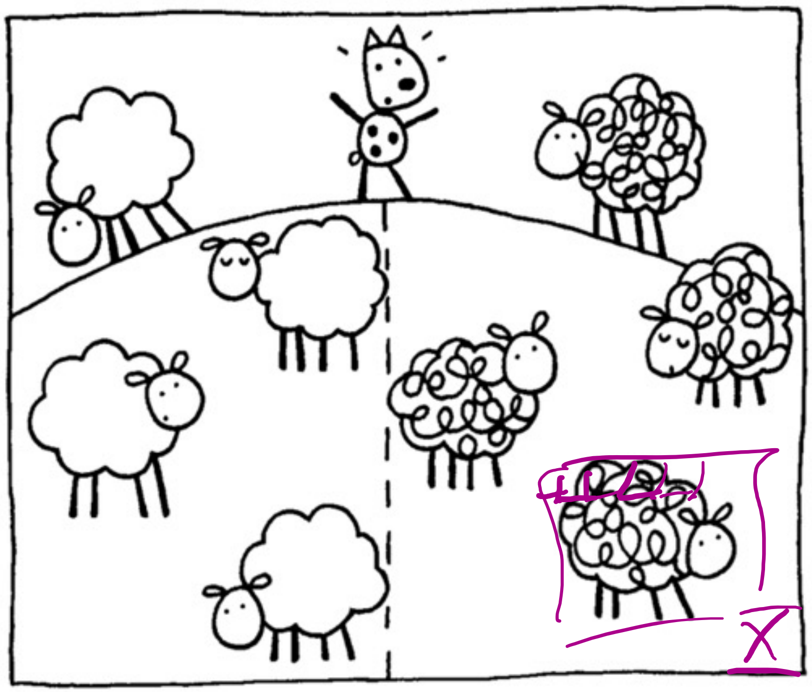
Support Vector Machines (SVM) Linear

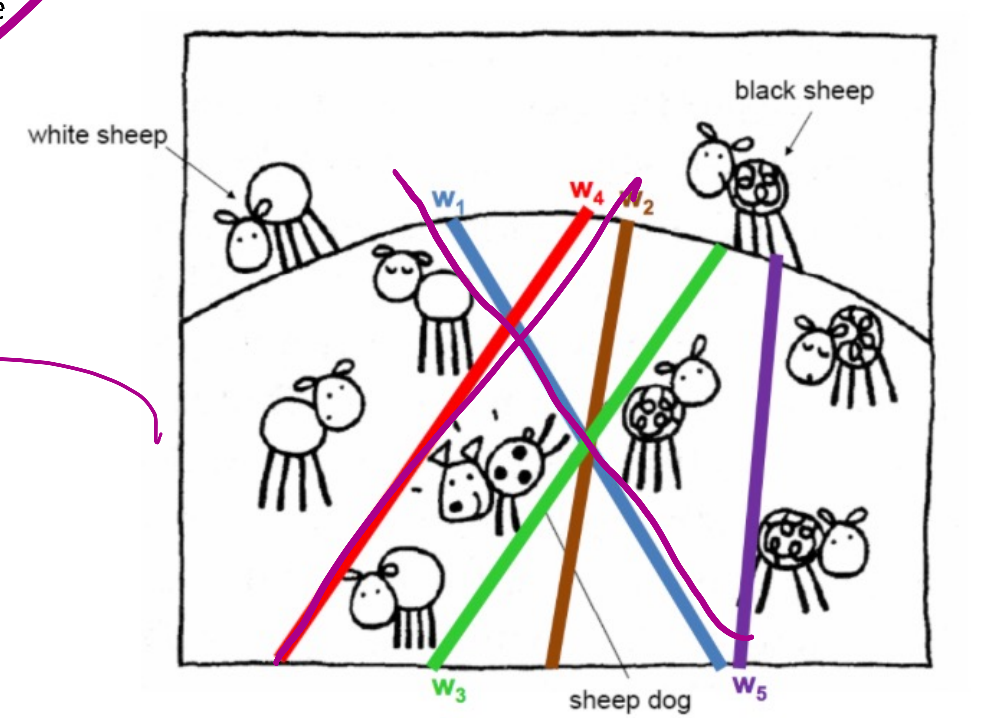
Smola - Schökopff.

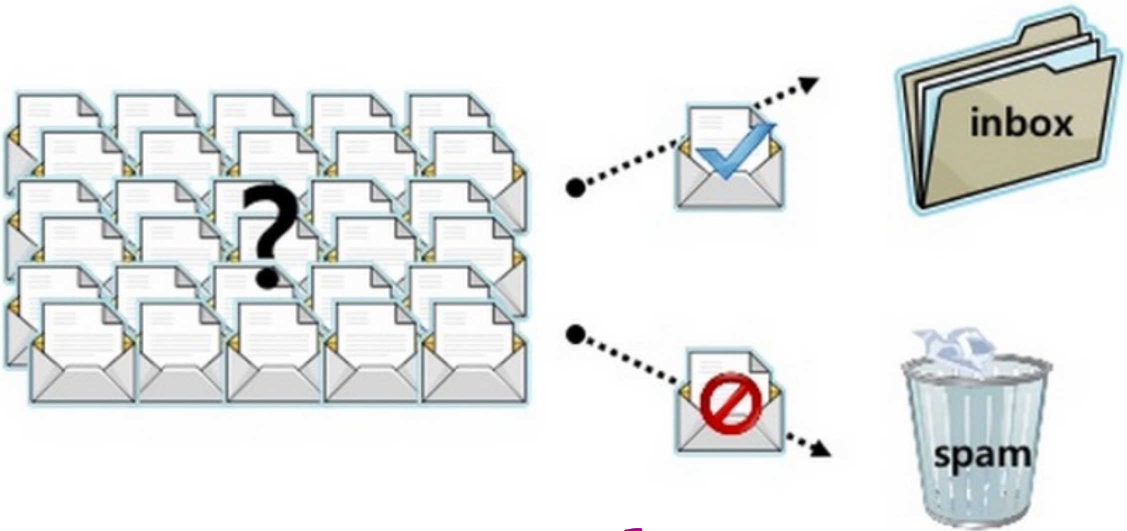
Then
Nonlinear (kernelized) SVM (KSVM)

Wide Margin Decision Hyperplane for
Supervised - Learning Classification

First a linear binary classification – decision boundary/hyperplane







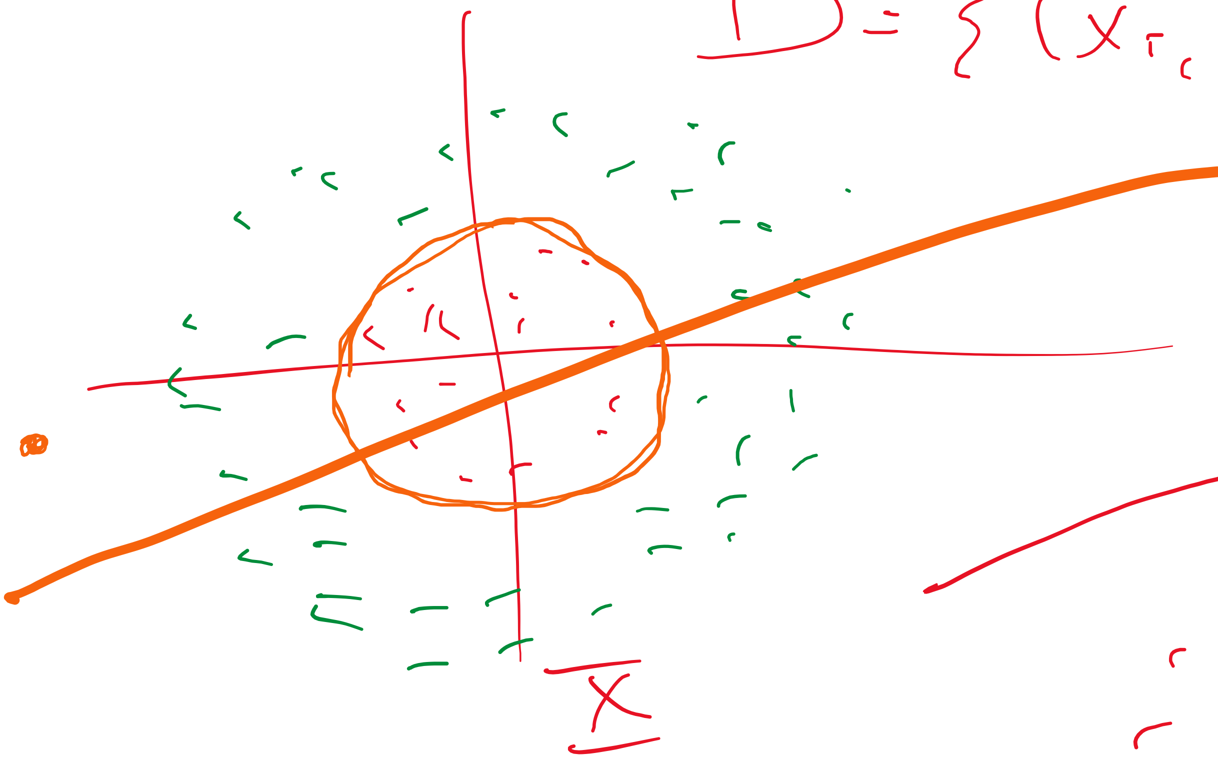
- Instance space: $x \in X$ ($|X| = n$ data points) $\{(x_i, y_i)\}_{i=1}^{n}$
  - Binary or real-valued feature vector x of word occurrences
  - d features (words + other things, d~100,000+)
- Class: $y \in Y$
  - y: Spam (+1), Ham (-1)

