统计机器学习

（小班研讨）

主讲教师： 刘峤

第4章 支持向量机与核方法

Support Vector Machines and Kernel Methods

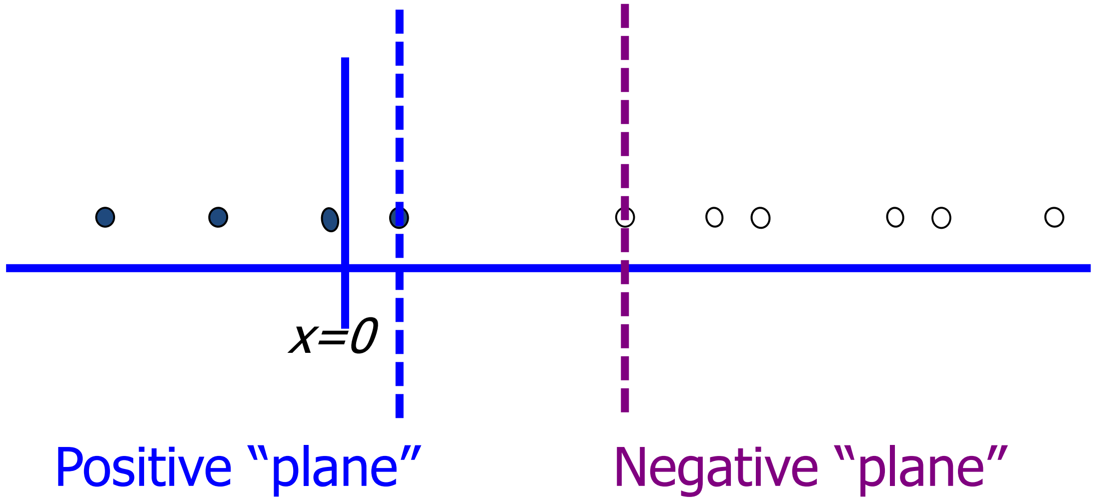
Kernel Trick & Kernel SVM

Roadmap

- Dual Support Vector Machine
  - another QP with valuable geometric messages
  - and almost no dependence on k
- Kernel Support Vector Machine
  - Kernel Trick
  - Polynomial Kernel
  - Gaussian Kernel
  - Comparison of Kernels

Suppose we’re in 1-dimension

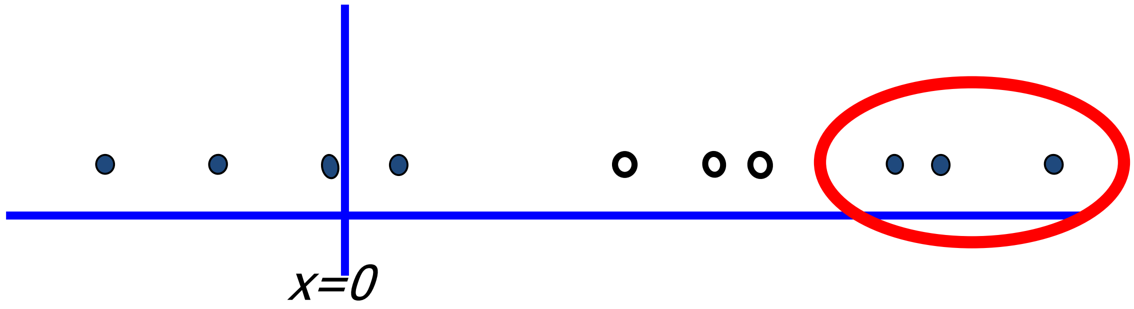
- What would SVMs do with this data?



- Not a big surprise

Harder 1-dimensional dataset

- What would SVMs do with this data?

