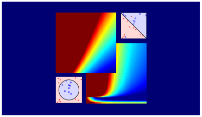
# Machine Learning Techniques
# (機器學習技巧)



## Lecture 13: RBF Networks

Hsuan-Tien Lin (林軒田)

`htlin@csie.ntu.edu.tw`

Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)

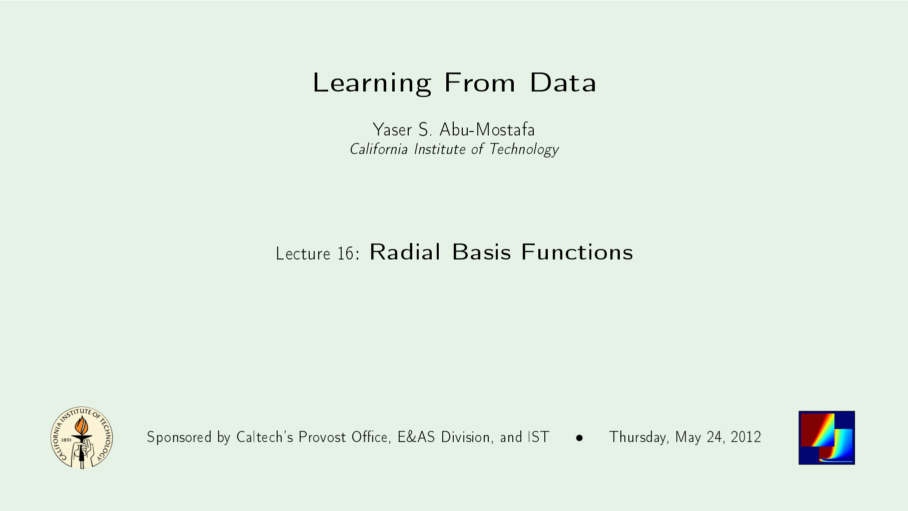
# Agenda

## Lecture 13: RBF Networks

- Full RBF Model
- Prototype Extraction
- RBF Network
- Connection to Other Views

# Disclaimer

Many parts of this lecture borrows
Prof. Yaser S. Abu-Mostafa's slides with permission.

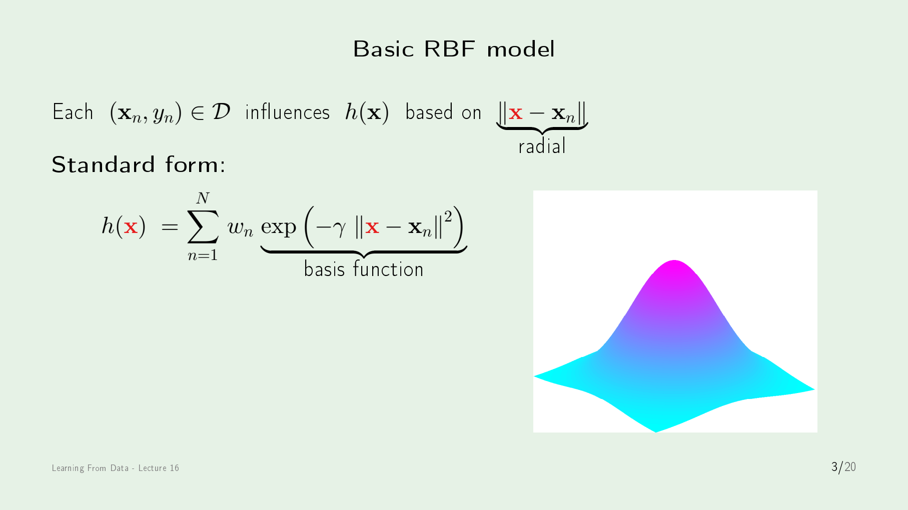


## Learning From Data

Yaser S. Abu-Mostafa
*California Institute of Technology*

Lecture 16: Radial Basis Functions

Sponsored by Caltech's Provost Office, E&AS Division, and IST • Thursday, May 24, 2012