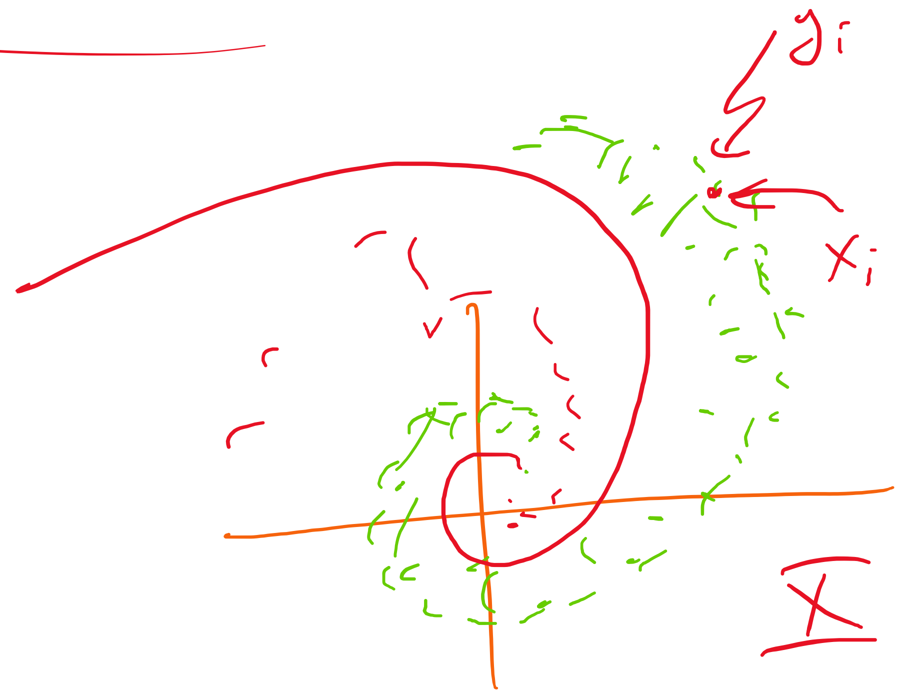
$y_i = \mathbb{Z}_g$

$g = 2$

Viagra	Learning	The	Dating	Nigeria	Is_spam
1	0	1	0	0	1
0	1	1	0	0	-1
0	0	0	0	1	1

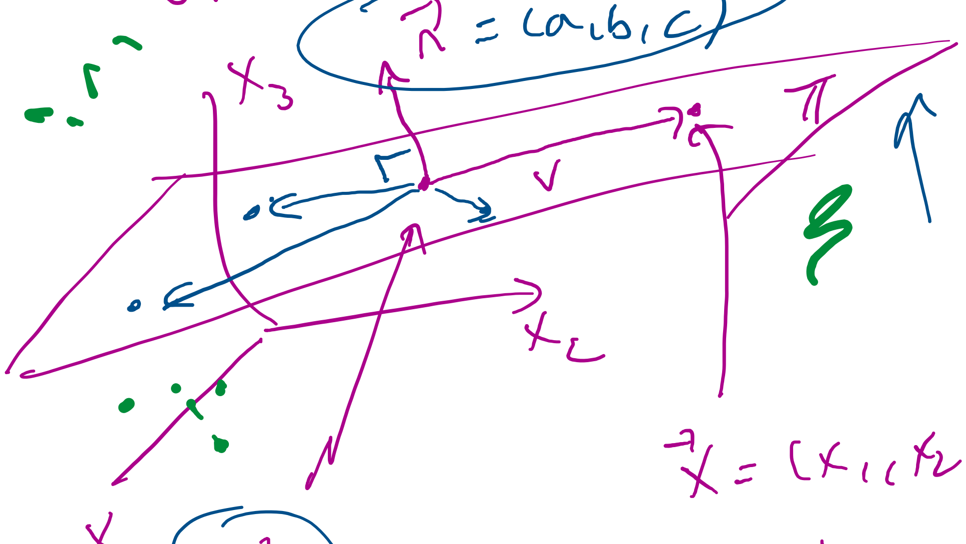
$D = \{(x_i, y_i)\}$





Review -- a hyperplane defined by a vector.

· 1 eq is a co-dim -1 restriction of space



$\vec{x} = (x_1, x_2, x_3)$

$\vec{x}_0 = (x_{1,0}, x_{2,0}, x_{3,0})$

$v = \vec{x} - \vec{x}_0 = \langle (x_1 - x_{1,0}), (x_2 - x_{2,0}), (x_3 - x_{3,0}) \rangle$

$\Pi := \{ x = (x_1, x_2, x_3) : \vec{v} = \cancel{\vec{x} - \vec{x}_0} \perp n \}$

$v \cdot n = 0 \iff v \perp \vec{n}$

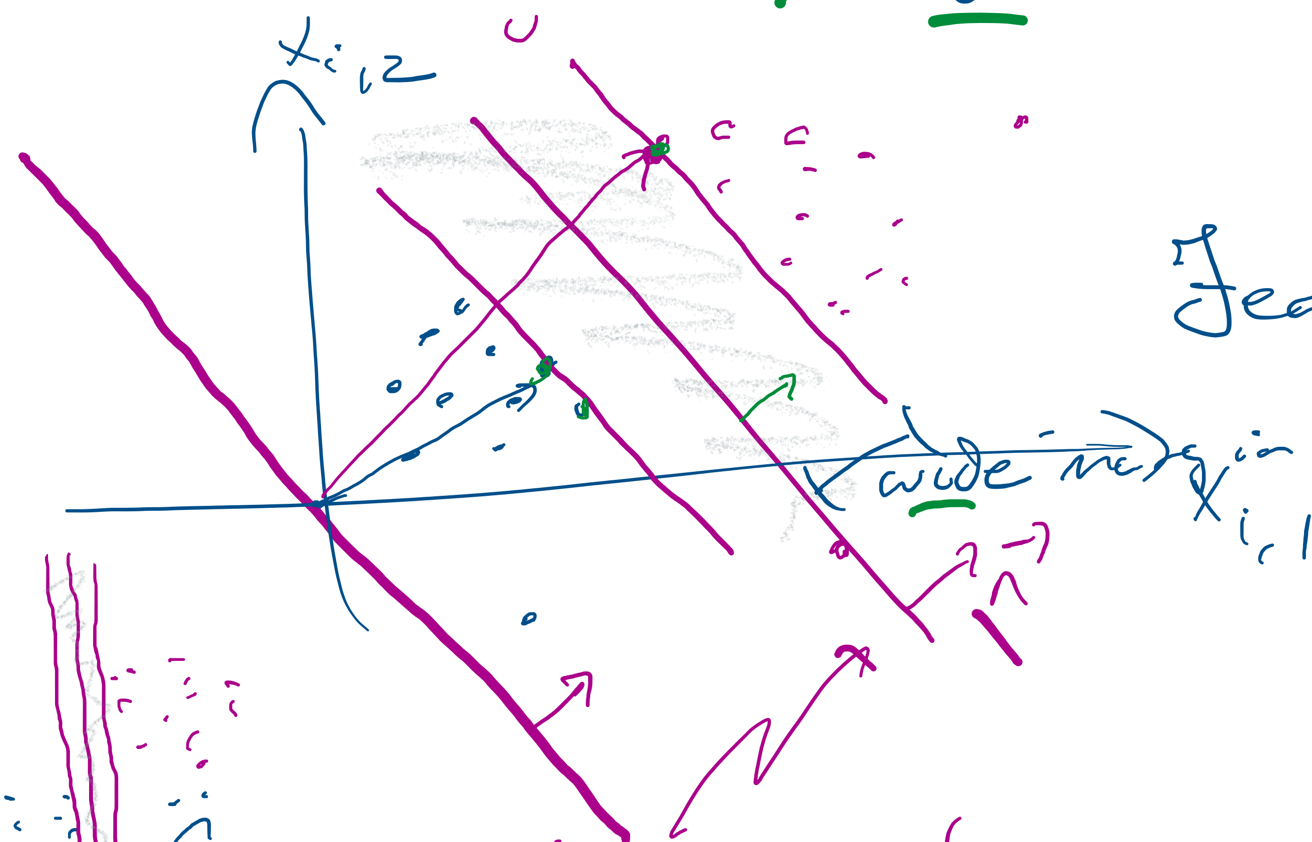
$n \cdot v = \langle a, b, c \rangle \cdot \langle x_1 - x_{1,0}, x_2 - x_{2,0}, x_3 - x_{3,0} \rangle = a(x_1 - x_{1,0}) + b(x_2 - x_{2,0}) + c(x_3 - x_{3,0}) = 0$