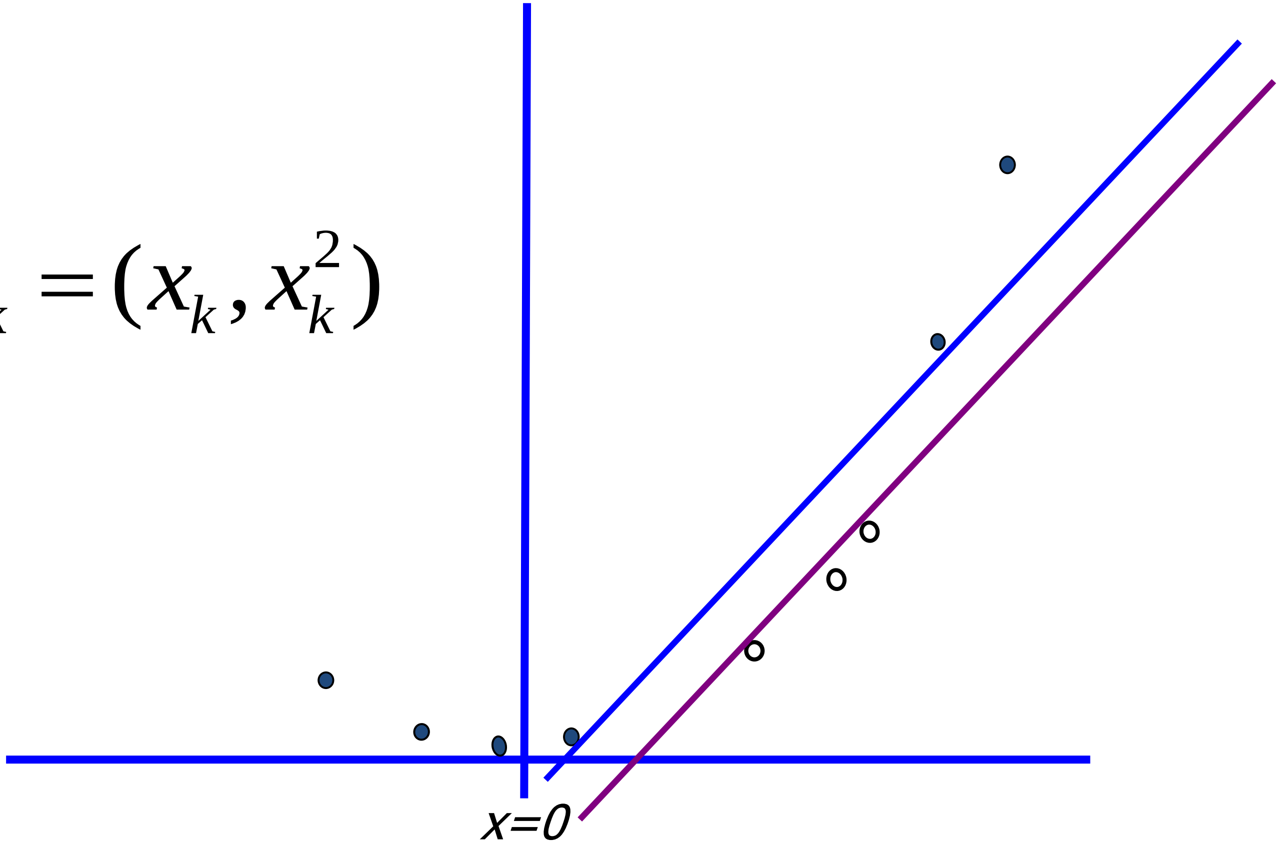


- That's wiped the smirk off SVM's face.
- What can be done about this?

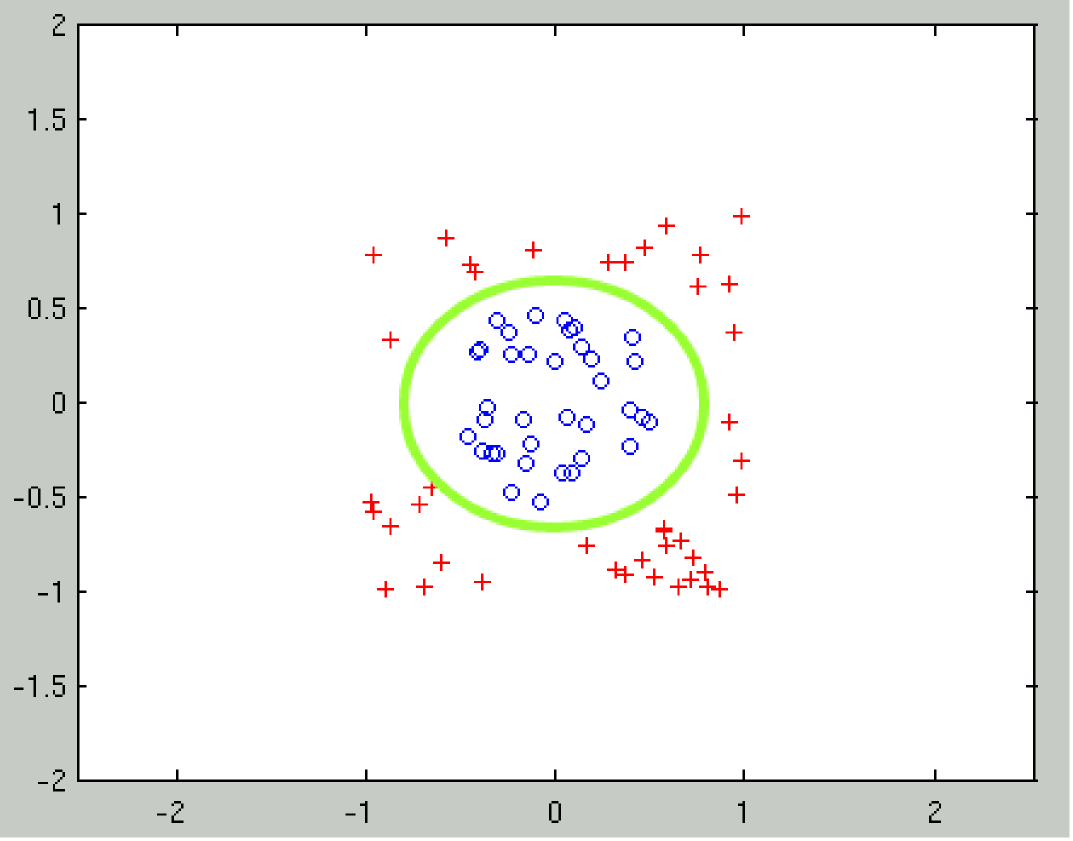
Harder 1-dimensional dataset

- Remember how permitting non-linear basis functions made linear regression so much nicer? Let's permit them here too