## Basic RBF model

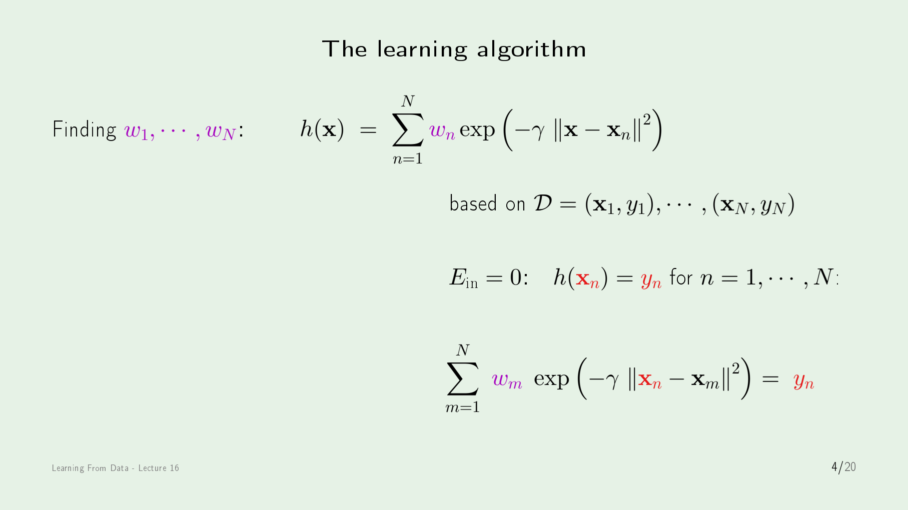
Each $(\mathbf{x}_n, y_n) \in \mathcal{D}$ influences $h(\mathbf{x})$ based on $\underbrace{\|\mathbf{x} - \mathbf{x}_n\|}_{\text{radial}}$

**Standard form:**

$$h(\mathbf{x}) = \sum_{n=1}^{N} w_n \underbrace{\exp\left(-\gamma \|\mathbf{x} - \mathbf{x}_n\|^2\right)}_{\text{basis function}}$$

Learning From Data - Lecture 16

3/20



## The learning algorithm

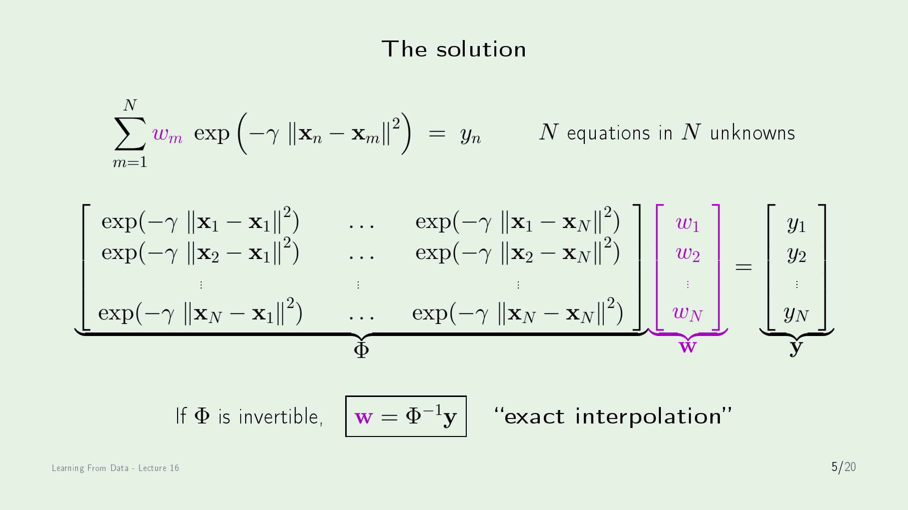
Finding $w_1, \cdots, w_N$:

$$h(\mathbf{x}) \;=\; \sum_{n=1}^{N} w_n \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{x}_n\right\|^2\right)$$

based on $\mathcal{D} = (\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$

$E_{\text{in}} = 0$: $\quad h(\mathbf{x}_n) = y_n$ for $n = 1, \cdots, N$:

$$\sum_{m=1}^{N} w_m \; \exp\left(-\gamma \left\|\mathbf{x}_n - \mathbf{x}_m\right\|^2\right) \;=\; y_n$$