SVM: $D = \{(x_i, y_i)\}_{i=1}^{n}$

$x_i \in \mathbb{R}^d$

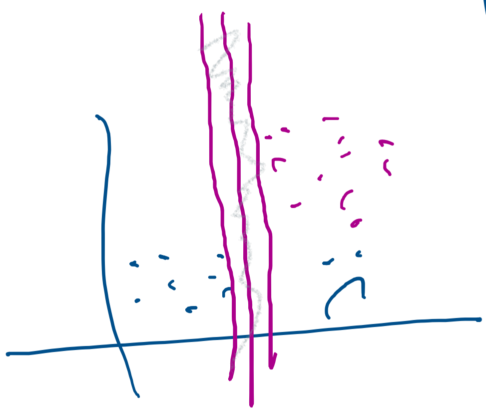
$y_i \in \mathbb{Z}_2 = \{-1, 1\}$

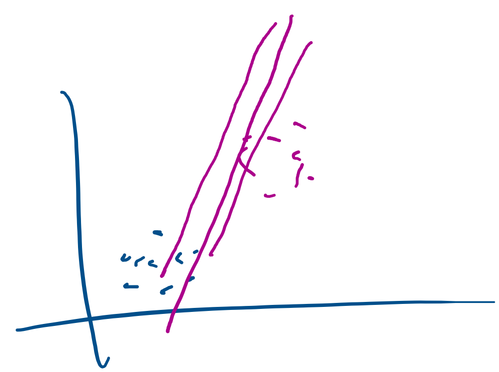
feature space

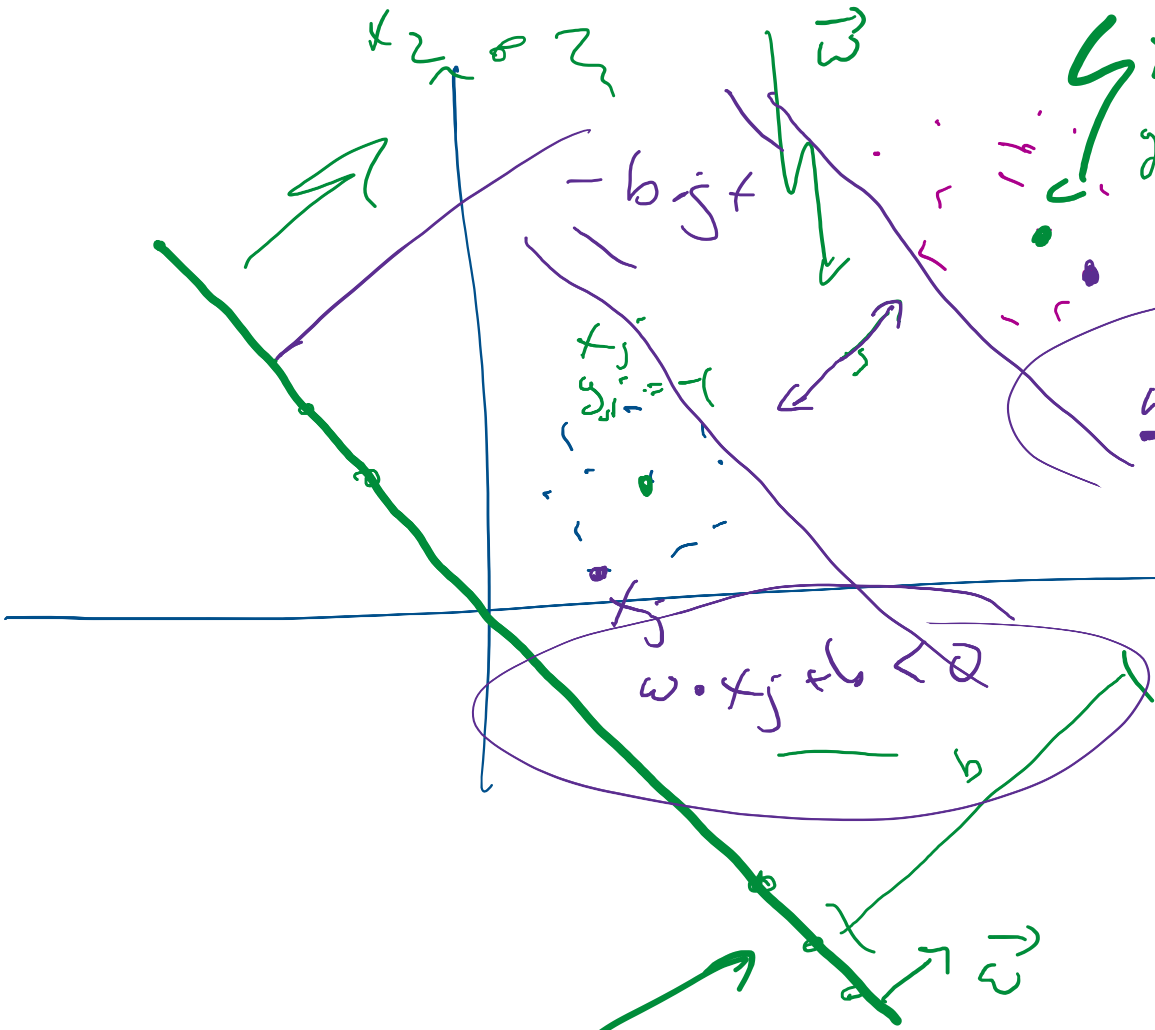


wide margin as big possible

hyper-plane







$\{x_i$
$y_i = 1$

find $\vec{w}$ & b
to optimize
"wide margin"

$\vec{w} \cdot x_i + b > 0$

x_1

$\vec{w} \cdot \vec{x} + b = 0$

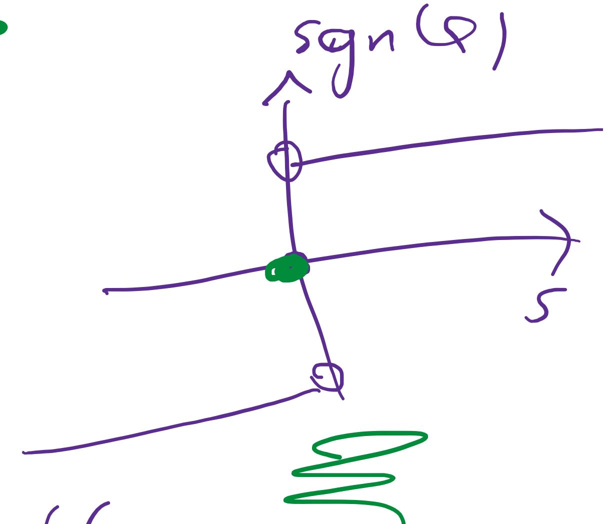
$\vec{w} \cdot x = 0$

$(\vec{x}_i, y_i)$, $y_i = -1$, 1

($y_i = 0$ or 1
apple or orange
dog or cat.

$y_j(w \cdot x_j + b) = \text{sgn}(w \cdot x_j + b)$ good label.

$$\text{sgn}(s) = \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{if } s < 0 \end{cases}$$



$\text{sgn}((s) \; w \cdot x_j + b) = 1$ labelled well.

$= -1$ mislabelled.

a loss function.

$$\ell(y_i, \bar{y}_i) = \ell(y_i, \underbrace{\mathrm{sgn}(w \cdot \vec{x}_i + b)}) = \begin{cases} 0 & \text{if correct label } y_i = \mathrm{sgn}(w \cdot x_i + b) \\ 1 & \text{incorrect label} \end{cases}$$

$\bar{y}_i$

label you infer from $\vec{x}_i$ alone if you have a good hyperplane $\vec{w}$ & b

Total loss

$$\sum_{i=1}^{N} \ell(y_i, \bar{y}_i)$$

Cost : ... ~~loss~~

$w \cdot x_j$

$$\operatorname{argmin} \frac{1}{2}\|w\|_2^2 \quad \text{subj} \quad y_i(w^T x_i - b) - 1 = 0$$

small $\|w\|$

subj every matches truth

dist between

dist big

$$\frac{2}{\|w\|_2^2} \text{ big} \approx \frac{\|w\|_2^2}{2} \text{ small}$$

constrained opt.

$(\vec{w}, \underline{b})$ $\leftarrow d+1=3$ $\quad \vec{w} \in \mathbb{R}^d,\ d=2$

$$\mathcal{L}(\underline{x}, \underline{s}, \underline{\theta}) = \underbrace{\frac{1}{2}\|w\|_2^2}_{\frac{1}{2} w \cdot w} - \sum_{i=1}^{n} s_i \left(y_i (w^T x_i - b) - 1 \right)$$

$\{(\underline{x_i}, \underline{y_i})\}$

$y_1 (w^T x_1 - b) - 1 = 0$

$y_2 (w^T x_2 - b) - 1 = 0$

$\vdots$

$y_n (w^T x_n - b) - 1 = 0$

$$= \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n} s_i y_i (w^T \vec{x}_i - b) \leftarrow \sum_{i=1}^{n} s_i$$

$\nabla_\theta \mathcal{L} = \vec{0}$

$\nabla_w \mathcal{L}(x, s, \theta) = w - \sum s_i y_i \vec{x}_i = \vec{0} \leftarrow d$ -eqns for d features

$\nabla_b \mathcal{L} = \frac{\partial}{\partial b} \mathcal{L} = \sum s_i y_i = 0$

$$w = \sum_{i=1}^{n} s_i y_i x_i \; ; \; \sum_{i=1}^{n} s_i y_i = \vec{s} \cdot \vec{y} = 0$$

$x_i \in \mathbb{R}^d$ $\quad$ $s \in \mathbb{R}^n$ $\quad$ $\in \mathbb{R}$

n-values.

