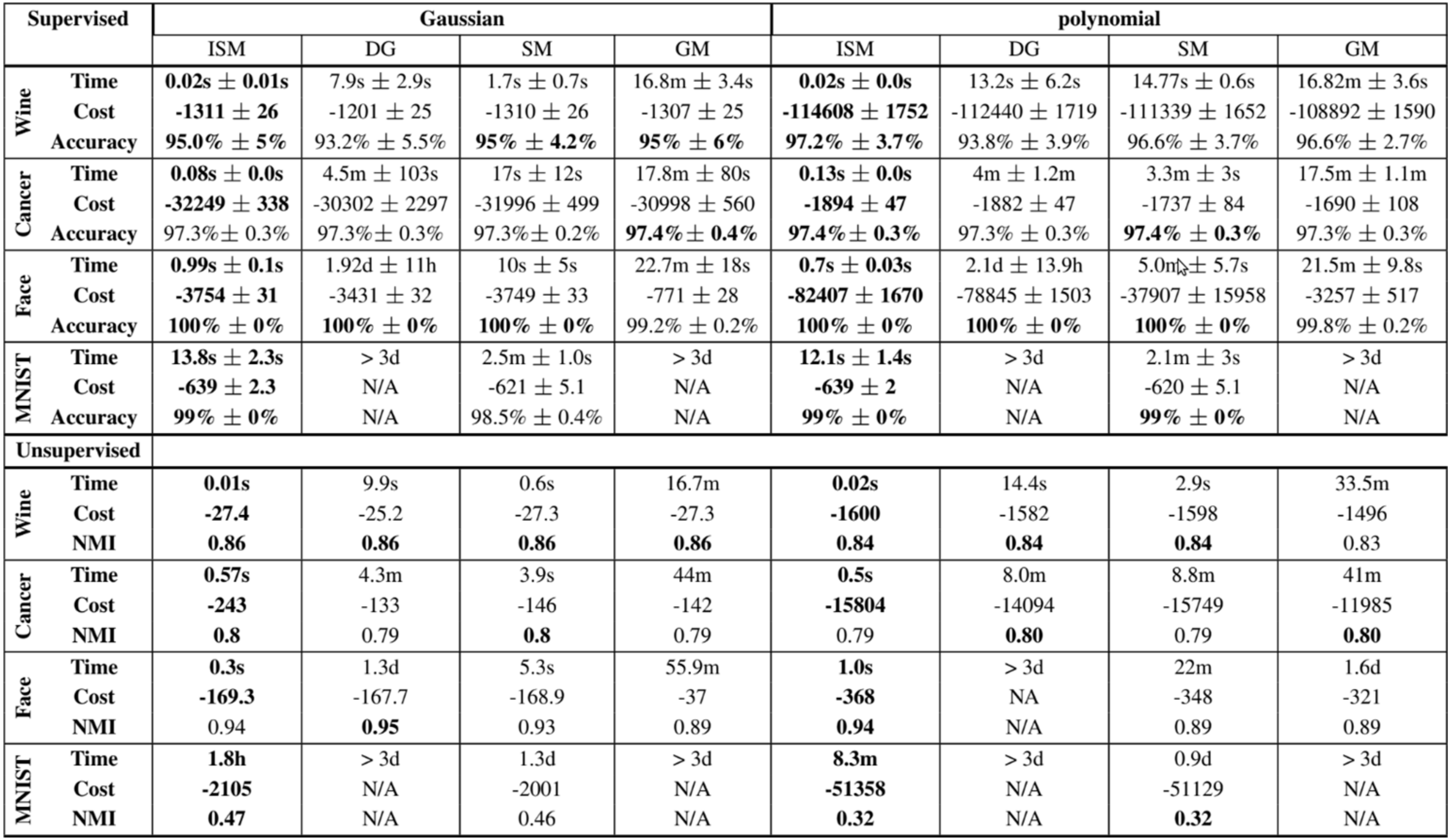
A ISM kernels can always obtain a Φ_o that's independent of W

Corollary 1 :