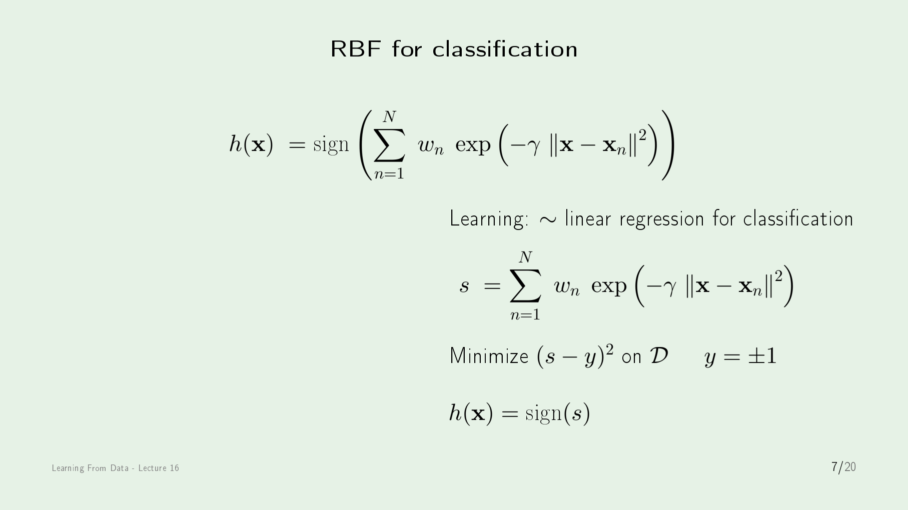
## The solution

$$\sum_{m=1}^{N} {\color{purple}w_m} \ \exp\left(-\gamma \ \|\mathbf{x}_n - \mathbf{x}_m\|^2\right) \ = \ y_n \qquad N \text{ equations in } N \text{ unknowns}$$

$$\underbrace{\begin{bmatrix} \exp(-\gamma \ \|\mathbf{x}_1 - \mathbf{x}_1\|^2) & \ldots & \exp(-\gamma \ \|\mathbf{x}_1 - \mathbf{x}_N\|^2) \\ \exp(-\gamma \ \|\mathbf{x}_2 - \mathbf{x}_1\|^2) & \ldots & \exp(-\gamma \ \|\mathbf{x}_2 - \mathbf{x}_N\|^2) \\ \vdots & \vdots & \vdots \\ \exp(-\gamma \ \|\mathbf{x}_N - \mathbf{x}_1\|^2) & \ldots & \exp(-\gamma \ \|\mathbf{x}_N - \mathbf{x}_N\|^2) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{y}}$$

If $\Phi$ is invertible, $\boxed{\mathbf{w} = \Phi^{-1}\mathbf{y}}$ "exact interpolation"

Learning From Data - Lecture 16 5/20



## RBF for classification

$$h(\mathbf{x}) = \operatorname{sign}\left(\sum_{n=1}^{N} w_n \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{x}_n\right\|^2\right)\right)$$

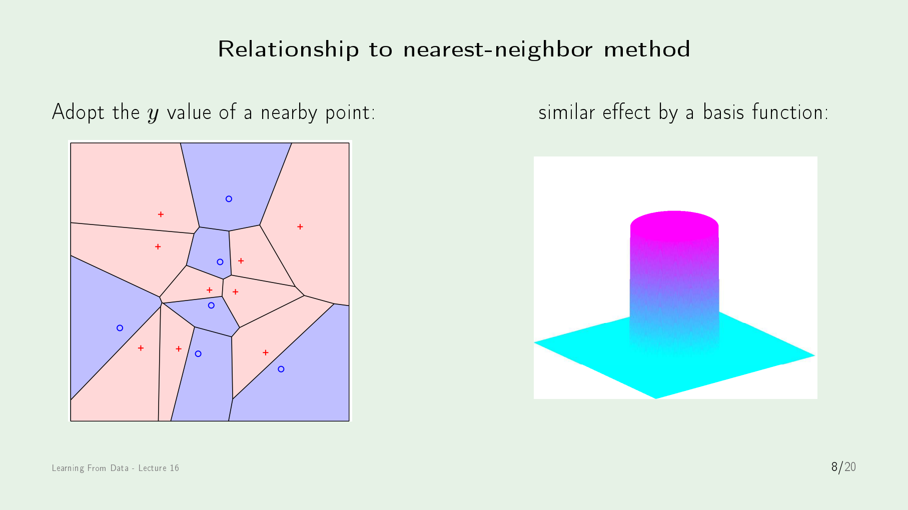
Learning: $\sim$ linear regression for classification

$$s = \sum_{n=1}^{N} w_n \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{x}_n\right\|^2\right)$$