"Trick" - KKT - Primal Dual form.

Primal Form. $\}$ $\min_{\theta, s} \mathcal{L}(x, s, \theta) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n} s_i y_i (w^T x_i + b)$

Dual Form

$$\max \; \mathcal{L}_D(x, s, \theta) = \sum s_i - \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} s_i s_j y_i y_j \boxed{(x_i \cdot x_j)}$$

$K(x_i, x_j)$

$\phi(x_i) \cdot \phi(x_j)$

s.t. $w = \sum_{i=1}^{n} s_i y_i x_i$ and

$\sum_{i=1}^{n} s_i y_i = 0$

KKT

Primal Problem:

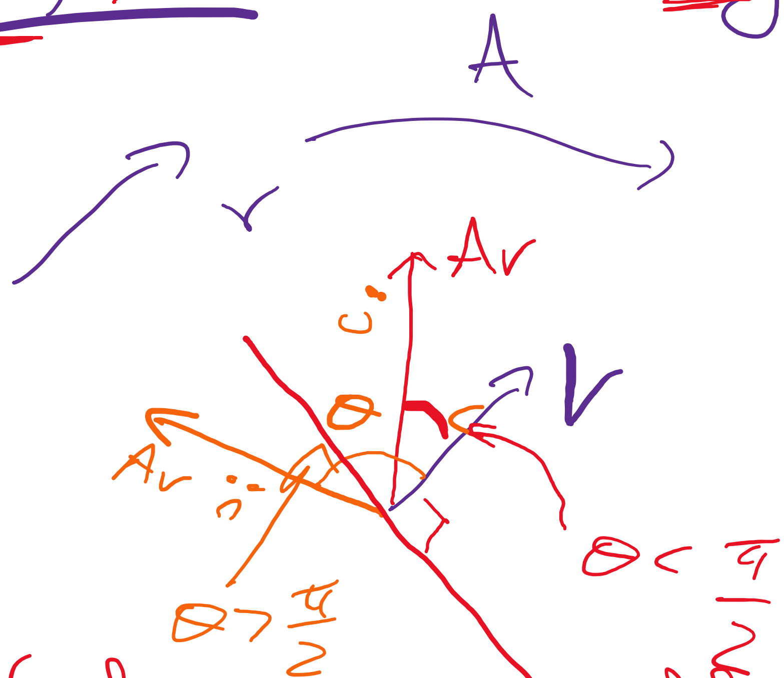
$$\begin{cases} \text{minimize: } \mathcal{L}(x,s) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} s_i y_i (w^T x_i + b) + \sum_{i=1}^{n} s_i \\ \text{such that: } s_i \geq 0, \forall i \end{cases}$$

Dual Problem:

$$\begin{cases} \text{maximize: } \mathcal{L}_D(x,s) = \sum_{i=1}^{n} s_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} s_i s_j y_i y_j (\vec{x}_i^T \vec{x}_j) \\ \text{using: } w = \sum_{i=1}^{n} s_i y_i x_i \text{, and } \sum_{i=1}^{n} s_i y_i = 0 \end{cases}$$

Pos. Semi-Defn Definitions -

- A matrix $A_{n\times m}$ is semi positive definite if $v\cdot(A\cdot v) \geq 0$ for any $v \neq 0$ in domain of A.



$$v\cdot(Ar) = \|v\|\,\|Ar\|\cos\theta$$

- A kernel fn is pos. semi definite if for $x_1, x_2, x_3,$ and $\bar{K}$ pos. semi def matrix for any input set

- $\mathcal{H}$ = Hilbert space – a complete inner product space

vector space

- a Hilbert space is a set $\mathcal{H}$ of vectors such that there is a complete inner product.

limits work properly.

dot

Spectral Decomp. thm:

Suppose A $n \times n$ is pos. def'n. symm. matrix with eigenvectors & eigenvalues

λ_i, v_i, $\quad 0 < \lambda_1 \le \lambda_2 \le \dots \le \lambda_n$

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

not $v_i^T v_i = v_i \cdot v_i$

$v_i \otimes v_i = P_i$ rank-1 projector

$\in \underline{\mathcal{X}} = \mathbb{R}^2$

- $x_i \cdot x_j$ need the dot product between $\underline{x_i}$ & $\underline{x_j}$ to do SVM.

- $\underline{K(x_i, x_j)} = \phi(x_i) \cdot \phi(x_j)$

corresponds.

$\phi : \underline{\mathcal{X}} \to \mathcal{H}$ $\in \mathbb{R}^6$

Gram matrix. - symmetric

$$\underline{K} = \begin{pmatrix} \underline{K(x_1, x_2)} & K(x_2, x_3) & \cdots & \underline{K(x_1, x_n)} \\ \vdots & & & \\ \underline{K(x_n, x_1)} & \cdots & & K(x_n, x_n) \end{pmatrix}$$

The amazing Kernel trick – nonlinear SVM through a kernel and all dot products in the high dimensional space
Done through a kerekl function

0 $\mathbb{X} = \mathbb{R}^2$

Now, we define a kernel $K : X \times X \mapsto \mathbb{R}$, which can take different forms such as:

- Linear kernel: $K(x, \tilde{x}) = x^T \tilde{x}$.
- Polynomial kernel: $K(x, \tilde{x}) = (x^T \tilde{x} - y)^d$. +1
- Gaussian RBF: $K(x, \tilde{x}) = e^{-\frac{Vertx - \tilde{x}\|^2}{2\sigma^2}}$ $e^{-\frac{\|x-\tilde{x}\|}{2\sigma^2}}$

Kernel is

- 0
- 1 symmetric
- 2 pos. semi-defn
- 3 cts. w.r.t. both inputs.

Consider the polynomial kernel, for $d = 2$, $X = \mathbb{R}^2$, then we have:

$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix}$

$$
\begin{aligned}
K(x, \tilde{x}) &= (x \cdot \tilde{x} + 1)^d \\
&= (x^T \tilde{x} + 1)^d \\
&= (x_1 \tilde{x}_1 + x_2 \tilde{x}_2 + 1)^2 \\
&= x_1^2 \tilde{x}_1^2 + 2x_1 \tilde{x}_1 + 2x_2 \tilde{x}_2 + x_1 \tilde{x}_1 x_2 \tilde{x}_2 + 1 \quad + x_2^2 \tilde{x}_2^2
\end{aligned}
$$

$K(x, \tilde{x}) = K(\tilde{x}, x)$

which interestingly can be re-written in terms of dot product:

$d \uparrow \quad D \uparrow\uparrow$ maybe

$D = \infty$

$$
\begin{aligned}
K(x,\tilde{x}) &= (x\cdot\tilde{x}+1)^d \\
&= (x^T\tilde{x}+1)^d \\
&= (x_1\tilde{x}_1+x_2\tilde{x}_2+1)^2 \\
&= x_1^2\tilde{x}_1^2+2x_1\tilde{x}_1+2x_2\tilde{x}_2+x_1\tilde{x}_1x_2\tilde{x}_2+1 \quad + x_2^2\tilde{x}_2^2
\end{aligned}
$$

which interestingly can be re-written in terms of dot product:

$$K(x,\tilde{x}) = (1,\sqrt{2}x_1,\sqrt{2}x_2,x_1^2,x_1x_2,x_2^2)\cdot(1,\sqrt{2}\tilde{x}_1,\sqrt{2}\tilde{x}_2,\tilde{x}_1^2,\tilde{x}_1\tilde{x}_2,\tilde{x}_2^2)$$

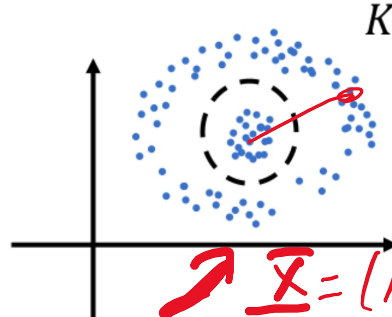
$n \qquad n-1$

$\phi:\mathbb{R}^2\to\mathbb{R}^6$

$K:\mathbb{R}^2\times\mathbb{R}^2\to\mathbb{R}$

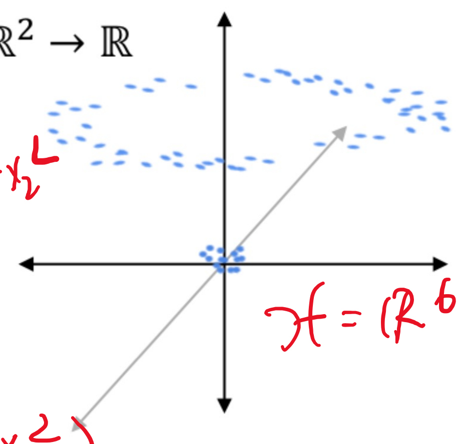
$$K(x,\tilde{x}) = (\phi(x)\cdot\phi(\tilde{x}))$$