$$\mathbf{z}_k = (x_k, x_k^2)$$



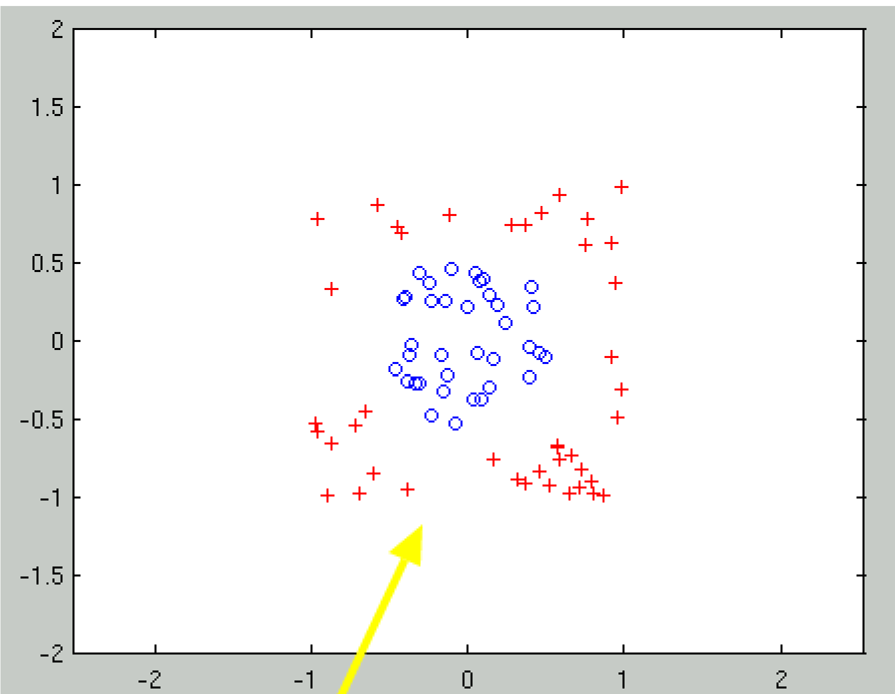
2-dimensional dataset

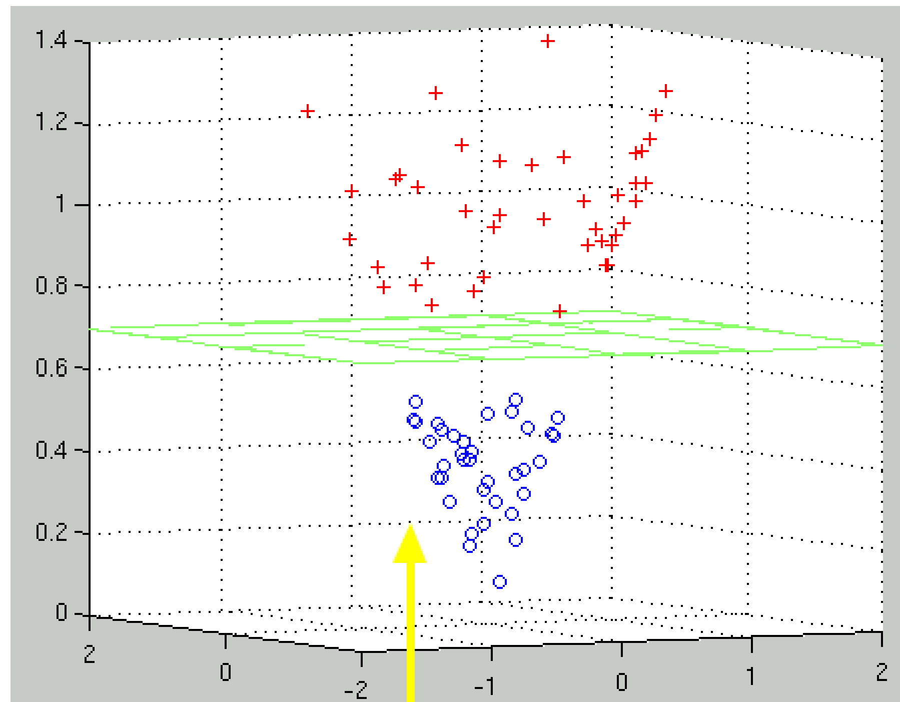


$$\mathbf{z} = \Phi(\mathbf{x}) = (x_1, x_2, \sqrt{x_1^2 + x_2^2})$$

2-dimensional dataset

$$(x_1, x_2) \Rightarrow (x_1, x_2, \sqrt{x_1^2 + x_2^2})$$

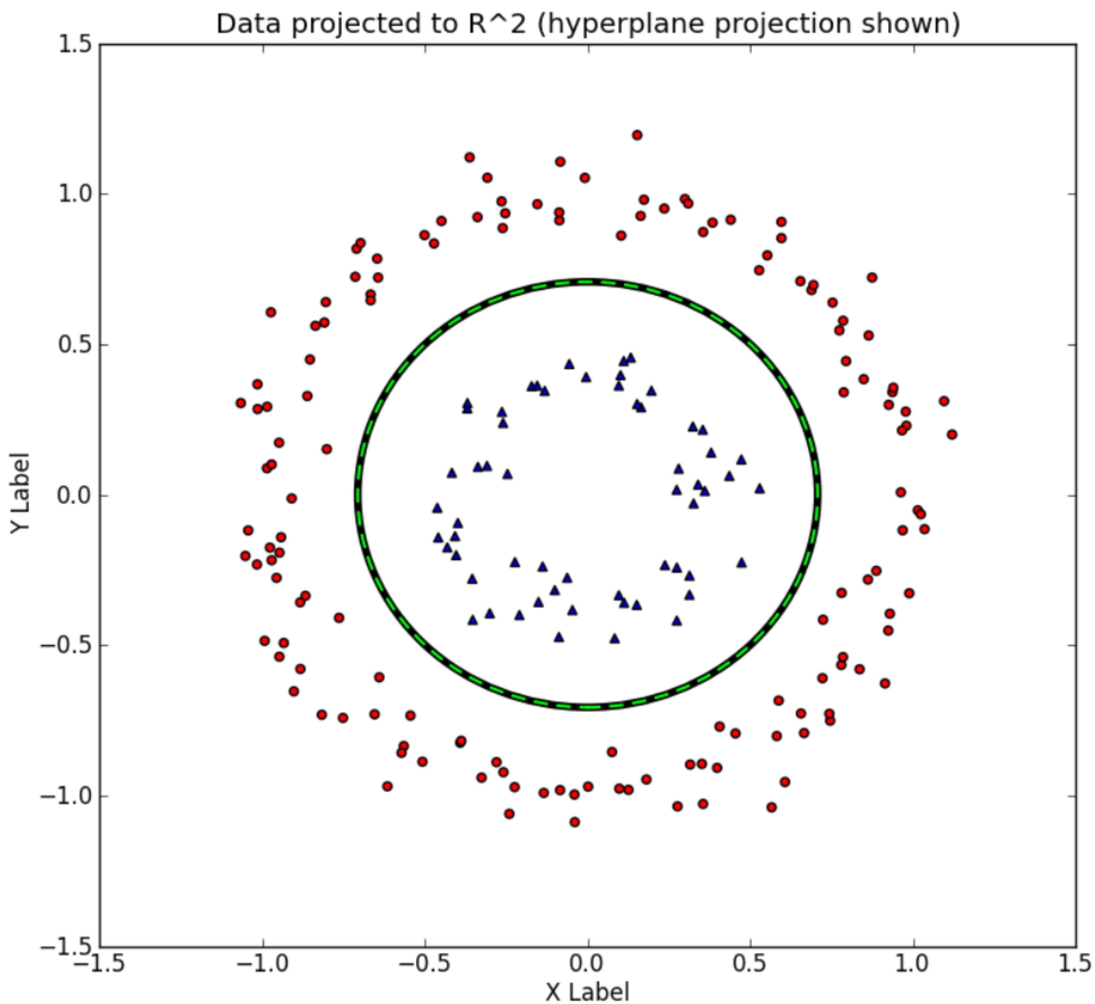




非线性可分的2维输入空间（Input space）

线性可分的3维特征空间（Feature space）

when transformed back to R^2, the decision boundary is nonlinear.