Minimize $(s - y)^2$ on $\mathcal{D}$ $\quad y = \pm 1$

$$h(\mathbf{x}) = \operatorname{sign}(s)$$

Learning From Data - Lecture 16 7/20

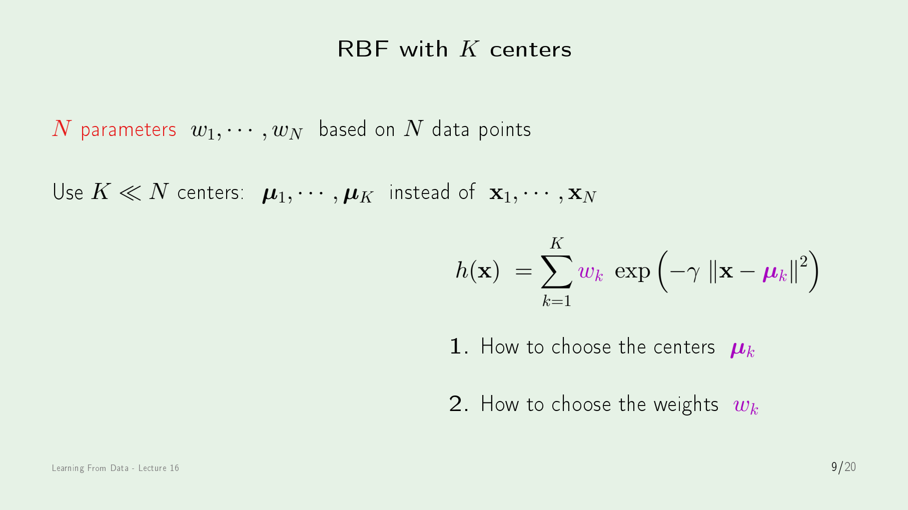


## Relationship to nearest-neighbor method

Adopt the $y$ value of a nearby point:

similar effect by a basis function:

# Fun Time



## RBF with $K$ centers

$N$ parameters $w_1, \cdots, w_N$ based on $N$ data points

Use $K \ll N$ centers: $\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K$ instead of $\mathbf{x}_1, \cdots, \mathbf{x}_N$

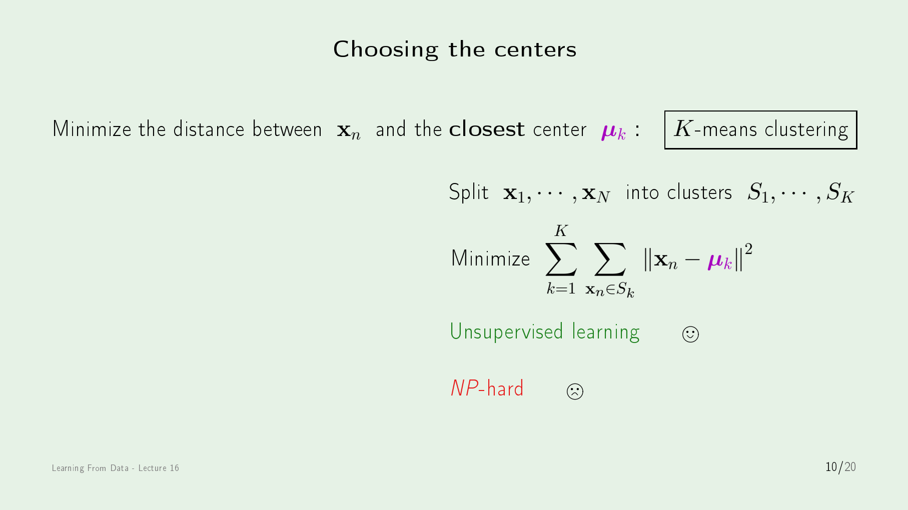
$$h(\mathbf{x}) = \sum_{k=1}^{K} w_k \exp\left(-\gamma \left\|\mathbf{x} - \boldsymbol{\mu}_k\right\|^2\right)$$

1. How to choose the centers $\boldsymbol{\mu}_k$

2. How to choose the weights $w_k$

Learning From Data - Lecture 16

9/20



## Choosing the centers

Minimize the distance between $\mathbf{x}_n$ and the **closest** center $\boldsymbol{\mu}_k$ : $\boxed{K\text{-means clustering}}$

Split $\mathbf{x}_1, \cdots, \mathbf{x}_N$ into clusters $S_1, \cdots, S_K$

$$\text{Minimize } \sum_{k=1}^{K} \sum_{\mathbf{x}_n \in S_k} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