$\mathbb{R}^2 \qquad \mathbb{R} \qquad C \qquad D$



$r = x_1^2 + x_2^2$

$\mathcal{X} = \mathbb{R}^2$



$\mathcal{H} = \mathbb{R}^6$

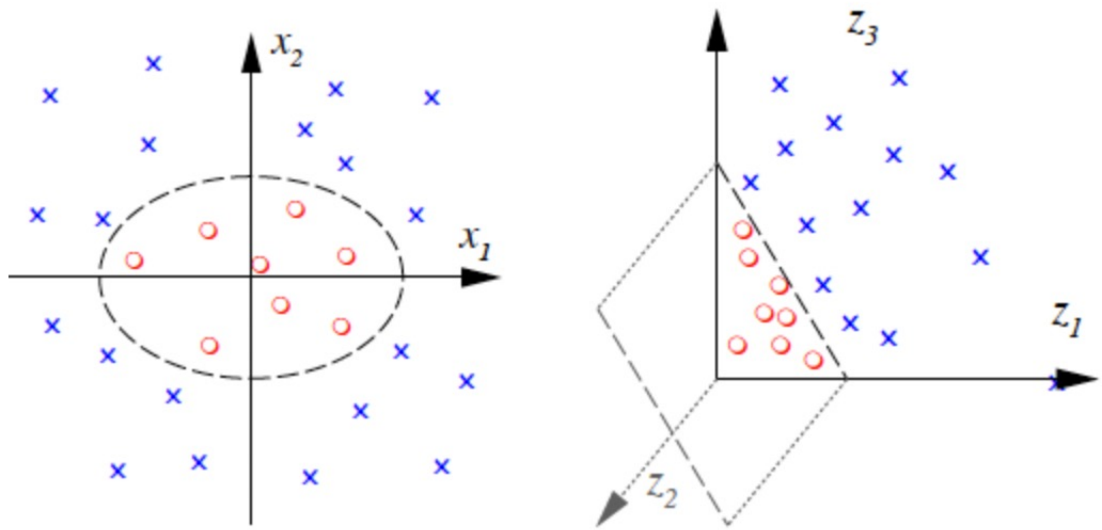
$\phi : \mathcal{X} \longrightarrow \mathcal{H}$

$x \longmapsto \phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_1 x_2, x_2^2)$

Hyper Plane Classifier in Feature Space

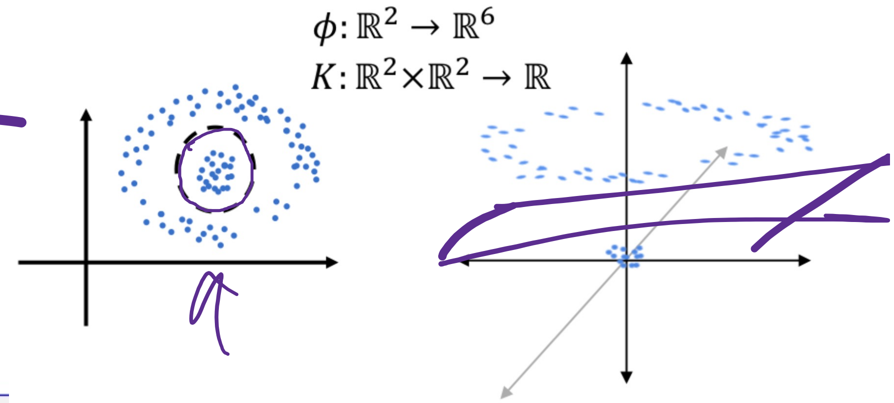
$$\Phi : R^2 \rightarrow R^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt(2)x_1x_2, x_2^2)$$



$$\phi(x_1, x_2) = (\phi_1(x_1, x_2), \phi_2(x_1, x_2), ..., \phi_6(x_1, x_2))$$

where $\phi : X \mapsto \mathcal{H}$.



Note that $X = \mathbb{R}^2$ is the domain, and $\mathcal{H}$ is the Hilbert space, which is (in machine learning literature) the feature space, and a set of features $\phi_i, \forall i$, is called dictionary.

Mantra

A major theme in machine learning is that sometimes things actually get easier in higher dimensions !!!.

- A linear plane in high dimensional feature space $\mathcal{H}$, may be a nonlinear curves in the domain space.
- $\mathcal{H}$ is a plane, with calculus with dot products is legit.

The following, we introduce Mercer's theorem, which generalizes spectral decomposition theorem.

$K : X \times X \to \mathbb{R}$

Theorem 5.5.1 — Mercer's Theorem — generalizes spectral decomp thm.

. Let $K \in L^2(X \times X)$, (i.e. $\int |K(x,\tilde{x})|^2 dx d\tilde{x} < \infty$) such that $T : L_2(X) \mapsto L_2(X)$ by $(T(f)(x)) = \int K(x,\tilde{x}) f(\tilde{x}) d\tilde{x}$ is positive definite. If $\phi_i \in L^2(X)$ is a normalized eigenfunction with eigenvalues $0 < \lambda_1, \leq \lambda_2 \leq \cdots \leq \lambda_N$, Then

$$K(x,\tilde{x}) = \sum_{i=1}^{N_{\mathcal{H}}} \lambda_i \phi_i(x) \phi_i(\tilde{x}) \tag{5.20}$$

$T(\phi_i)(x) = \lambda_i \phi_i(x)$ ⟸ eigen fn.

for almost every $(x,\tilde{x})$. Where $N_{\mathcal{H}} = dim(\mathcal{H})$, and the convergance of $K(x,\tilde{x})$ is absolute.