Dual SVM Revisited

- Goal: SVM without dependence on k

- optimal $\alpha = QP(\mathbf{Q}, \mathbf{p}, \mathbf{A}, \mathbf{c})$

$$\min_{\alpha} \quad \frac{1}{2}\alpha^T \mathbf{Q}\mathbf{u} + \mathbf{p}^T \alpha$$
$$\text{subject to} \quad \mathbf{a}_i^T \alpha \geq c_i, \quad \text{for } i = 1, 2, \ldots, N$$

- where : $\mathbf{Q}_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ → inner product in $\mathbb{R}^k$

- 如果需要对 **x** 进行升维? （例如出现线性不可分的情况）

$$\mathbf{z} = \Phi(\mathbf{x}) \Rightarrow \mathbf{z}_i^T \mathbf{z}_j = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$
$$\text{where } \mathbf{z} \in \mathbb{R}^d, d >> k$$

- 问题：是否仍然可以将内积计算的时间复杂度控制在 O(k)？

Fast Inner Product

- 2nd order polynomial transform

$$\Phi_2(\mathbf{x}) = (1, x_1, x_2, \cdots, x_k, x_1^2, x_1x_2, \cdots, x_1x_k, x_2x_1, x_2^2, \cdots, x_2x_k, \cdots, x_k^2)$$

$$\begin{aligned}\Phi_2(\mathbf{x})^T\Phi_2(\mathbf{x}') &= 1 + \sum_{i=1}^{k} x_i x_i' + \sum_{i=1}^{k}\sum_{j=1}^{k} x_i x_j x_i' x_j' \\ &= 1 + \sum_{i=1}^{k} x_i x_i' + \sum_{i=1}^{k} x_i x_i' \sum_{j=1}^{k} x_j x_j' \\ &= 1 + \mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')(\mathbf{x}^T\mathbf{x}')\end{aligned}$$

- 计算二阶多项式变换的内积的时间复杂度可以控制在 O(k)
- 由此可以定义kernel function：

$$K_\Phi(x, x') \equiv \Phi(\mathbf{x})^T\Phi(\mathbf{x}')$$

Kernel SVM with QP

- Kernel Hard-Margin SVM Algorithm

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{z}_i^T \mathbf{z}_j) - \sum_{i=1}^{N} \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i y_i = 0; \quad \alpha_i \geq 0,\ i = 1, 2, \cdots, N$$

$(1)\ \mathbf{Q}_{ij} = y_i y_j K(\mathbf{x}_i^T \mathbf{x}_j)$ **O(N²)** (kernel evaluation)

$(2)\ \alpha^* = QP(\mathbf{Q}, \mathbf{p}, \mathbf{A}, \mathbf{c})$ QP with **N** variables and **N + 1** constraints

$(3)\ b^* = y_j - \sum_{SV} \alpha_i y_i K(\mathbf{x}_i \cdot \mathbf{x}_j)$ **O(#SV)**

$(4)\ g_{svm}(x) = \text{sign}\left(\sum_{SV} \alpha_i^* y_i (\mathbf{x} \cdot \mathbf{x}_i) + b^*\right)$ **O(#SV)**

General Poly-2 Kernel

- Kernel Hard-Margin SVM Algorithm

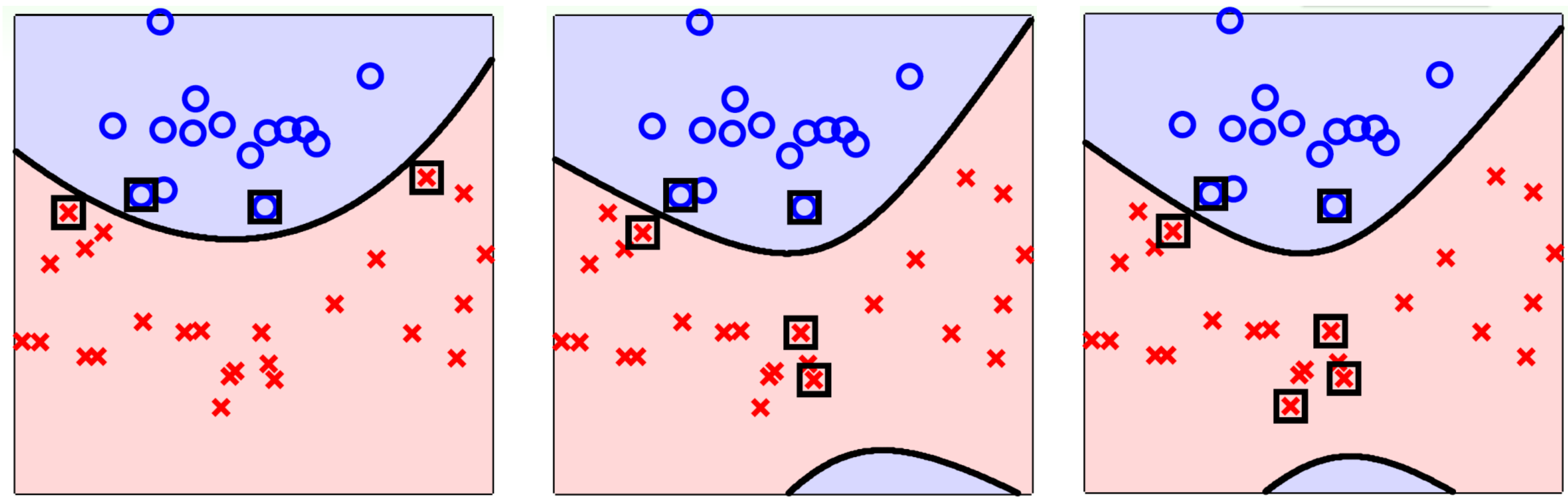
$$\Phi_2(\mathbf{x}) = (1, x_1, \cdots, x_k, x_1^2, \cdots, x_k^2) \quad \Rightarrow K_{\Phi_2}(\mathbf{x}, \mathbf{x}') = 1 + \mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')^2$$

$$\Phi_2'(\mathbf{x}) = (1, \sqrt{2}x_1, \cdots, \sqrt{2}x_k, x_1^2, \cdots, x_k^2) \Rightarrow K_{\Phi_2'}(\mathbf{x}, \mathbf{x}') = 1 + 2\mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')^2$$

$$\Phi_2''(\mathbf{x}) = (1, \sqrt{2\gamma}x_1, \cdots, \sqrt{2\gamma}x_k, \gamma x_1^2, \cdots, \gamma x_k^2) \quad \Rightarrow K_{\Phi_2''}(\mathbf{x}, \mathbf{x}') = (1 + \gamma\mathbf{x}^T\mathbf{x}')^2$$