Unsupervised learning ☺

*NP*-hard ☹

Learning From Data - Lecture 16

10/20

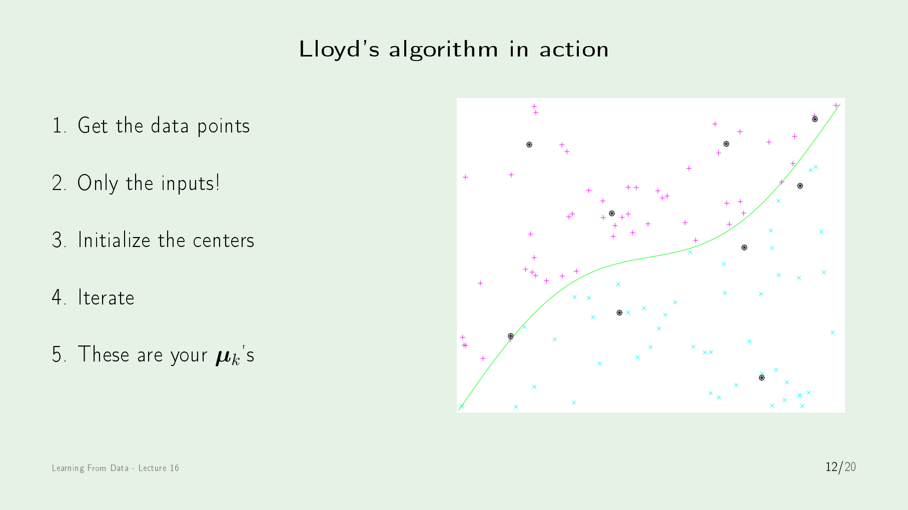
## An iterative algorithm

**Lloyd's algorithm:** Iteratively minimize $\displaystyle\sum_{k=1}^{K}\sum_{\mathbf{x}_n\in S_k}\|\mathbf{x}_n-\boldsymbol{\mu}_k\|^2$ w.r.t. $\boldsymbol{\mu}_k, S_k$

$$\boldsymbol{\mu}_k \leftarrow \frac{1}{|S_k|}\sum_{\mathbf{x}_n\in S_k}\mathbf{x}_n$$

$$S_k \leftarrow \{\mathbf{x}_n : \|\mathbf{x}_n-\boldsymbol{\mu}_k\| \le \text{all } \|\mathbf{x}_n-\boldsymbol{\mu}_\ell\|\}$$

Convergence ⟶ **local minimum**



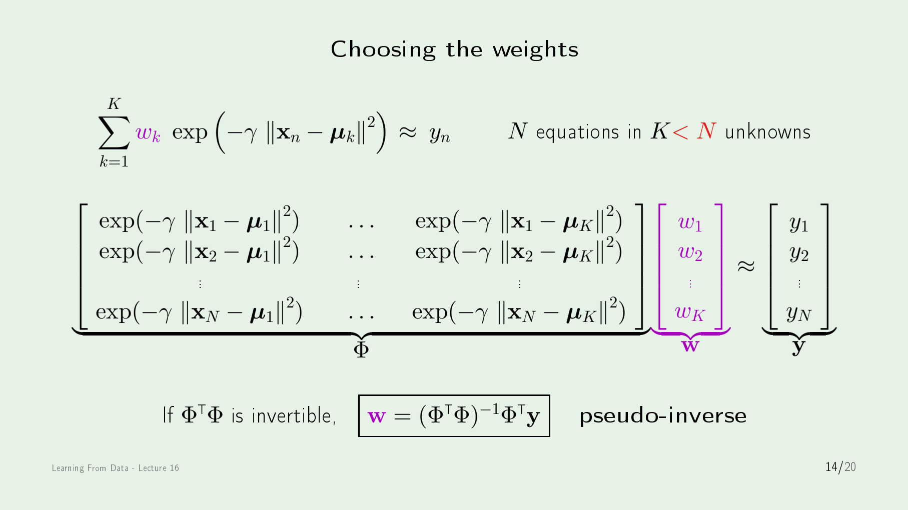
## Lloyd's algorithm in action

1. Get the data points
2. Only the inputs!
3. Initialize the centers
4. Iterate
5. These are your $\boldsymbol{\mu}_k$'s

Learning From Data - Lecture 16

12/20

# Fun Time



## Choosing the weights

$$\sum_{k=1}^{K} w_k \exp\left(-\gamma \left\|\mathbf{x}_n - \boldsymbol{\mu}_k\right\|^2\right) \approx y_n$$

$N$ equations in $K < N$ unknowns

$$\underbrace{\begin{bmatrix} \exp(-\gamma \|\mathbf{x}_1 - \boldsymbol{\mu}_1\|^2) & \dots & \exp(-\gamma \|\mathbf{x}_1 - \boldsymbol{\mu}_K\|^2) \\ \exp(-\gamma \|\mathbf{x}_2 - \boldsymbol{\mu}_1\|^2) & \dots & \exp(-\gamma \|\mathbf{x}_2 - \boldsymbol{\mu}_K\|^2) \\ \vdots & \vdots & \vdots \\ \exp(-\gamma \|\mathbf{x}_N - \boldsymbol{\mu}_1\|^2) & \dots & \exp(-\gamma \|\mathbf{x}_N - \boldsymbol{\mu}_K\|^2) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}}_{\mathbf{w}} \approx \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}}_{\mathbf{y}}$$

If $\Phi^{\mathsf{T}}\Phi$ is invertible, $\boxed{\mathbf{w} = (\Phi^{\mathsf{T}}\Phi)^{-1}\Phi^{\mathsf{T}}\mathbf{y}}$ pseudo-inverse