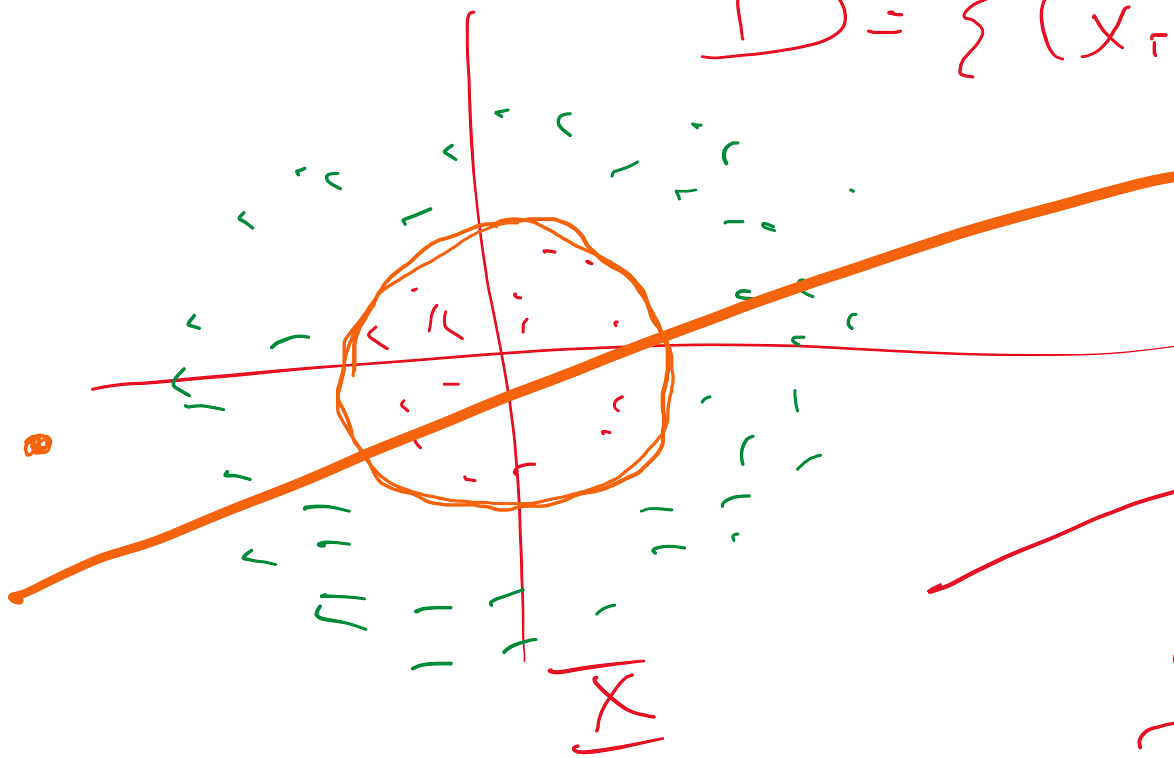
usual $Av = U$

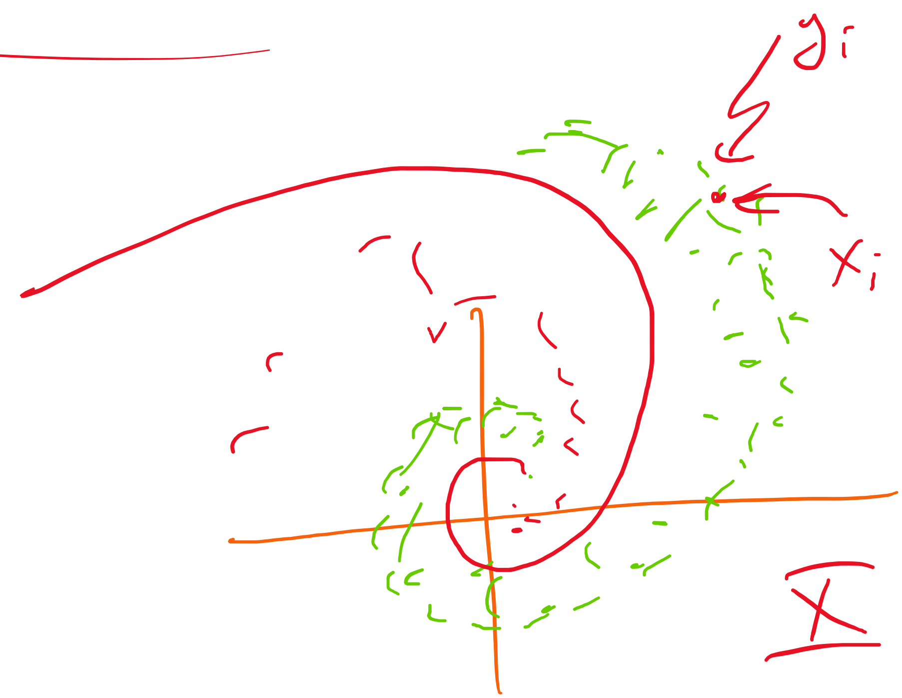
. e eigen $Aw = \lambda w$

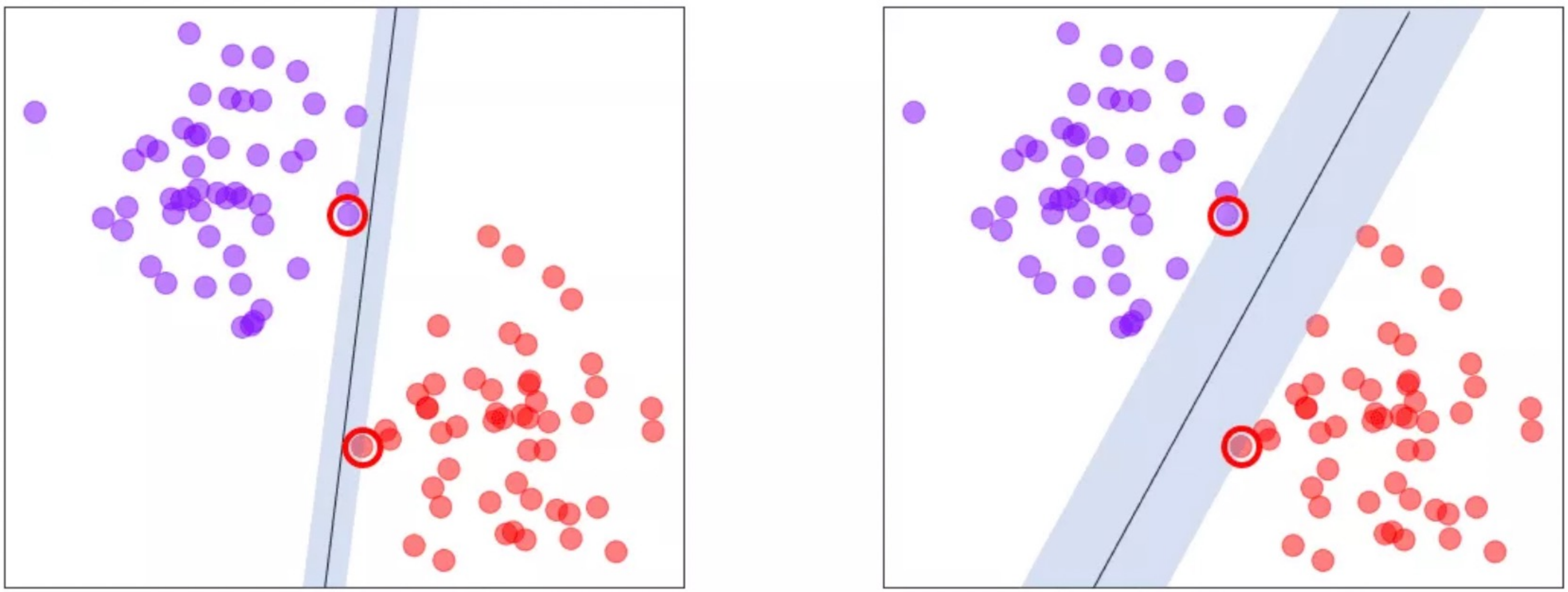
Mercer's theorem itself is a generalization of the result that any symmetric positive-semidefinite matrix is the Gramian matrix of a set of vectors.

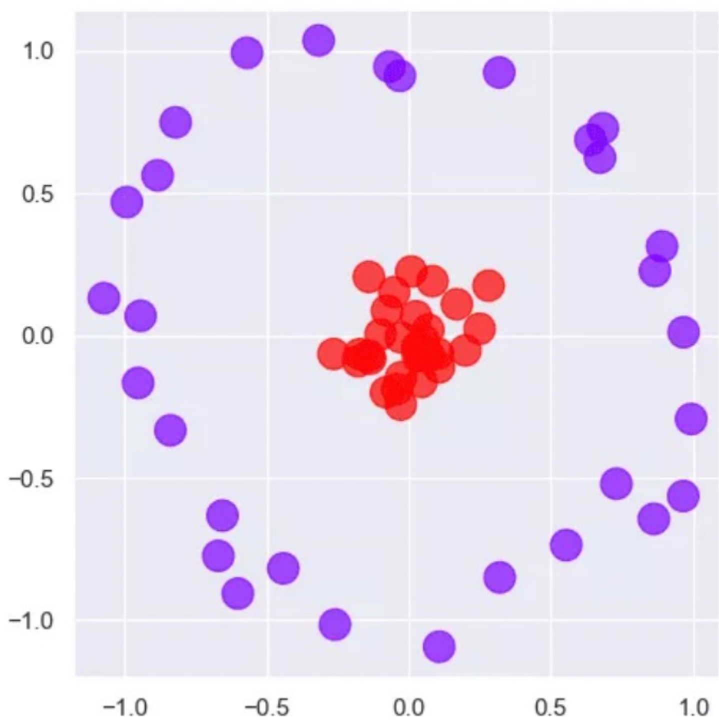
ϕ_i's exist ∃ I can use them in KSVM.

