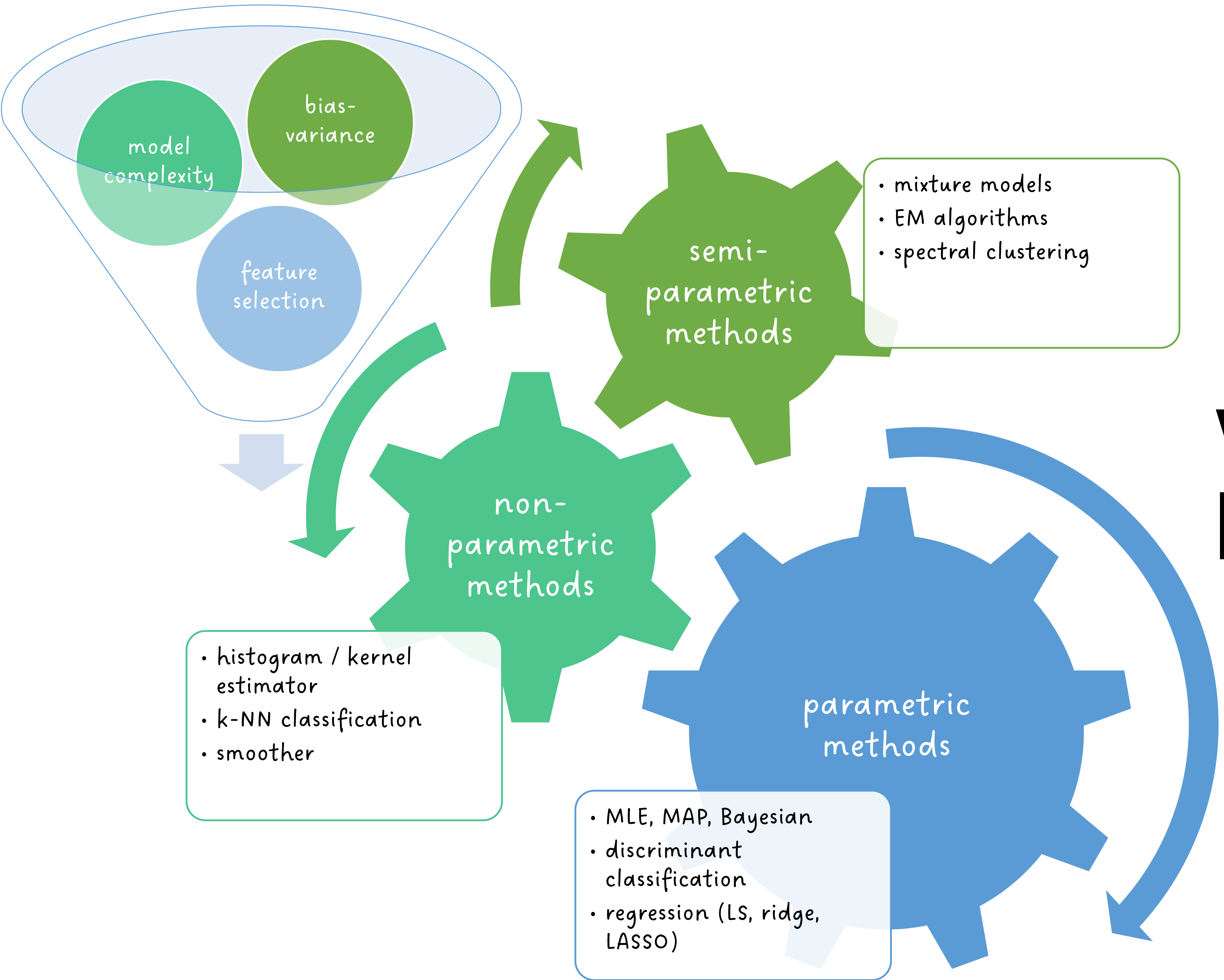
Kernel

STATS 303 Statistical Machine Learning

Spring 2022

Lecture 11

what we have so far …



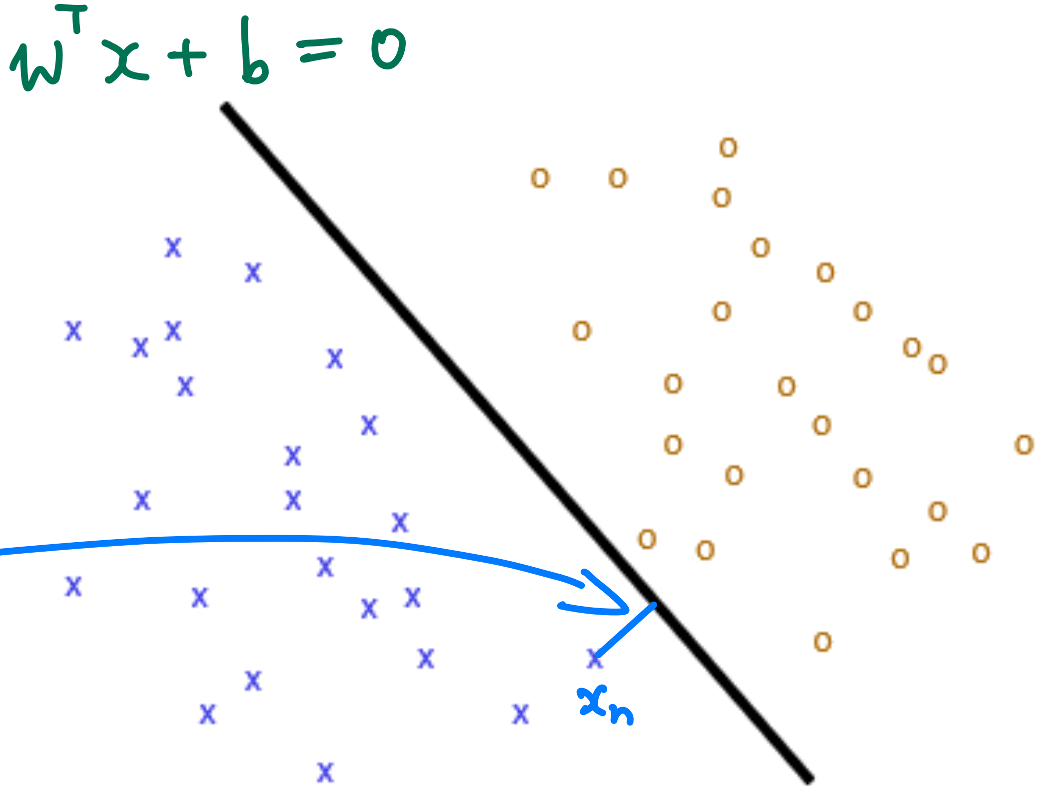
density-based
(generative approach)

Vapnik's Principle

- If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step.

linear SVM

- The best example that follows Vapnik's Principle is the support vector machine (SVM) that we learned in STATS302.



$$\frac{|w^T x_n + b|}{\|w\|}$$

linear SVM

- Let $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$ be the sample of training data for the classification problem, where $t_n \in \{+1, -1\}$ is the label.
- The linear SVM solves

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 \\ \text{s.t.} \quad & t_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b\right) \geq 1 \quad \text{for any } n \end{aligned}$$