- $K_{\Phi_2'}(\mathbf{x}, \mathbf{x}')$ somewhat easier to calculate than $K_{\Phi_2}(\mathbf{x}, \mathbf{x}')$
- $\Phi_2(\mathbf{x})$ and $\Phi_2''(\mathbf{x})$: equivalent power, different inner product
  - different inner product means different geometry
- commonly used : $K_{\Phi_2''}(\mathbf{x}, \mathbf{x}') = (1 + \gamma\mathbf{x}^T\mathbf{x}')^2$ with $\gamma > 0$

Poly-2 Kernels in Action



$(1 + 0.001\mathbf{x}^T\mathbf{x}')^2$ $\quad$ $1 + \mathbf{x}^T\mathbf{x}' + (\mathbf{x}^T\mathbf{x}')^2$ $\quad$ $(1 + 1000\mathbf{x}^T\mathbf{x}')^2$

- $\mathbf{g}_{\mathbf{SVM}}$ different, **SV**s different
  - hard to say which is better before learning
- change of kernel , change of margin definition
- need selecting kernel, just like selecting \phi

The RBF kernel

- Recall a kernel is any function of the form:

$$k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$$

where Φ is a function that projections vectors x into a new vector space.

- The RBF kernel is defined as

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$

Where γ is a parameter that sets the "spread" of the kernel.

The Φ function for an RBF kernel projects vectors into an **infinite dimensional** space. For Euclidean vectors, this space is an infinite dimensional Euclidean space. That is, we prove that

$$\Phi_{RBF} : \mathbb{R}^n \to \mathbb{R}^\infty$$

The RBF kernel

- Proof: Without loss of generality, let $\gamma = \frac{1}{2}$

$$
\begin{aligned}
k_{RBF}(\mathbf{x}, \mathbf{x}') &= \exp\left\{-\tfrac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right\} = \exp\left\{-\tfrac{1}{2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')\right\} \\
&= \exp\left\{-\tfrac{1}{2}\left(\mathbf{x}^T(\mathbf{x} - \mathbf{x}') - \mathbf{x}'^T(\mathbf{x} - \mathbf{x}')\right)\right\} \\
&= \exp\left\{-\tfrac{1}{2}\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{x}' - \mathbf{x}'^T\mathbf{x} + \mathbf{x}'^T\mathbf{x}'\right)\right\} \\
&= \exp\left\{-\tfrac{1}{2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathbf{x}^T\mathbf{x}'\right)\right\} \\
&= \exp\left\{-\tfrac{1}{2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2\right)\right\} \exp\left\{\mathbf{x}^T\mathbf{x}'\right\} \\
&= Ce^{\mathbf{x}^T\mathbf{x}'} \Rightarrow C := \exp\left\{-\tfrac{1}{2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2\right)\right\} \text{ is a constant} \\
&= C \sum_{n=0}^{\infty} \frac{(\mathbf{x}^T\mathbf{x}')^n}{n!} \Rightarrow \text{Taylor expansion of } e^x \\
&= C \sum_{n=0}^{\infty} \frac{K_{poly(n)}(\mathbf{x}^T\mathbf{x}')}{n!}
\end{aligned}
$$

Gaussian SVM

- The RBF kernel is defined as

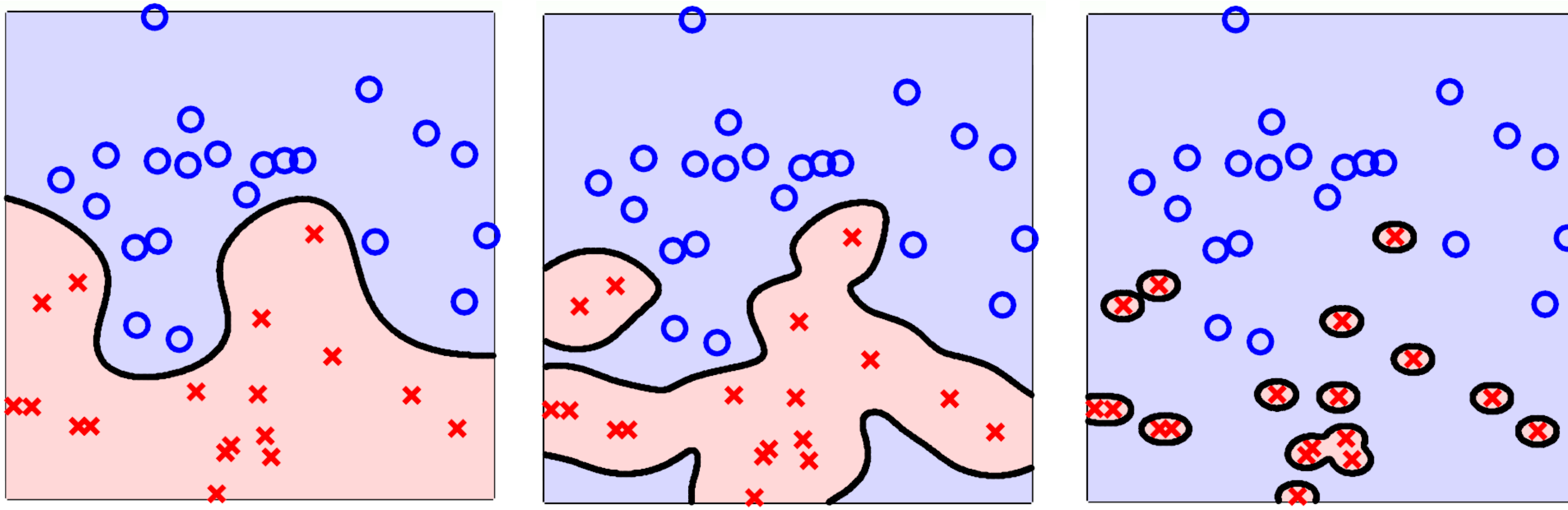
$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$

Where γ is a parameter that sets the "spread" of the kernel.