The Φ matrix of a conic combination of kernels $K_1, K_2, K_3, \ldots$

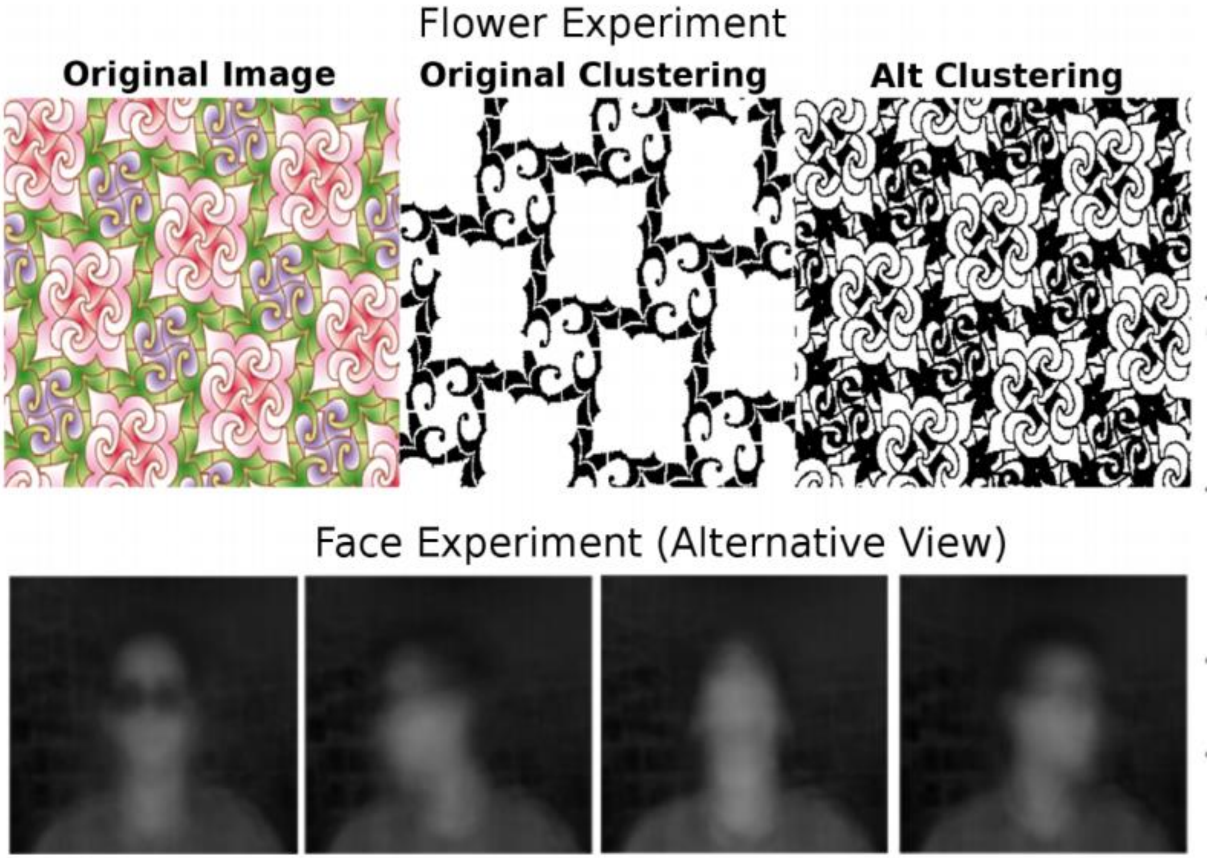
is equal to $\Phi = \Phi_1 + \Phi_2 + \Phi_3 + \ldots$

ISM solves many different IKDR problems.