Learning From Data - Lecture 16

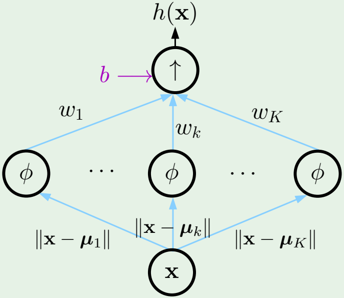
14/20

## RBF network

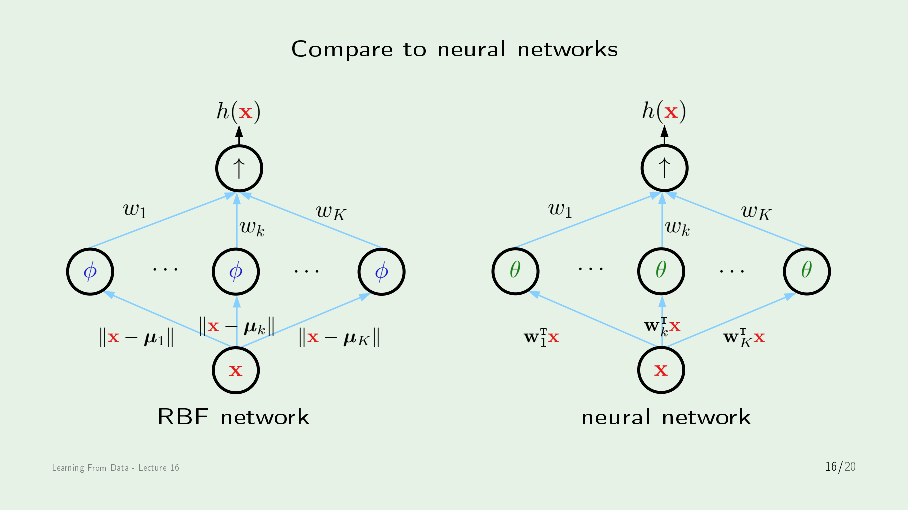
The "features" are $\exp\left(-\gamma \left\|\mathbf{x}-\boldsymbol{\mu}_k\right\|^2\right)$

Nonlinear transform depends on $\mathcal{D}$

$\Longrightarrow$ No longer a linear model


$h(\mathbf{x})$
$\uparrow$
$b$
$w_1$
$w_k$
$w_K$
$\phi$
$\cdots$
$\phi$
$\cdots$
$\phi$
$\|\mathbf{x}-\boldsymbol{\mu}_1\|$
$\|\mathbf{x}-\boldsymbol{\mu}_k\|$
$\|\mathbf{x}-\boldsymbol{\mu}_K\|$
$\mathbf{x}$

A bias term ($b$ or $w_0$) is often added



Compare to neural networks

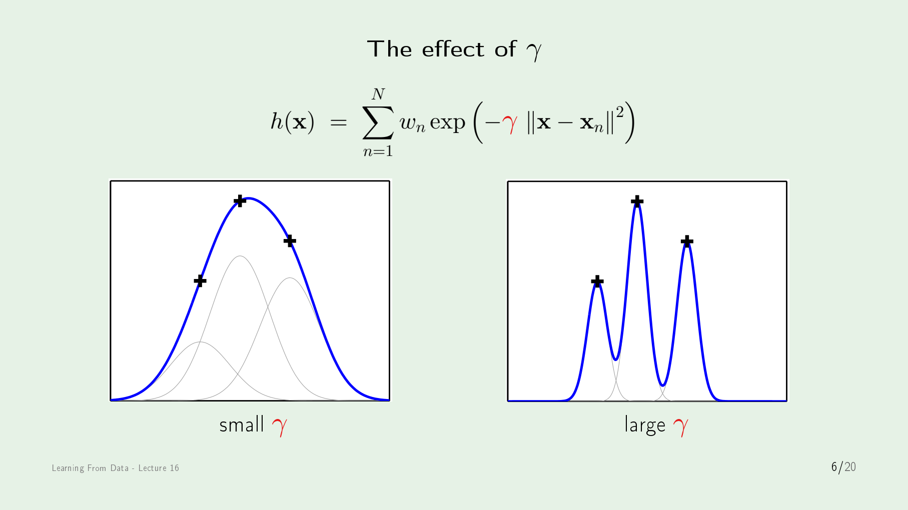
$h(\mathbf{x})$
$\uparrow$
$w_1$
$w_k$
$w_K$
$\phi$
$\cdots$
$\phi$
$\cdots$
$\phi$
$\|\mathbf{x}-\boldsymbol{\mu}_1\|$
$\|\mathbf{x}-\boldsymbol{\mu}_k\|$
$\|\mathbf{x}-\boldsymbol{\mu}_K\|$
$\mathbf{x}$
$h(\mathbf{x})$
$\uparrow$
$w_1$
$w_k$
$w_K$
$\theta$
$\cdots$
$\theta$
$\cdots$
$\theta$
$\mathbf{w}_1^{\mathrm{T}}\mathbf{x}$
$\mathbf{w}_k^{\mathrm{T}}\mathbf{x}$
$\mathbf{w}_K^{\mathrm{T}}\mathbf{x}$
$\mathbf{x}$