- Gaussian SVM:
  - find α_i to combine Gaussians centered at SVs $\mathbf{x}_i$
  - & achieve large margin in infinite-dim. space
- linear combination of Gaussians **centered at SVs**

$$g_{svm}(x) = \text{sign}\left(\sum_{SV_i} \alpha_i y_i \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2) + b\right)$$

Gaussian SVM in Action



$\exp(-1\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-10\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-100\|\mathbf{x}-\mathbf{x}'\|^2)$

- Large γ ? $\Rightarrow$ sharp Gaussians $\Rightarrow$ **Overfit ?**
- **warning: SVM can still overfit :-(**
- Gaussian SVM: need careful selection of γ

Quize

Consider the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$. What function does the kernel converge to if $\gamma \to \infty$?

1. $K_{\text{lim}}(\mathbf{x}, \mathbf{x}') = 0$
2. $K_{\text{lim}}(\mathbf{x}, \mathbf{x}') = [\![\mathbf{x} = \mathbf{x}']\!]$
3. $K_{\text{lim}}(\mathbf{x}, \mathbf{x}') = [\![\mathbf{x} \neq \mathbf{x}']\!]$
4. $K_{\text{lim}}(\mathbf{x}, \mathbf{x}') = 1$

Reference Answer: 2

If $\mathbf{x} = \mathbf{x}'$, $K(\mathbf{x}, \mathbf{x}') = 1$ regardless of γ. If $\mathbf{x} \neq \mathbf{x}'$, $K(\mathbf{x}, \mathbf{x}') = 0$ when $\gamma \to \infty$. Thus, K_{lim} is an impulse function, which is an extreme case of how the Gaussian gets sharper when $\gamma \to \infty$.

kernel trick

- **Kernel trick**
  - plug in efficient kernel function to avoid dependence on k
- **Overfitting ? … with this enormous number of terms**
  - The use of Maximum Margin magically makes this not a problem
- **The evaluation phase will be very expensive (why?)**
  - evaluation phase : doing a set of predictions on a test set
  - *What can be done?*
- **kernel SVM: predict with SV only**

Common SVM Kernel functions

- Linear:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$$

- Polynomial :

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \cdot \mathbf{x}_j + r)^d, \ \gamma > 0$$

- Radial Basis Function (RBF)

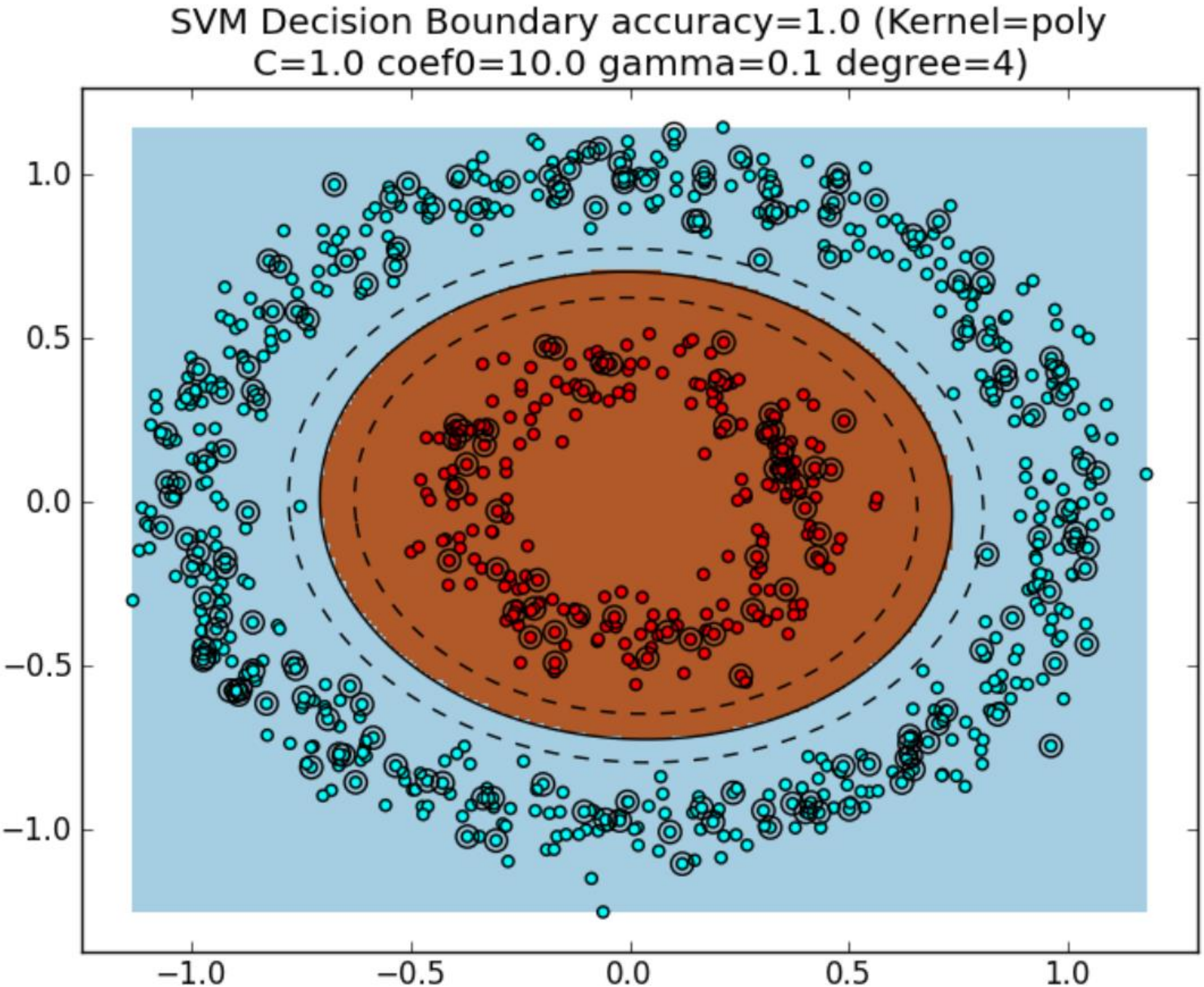
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \ \gamma > 0$$