Supervised		Gaussian				polynomial			
		ISM	DG	SM	GM	ISM	DG	SM	GM
Wine	Time	**0.02s ± 0.01s**	7.9s ± 2.9s	1.7s ± 0.7s	16.8m ± 3.4s	**0.02s ± 0.0s**	13.2s ± 6.2s	14.77s ± 0.6s	16.82m ± 3.6s
	Cost	**-1311 ± 26**	-1201 ± 25	-1310 ± 26	-1307 ± 25	**-114608 ± 1752**	-112440 ± 1719	-111339 ± 1652	-108892 ± 1590
	Accuracy	**95.0% ± 5%**	93.2% ± 5.5%	**95% ± 4.2%**	**95% ± 6%**	**97.2% ± 3.7%**	93.8% ± 3.9%	96.6% ± 3.7%	96.6% ± 2.7%
Cancer	Time	**0.08s ± 0.0s**	4.5m ± 103s	17s ± 12s	17.8m ± 80s	**0.13s ± 0.0s**	4m ± 1.2m	3.3m ± 3s	17.5m ± 1.1m
	Cost	**-32249 ± 338**	-30302 ± 2297	-31996 ± 499	-30998 ± 560	**-1894 ± 47**	-1882 ± 47	-1737 ± 84	-1690 ± 108
	Accuracy	97.3%± 0.3%	97.3%± 0.3%	97.3%± 0.2%	**97.4%± 0.4%**	**97.4%± 0.3%**	97.3% ± 0.3%	**97.4% ± 0.3%**	97.3% ± 0.3%
Face	Time	**0.99s ± 0.1s**	1.92d ± 11h	10s ± 5s	22.7m ± 18s	**0.7s ± 0.03s**	2.1d ± 13.9h	5.0m ± 5.7s	21.5m ± 9.8s
	Cost	**-3754 ± 31**	-3431 ± 32	-3749 ± 33	-771 ± 28	**-82407 ± 1670**	-78845 ± 1503	-37907 ± 15958	-3257 ± 517
	Accuracy	**100% ± 0%**	**100% ± 0%**	**100% ± 0%**	99.2% ± 0.2%	**100% ± 0%**	**100% ± 0%**	**100% ± 0%**	99.8% ± 0.2%
MNIST	Time	**13.8s ± 2.3s**	> 3d	2.5m ± 1.0s	> 3d	**12.1s ± 1.4s**	> 3d	2.1m ± 3s	> 3d
	Cost	**-639 ± 2.3**	N/A	-621 ± 5.1	N/A	**-639 ± 2**	N/A	-620 ± 5.1	N/A
	Accuracy	**99% ± 0%**	N/A	98.5% ± 0.4%	N/A	**99% ± 0%**	N/A	**99% ± 0%**	N/A
Unsupervised									
Wine	Time	**0.01s**	9.9s	0.6s	16.7m	**0.02s**	14.4s	2.9s	33.5m
	Cost	**-27.4**	-25.2	-27.3	-27.3	**-1600**	-1582	-1598	-1496
	NMI	**0.86**	**0.86**	**0.86**	**0.86**	**0.84**	**0.84**	**0.84**	0.83
Cancer	Time	**0.57s**	4.3m	3.9s	44m	**0.5s**	8.0m	8.8m	41m
	Cost	**-243**	-133	-146	-142	**-15804**	-14094	-15749	-11985
	NMI	**0.8**	0.79	**0.8**	0.79	0.79	**0.80**	0.79	**0.80**
Face	Time	**0.3s**	1.3d	5.3s	55.9m	**1.0s**	> 3d	22m	1.6d
	Cost	**-169.3**	-167.7	-168.9	-37	**-368**	NA	-348	-321
	NMI	0.94	**0.95**	0.93	0.89	**0.94**	N/A	0.89	0.89
MNIST	Time	**1.8h**	> 3d	1.3d	> 3d	**8.3m**	> 3d	0.9d	> 3d
	Cost	**-2105**	N/A	-2001	N/A	**-51358**	N/A	-51129	N/A
	NMI	**0.47**	N/A	0.46	N/A	**0.32**	N/A	**0.32**	N/A

ISM can also be used for Alternative Clustering...



Northeastern University

Come visit our poster at NIPS 2019 at :

Dec/8 - Dec/14

Poster : 4340

Vancouver Convention Centre , Vancouver Canada

Jennifer Dy
jdy@ece.neu.edu
Electrical & Computer Engineering Dept
Northeastern University

Chieh Wu
wu.chie@husky.neu.edu
Electrical & Computer Engineering Dept
Northeastern University