RBF network

neural network

Learning From Data - Lecture 16

16/20



## The effect of $\gamma$

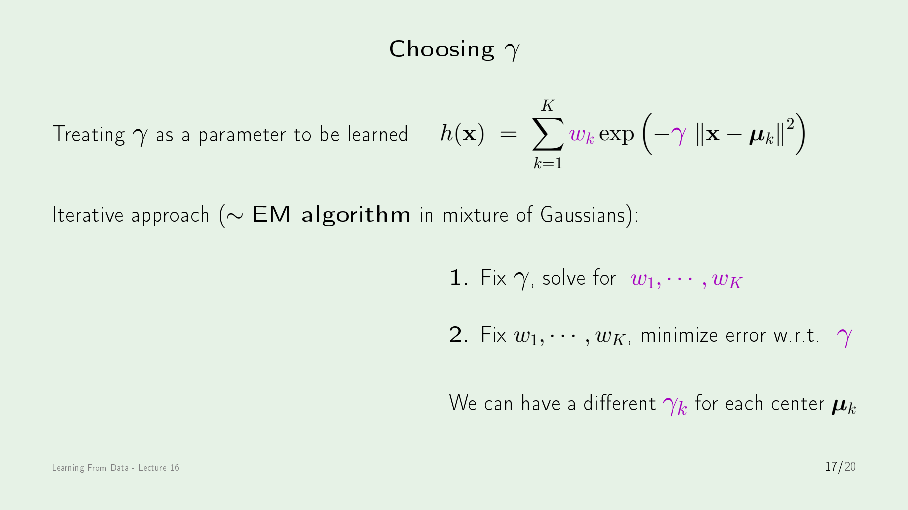
$$h(\mathbf{x}) = \sum_{n=1}^{N} w_n \exp\left(-\gamma \left\|\mathbf{x} - \mathbf{x}_n\right\|^2\right)$$

small $\gamma$

large $\gamma$

Learning From Data - Lecture 16

6/20



## Choosing $\gamma$

Treating $\gamma$ as a parameter to be learned $$h(\mathbf{x}) = \sum_{k=1}^{K} w_k \exp\left(-\gamma \left\|\mathbf{x} - \boldsymbol{\mu}_k\right\|^2\right)$$

Iterative approach ($\sim$ **EM algorithm** in mixture of Gaussians):

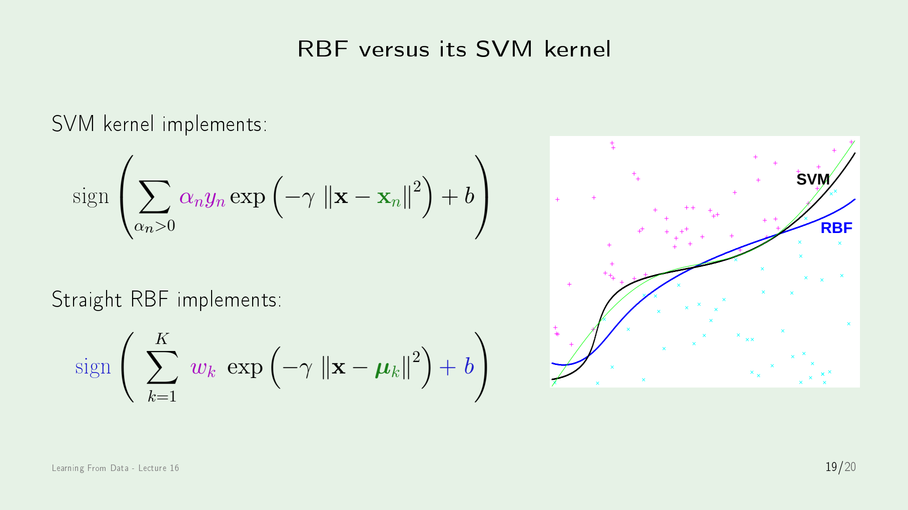
1. Fix $\gamma$, solve for $w_1, \cdots, w_K$
2. Fix $w_1, \cdots, w_K$, minimize error w.r.t. $\gamma$