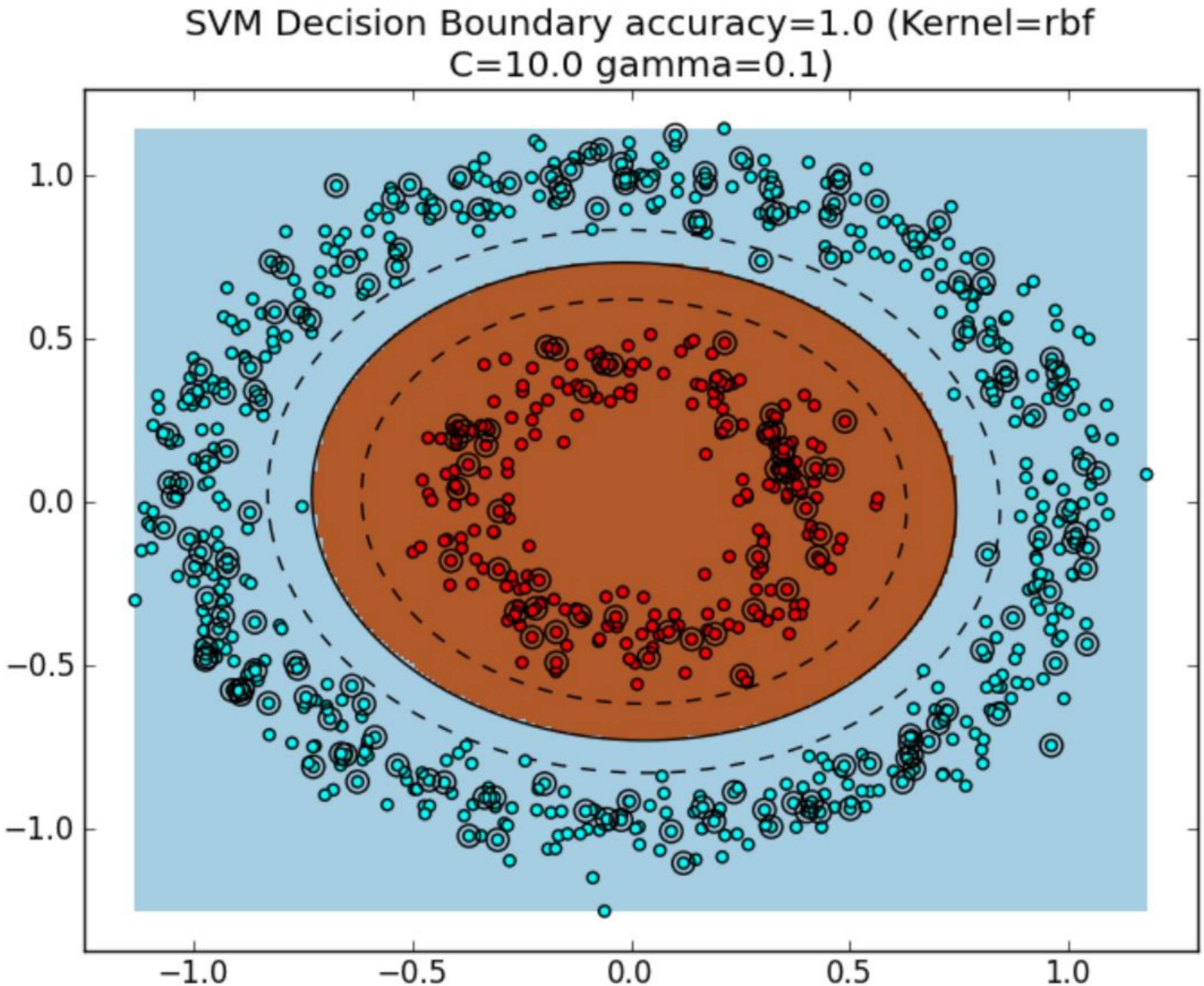
- Sigmoid:

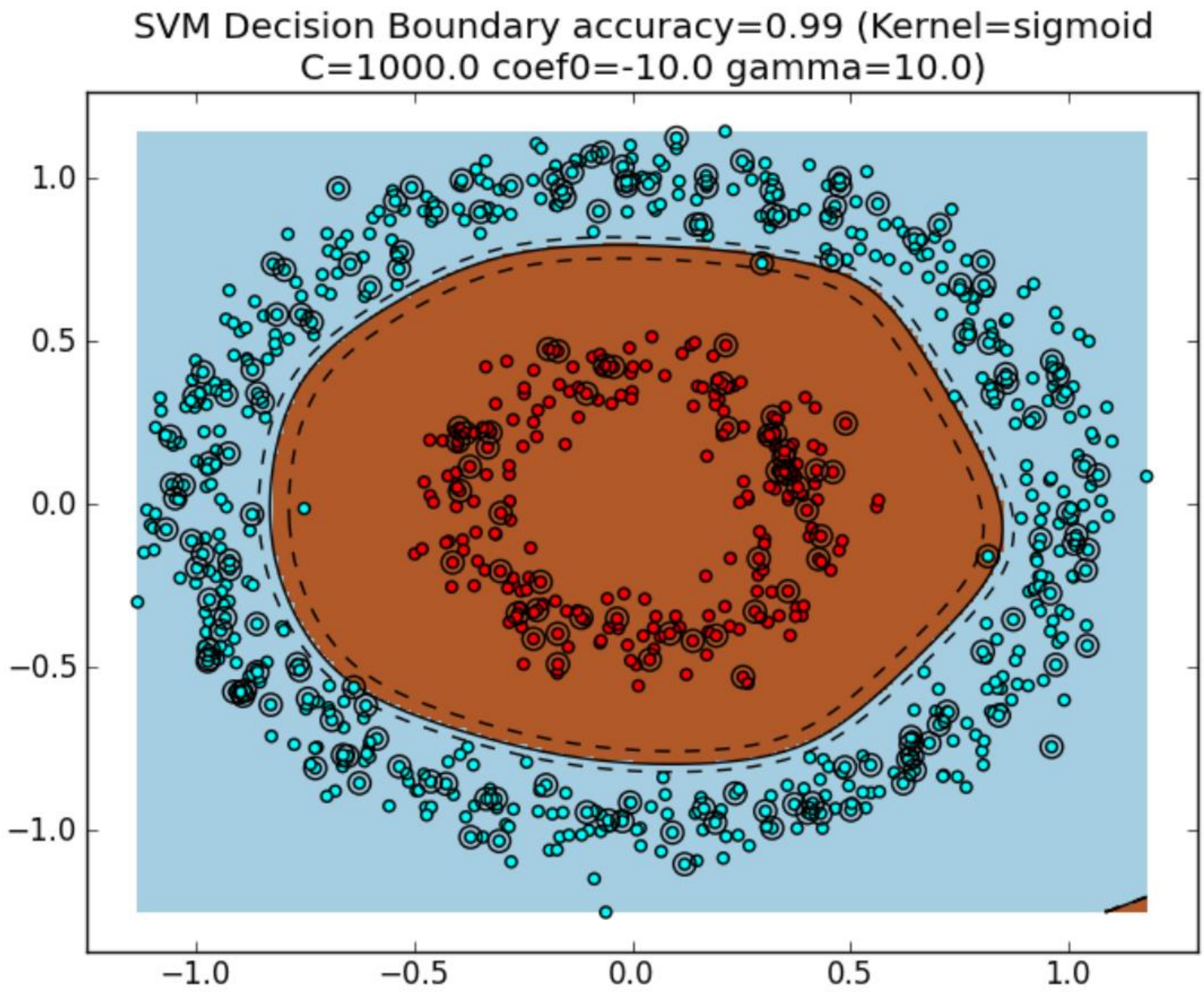
$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i \cdot \mathbf{x}_j + r)$$