- The dual problem is

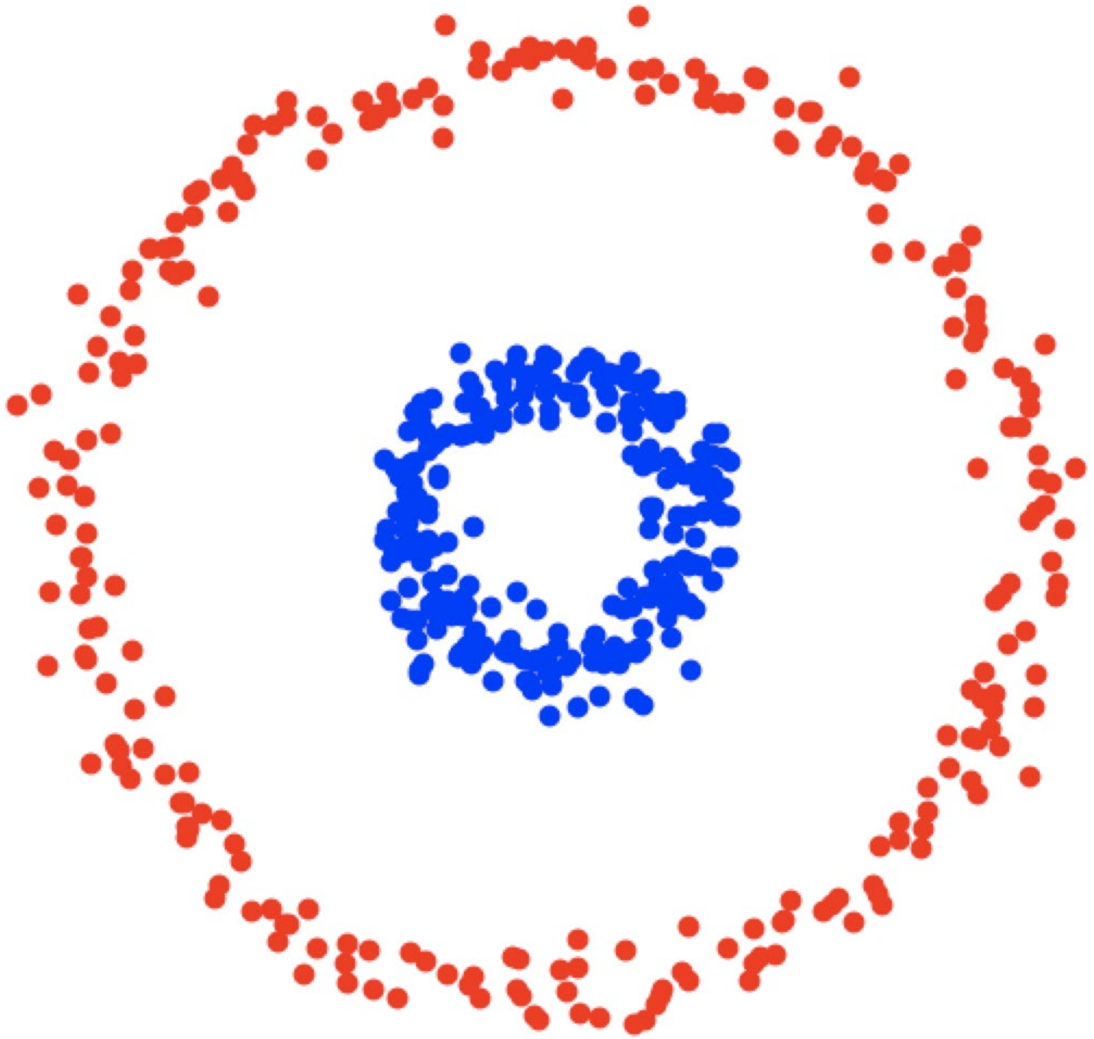
$$w = \sum_{n=1}^{N} a_n t_n x_n$$

$$\begin{aligned} \max_{\mathbf{a}} \quad & \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m \mathbf{x}_n{}^{\mathrm{T}} \mathbf{x}_m \\ \text{s.t.} \quad & a_n \geq 0 \quad \text{for all } n; \quad \text{and } \sum_{n=1}^{N} a_n t_n = 0 \end{aligned}$$