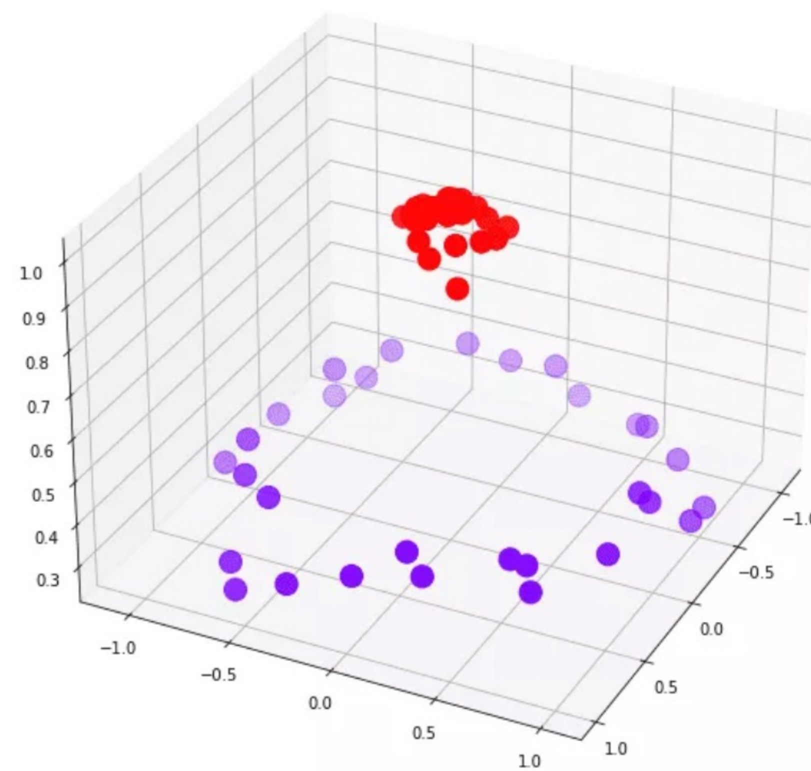
$$D = \{(x_i, y_i)\}$$

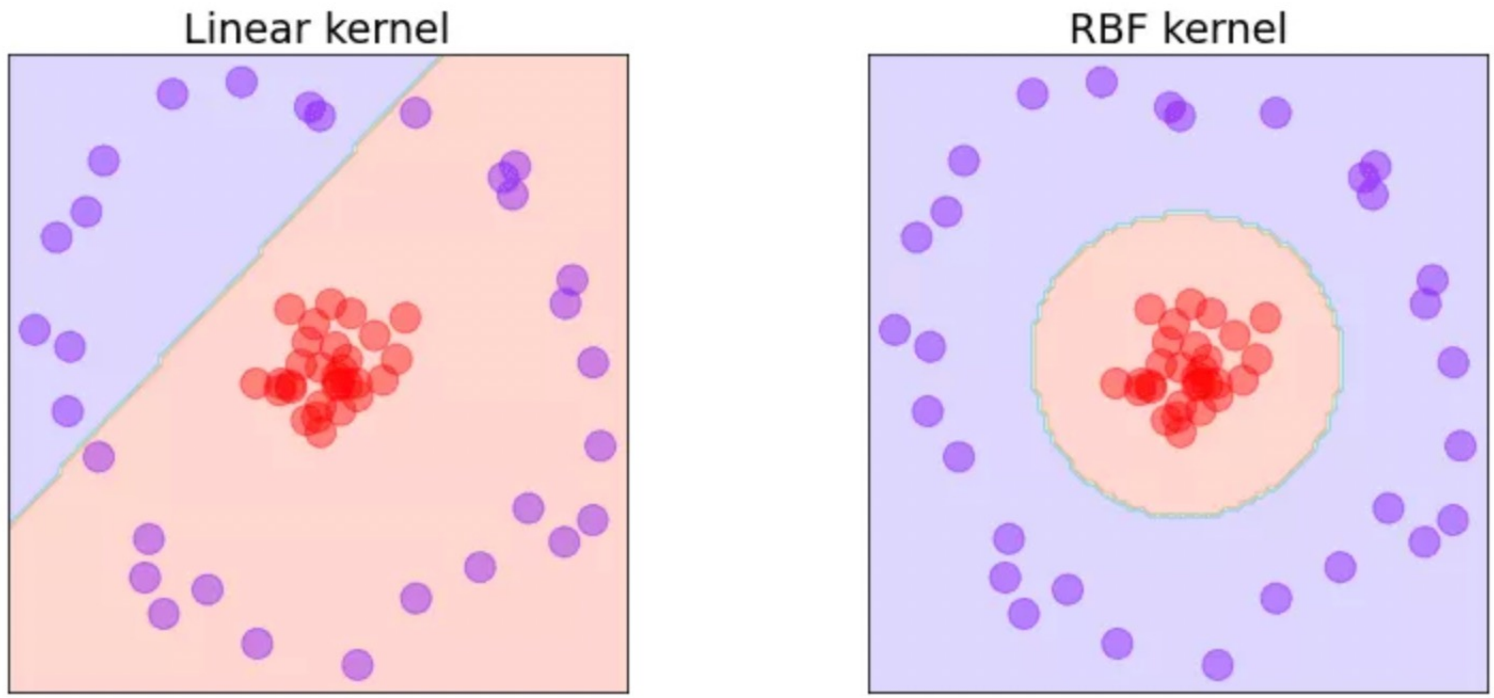


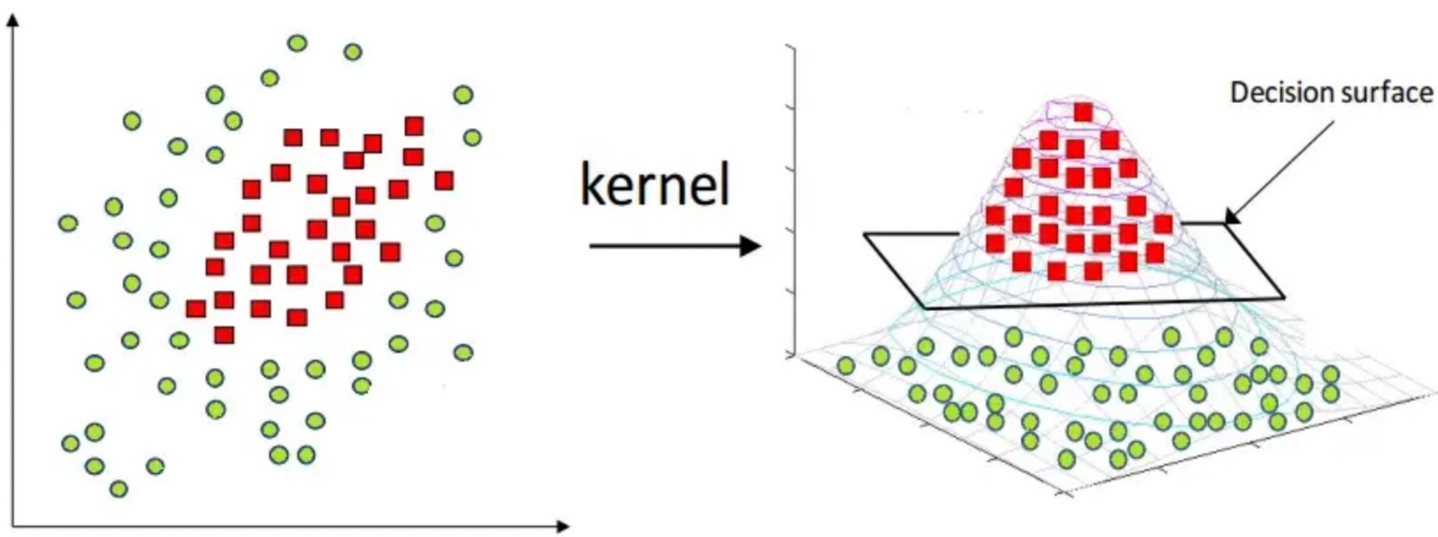


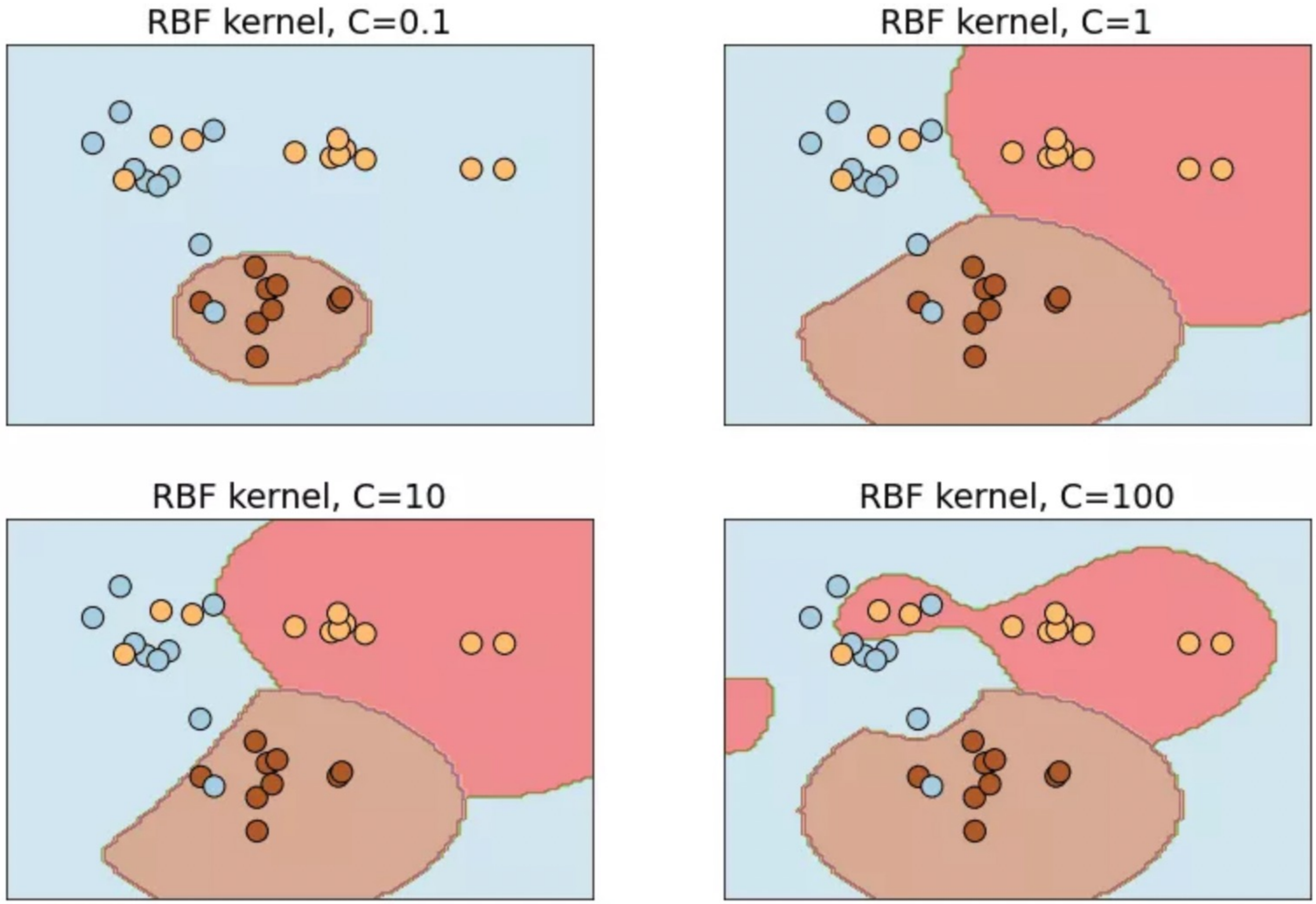



1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

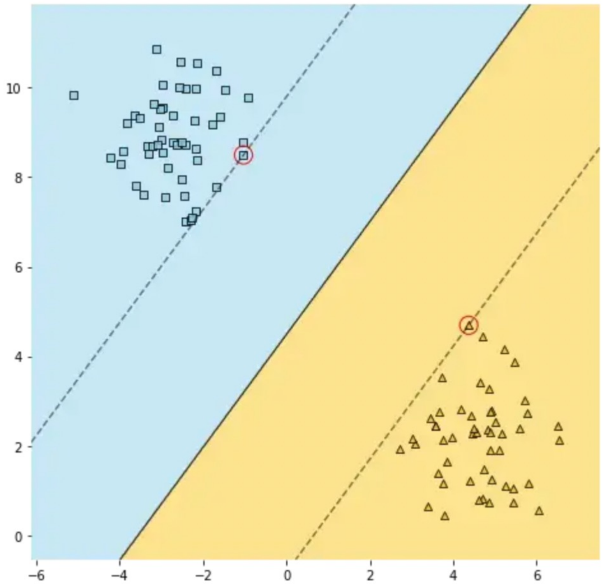

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
−1.0
−0.5
0.0
0.5
1.0
−1.0
−0.5
0.0
0.5
1.0