$$\text{where}: \ \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

γ and *r* are magic parameters that must be chosen by a model selection method such as CV


SVM Decision Boundary accuracy=1.0 (Kernel=poly
C=1.0 coef0=10.0 gamma=0.1 degree=4)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0


SVM Decision Boundary accuracy=1.0 (Kernel=rbf
C=10.0 gamma=0.1)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0


SVM Decision Boundary accuracy=0.99 (Kernel=sigmoid
C=1000.0 coef0=-10.0 gamma=10.0)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

Other Valid Kernels

- kernel represents **special** similarity： $K(x,y)=\Phi(x)\cdot\Phi(y)$
  - any similarity → valid kernel? **not really**
- Mercer's condition
  - **necessary & sufficient** conditions for valid kernel
  - K(x, y) has to be positive semi-definite function
    - i.e., for any function f(x) whose $\int f^2(x)dx$ is finite
    - and the following inequality holds

$$\int dxdyf(x)K(x,y)f(y)\geq 0$$

Tips for evaluating whether a proposed kernel is valid

- define your own kernel: possible, **but hard**
- In a sense, a kernel is the opposite of a metric (distance).
  - A kernel function measures similarity: it is relatively large for similar inputs and relatively small for different inputs.
  - The opposite is true for a metric. A function which behaves like a metric is not a valid kernel!

Tips for evaluating whether a proposed kernel is valid

- It's not always easy to verify that the Gram matrix K will always be positive semi-definite.
  - But if you can find a small example data set (e.g. two or three points) for which K has negative determinant, it follows that K is not PSD.
  - This follows from the fact that the determinant of a matrix is the product of its eigenvalues, so negative determinant implies a negative eigenvalue.

Quize

Which of the following is not a valid kernel? (*Hint: Consider two 1-dimensional vectors* $\mathbf{x}_1 = (1)$ *and* $\mathbf{x}_2 = (-1)$ *and check Mercer's condition.*)

1. $K(\mathbf{x}, \mathbf{x}') = (-1 + \mathbf{x}^T\mathbf{x}')^2$
2. $K(\mathbf{x}, \mathbf{x}') = (0 + \mathbf{x}^T\mathbf{x}')^2$
3. $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T\mathbf{x}')^2$
4. $K(\mathbf{x}, \mathbf{x}') = (-1 - \mathbf{x}^T\mathbf{x}')^2$

Reference Answer: 1

The kernels in 2 and 3 are just polynomial kernels. The kernel in 4 is equivalent to the kernel in 3. For 1, the matrix K formed from the kernel and the two examples is not positive semi-definite. Thus, the underlying kernel is not a valid one.

Tips for evaluating whether a proposed kernel is valid

- If a function can be constructed as a **composition** of known valid kernels (like those listed above), it is a kernel.
- Any function φ(x) can be used to generate a kernel using

$$k(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})^T \Phi(\mathbf{y}).$$

- Here are some construction rules …

Kernel construction rules

(1) $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$

$c > 0$

(2) $K(\mathbf{x}, \mathbf{x}') = cK_1(\mathbf{x}, \mathbf{x}')$

$K_1()$ and $K_2()$ are valid kernels

(3) $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}')$

(4) $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}')K_2(\mathbf{x}, \mathbf{x}')$

(5) $K(\mathbf{x}, \mathbf{x}') = q(K_1(\mathbf{x}, \mathbf{x}'))$

$q()$ is a polynomial with positive coefficients

(6) $K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})K_1(\mathbf{x}, \mathbf{x}')f(\mathbf{x}')$

$f()$ is any function

(7) $K(\mathbf{x}, \mathbf{x}') = \exp(K_1(\mathbf{x}, \mathbf{x}'))$

(8) $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{A} \mathbf{x}'$

$\mathbf{A}$ is a PSD matrix

Validity of Gaussian kernel

- Gaussian kernel

$$k(\mathbf{x},\mathbf{x}') = \exp(-\|\mathbf{x}-\mathbf{x}'\|^2/2\sigma^2)$$

$$= \exp(-\mathbf{x}^\top\mathbf{x}/2\sigma^2)\exp(\mathbf{x}^\top\mathbf{x}'/\sigma^2)\exp(-\mathbf{x}'^\top\mathbf{x}'/2\sigma^2)$$

$$= f(\mathbf{x})\exp(\mathbf{x}^\top\mathbf{x}'/\sigma^2)f(\mathbf{x}')$$

- linear combination of Gaussians centered at SVs

$$g_{svm}(x) = \text{sign}\left(\sum_{SV_i}\alpha_i y_i \exp(-\gamma\|\mathbf{x}-\mathbf{x}_i\|^2)+b\right)$$

- Gaussian SVM:
  - find α_i to combine Gaussians centered at SVs $\mathbf{x}_i$
  - & achieve large margin in infinite-dim. space

Kernel Tricks

- Pro
  - Introducing nonlinearity into the model
  - Computational cheap
- Con
  - Still have potential overfitting problems

电子科技大学

UESTC

1956