We can have a different $\gamma_k$ for each center $\boldsymbol{\mu}_k$

Learning From Data - Lecture 16

17/20

# Fun Time



## RBF versus its SVM kernel

SVM kernel implements:

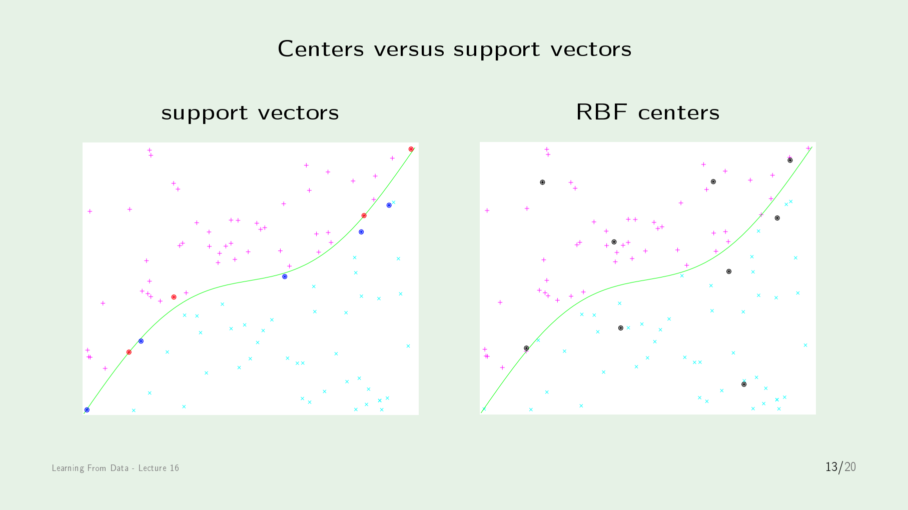
$$\text{sign}\left(\sum_{\alpha_n>0} \alpha_n y_n \exp\left(-\gamma \left\|\mathbf{x}-\mathbf{x}_n\right\|^2\right) + b\right)$$

Straight RBF implements:

$$\text{sign}\left(\sum_{k=1}^{K} w_k \, \exp\left(-\gamma \left\|\mathbf{x}-\boldsymbol{\mu}_k\right\|^2\right) + b\right)$$

Learning From Data - Lecture 16

19/20



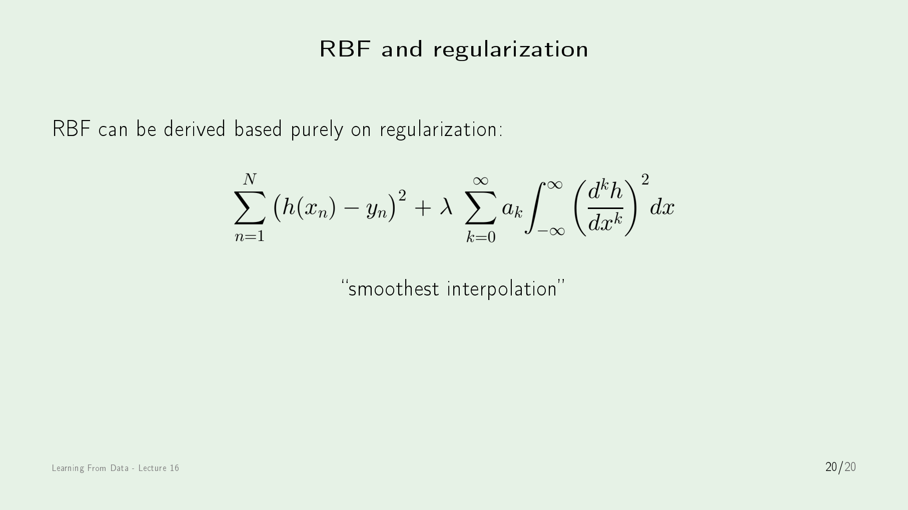
## Centers versus support vectors

support vectors

RBF centers

Learning From Data - Lecture 16

13/20



## RBF and regularization

RBF can be derived based purely on regularization:

$$\sum_{n=1}^{N}\left(h(x_n)-y_n\right)^2+\lambda\sum_{k=0}^{\infty}a_k\int_{-\infty}^{\infty}\left(\frac{d^kh}{dx^k}\right)^2dx$$

"smoothest interpolation"

Learning From Data - Lecture 16

20/20

# Fun Time

# Summary

## Lecture 13: RBF Networks

- Full RBF Model
- Prototype Extraction
- RBF Network
- Connection to Other Views