Linear kernel
RBF kernel

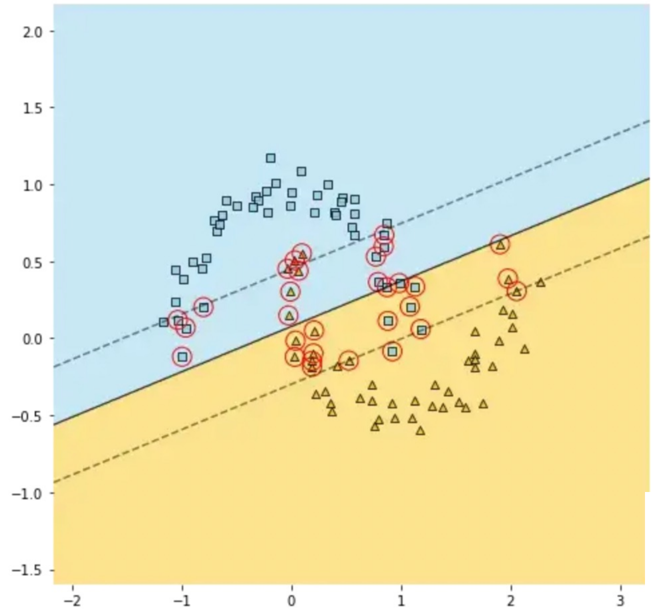

kernel
Decision surface


RBF kernel, C=0.1
RBF kernel, C=1
RBF kernel, C=10
RBF kernel, C=100

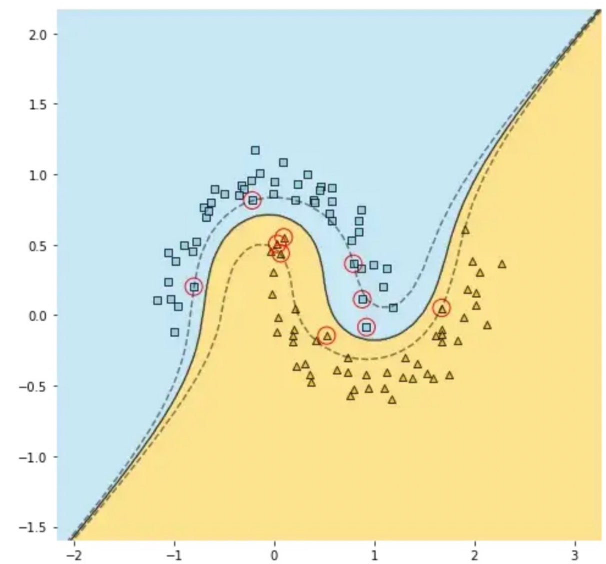

George W Bush
Gerhard Schroeder
Donald Rumsfeld
Tony Blair
Donald Rumsfeld
Colin Powell
George W Bush
Colin Powell



Linear SVM with linearly separable data works pretty well.



Linear SVM with linearly non-separable data does not work at all.



Decision boundary with a polynomial kernel.

Theorem 2 (Mercer 1909). *Suppose $k \in L_\infty(\mathcal{X}^2)$ such that the integral operator $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$,*

$$T_k f(\cdot) := \int_{\mathcal{X}} k(\cdot, x) f(x) d\mu(x) \tag{20}$$

is positive (here μ denotes a measure on $\mathcal{X}$ with $\mu(\mathcal{X})$ finite and $\mathrm{supp}(\mu) = \mathcal{X}$*). Let $\psi_j \in L_2(\mathcal{X})$ be the eigenfunction of T_k associated with the eigenvalue $\lambda_j \neq 0$ and normalized such that $\|\psi_j\|_{L_2} = 1$ and let $\overline{\psi_j}$ denote its complex conjugate. Then*

1. $(\lambda_j(T))_j \in \ell_1$.
2. $k(x, x') = \sum_{j \in \mathbb{N}} \lambda_j \overline{\psi_j(x)} \psi_j(x')$ *holds for almost all* (x, x'), *where the series converges absolutely and uniformly for almost all* (x, x').

Let's take it for granted that this is a valid positive semidefinite kernel. Let $k_{\texttt{poly(r)}}$ denote a polynomial kernel of degree r, and let $\gamma = 1/2$. Then

$$\begin{aligned}
k_{\text{RBF}}(\mathbf{x}, \mathbf{y}) &= \exp\Big(-\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2\Big) \\
&= \exp\Big(-\frac{1}{2}\langle \mathbf{x}-\mathbf{y}, \mathbf{x}-\mathbf{y}\rangle\Big) \\
&\stackrel{\star}{=} \exp\Big(-\frac{1}{2}\Big[\langle \mathbf{x}, \mathbf{x}-\mathbf{y}\rangle - \langle \mathbf{y}, \mathbf{x}-\mathbf{y}\rangle\Big]\Big) \\
&\stackrel{\star}{=} \exp\Big(-\frac{1}{2}\Big[\langle \mathbf{x}, \mathbf{x}\rangle - \langle \mathbf{x}, \mathbf{y}\rangle - \big[\langle \mathbf{y}, \mathbf{x}\rangle - \langle \mathbf{y}, \mathbf{y}\rangle\big]\rangle\Big]\Big) \\
&= \exp\Big(-\frac{1}{2}\Big[\langle \mathbf{x}, \mathbf{x}\rangle + \langle \mathbf{y}, \mathbf{y}\rangle - 2\langle \mathbf{x}, \mathbf{y}\rangle\Big]\Big) \\
&= \exp\Big(-\frac{1}{2}\|\mathbf{x}\|^2\Big)\exp\Big(-\frac{1}{2}\|\mathbf{y}\|^2\Big)\exp\Big(-2\langle \mathbf{x}, \mathbf{y}\rangle\Big)
\end{aligned}$$

Above, the two steps labeled $\star$ leverage the fact that

$$\langle \mathbf{u}+\mathbf{v}, \mathbf{w}\rangle = \langle \mathbf{u}, \mathbf{w}\rangle + \langle \mathbf{v}, \mathbf{w}\rangle$$

in general for inner products (see here){:target="_blank"}. Now let C be a constant,

$$C \equiv \exp\Big(-\frac{1}{2}\|\mathbf{x}\|^2\Big)\exp\Big(-\frac{1}{2}\|\mathbf{y}\|^2\Big).$$

and note that the Taylor expansion of $e^{f(x)}$ is

$$e^{f(x)} = \sum_{r=0}^{\infty}\frac{[f(x)]^r}{r!}.$$

We can write the RBF kernel as

$$\begin{aligned}
k_{\text{RBF}}(\mathbf{x}, \mathbf{y}) &= C\exp\big(-2\langle \mathbf{x}, \mathbf{y}\rangle\big) \\
&= C\sum_{r=0}^{\infty}\frac{\langle \mathbf{x}, \mathbf{y}\rangle^r}{r!} \\
&= C\sum_{r}^{\infty}\frac{k_{\texttt{poly(r)}}(\mathbf{x}, \mathbf{y})}{r!}.
\end{aligned}$$

So the RBF kernel can be viewed as an infinite sum over polynomial kernels. As r increases, each polynomial kernel lifts the data into higher dimensions, and the RBF kernel is an infinite sum over these

SVD/POD/PCA/KL – unsupervised – ROM – structure of data (shape/geometry of data)– manifold learn – given x_i

DMD – supervised – forecasting – ROM – spectral analysis – structure of the process features are important, given x_i -> x_i,x_i+1. mght as well call the x_i+1=y_i – regression

Regression – onto general basis sets – supervised – find y=f(x) given examples of (x_i,y_i)

Neural nets – classification of handwriting digits USPS – supervised,
forecasting also regression to the flow function for forecasting
auto-encoder is a unsupervised algoritghm using ANN with a bottleneck. – ROM
random version was reservoir computing

Kmeans – clustering (given x develop labels – as y) – partitioning the data –

LDA – linear discriminant analysis – Fischer 1936 – classification – given labeled data xI with labels yi learn y=f(x)

SVM – linear method for classification supervised – support vector machine
kernelized version is nonlinear – reproducing kernel Hilbert space – KSVM – KSVD

Manifold learning – unsupervised – structure of the data – POD, autoencoder, ISOMAP, Diffusion Map

Regression, and classification – supervised