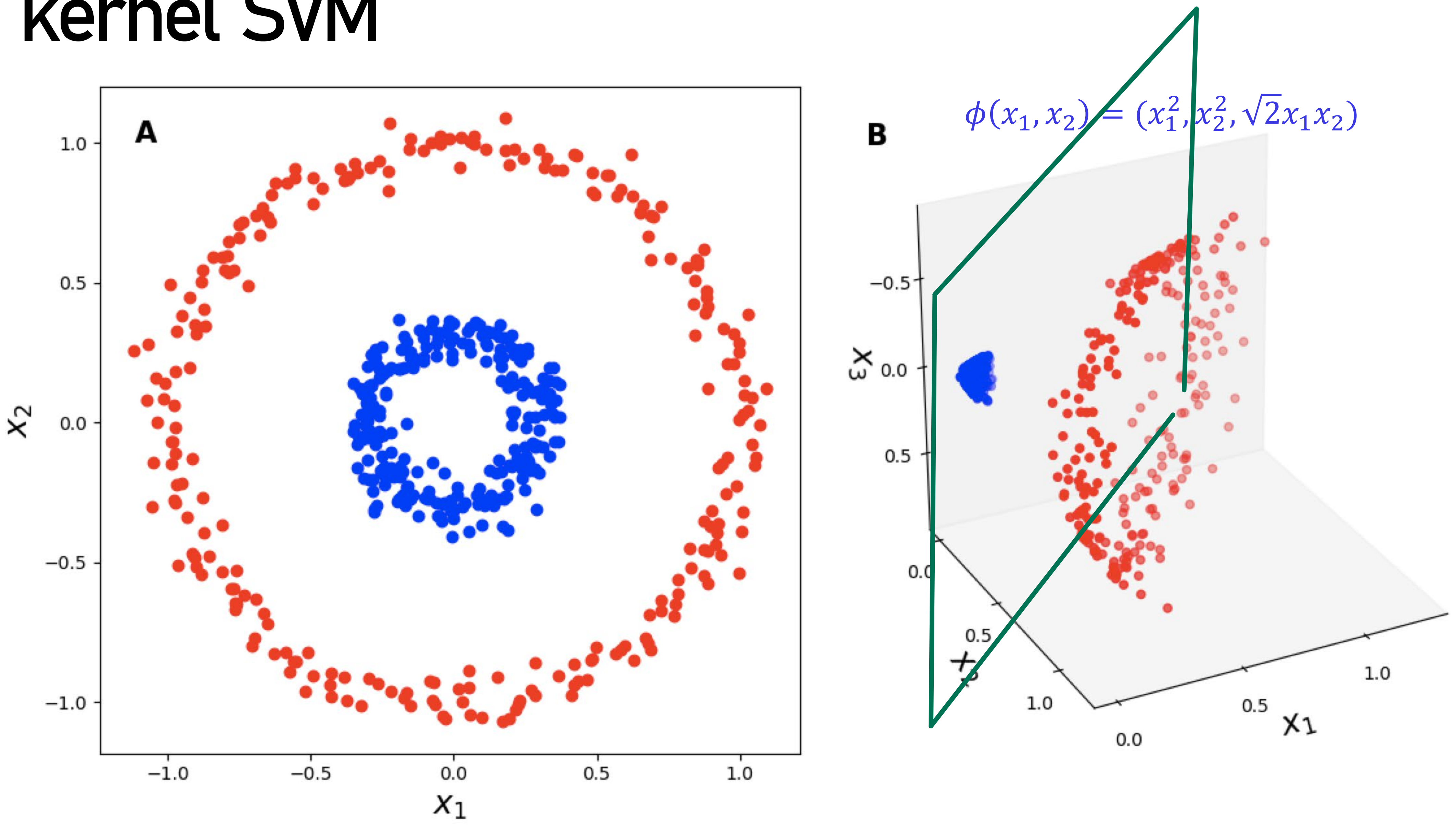
Lagrange multipliers

kernel SVM

- If data are not linearly separable:



kernel SVM


A
1.0
0.5
0.0
−0.5
−1.0
x_2
−1.0
−0.5
0.0
0.5
1.0
x_1
B
$\phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$
−0.5
0.0
0.5
x_3
0.0
0.5
1.0
x_2
0.0
0.5
1.0
x_1

kernel SVM

- Replacing each data point $\mathbf{x}_n$ by $\phi(\mathbf{x}_n)$ in the formulation of SVM:

$$\min_{\mathbf{w},b} \quad \frac{1}{2} \| \mathbf{w} \|^2$$

$$\text{s.t.} \quad t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b\right) \geq 1 \quad \text{for any } n$$

- Also, the dual problem becomes:

$$\max_{\mathbf{a}} \quad \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m \phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m)$$

$$\text{s.t.} \quad a_n \geq 0 \quad \text{for all } n; \quad \text{and } \sum_{n=1}^{N} a_n t_n = 0$$

kernel SVM

- Oftentimes, instead of finding a good mapping $\phi(\cdot)$, it is easier to define a "kernel" K , such that

$$K(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^{\mathrm{T}} \phi(\mathbf{x}_m)$$

- Given that, the dual form of the kernel SVM problem becomes

$$\max_{\mathbf{a}} \quad \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m K(\mathbf{x}_n, \mathbf{x}_m)$$

$$\text{s.t.} \quad a_n \geq 0 \quad \text{for all } n; \qquad \text{and } \sum_{n=1}^{N} a_n t_n = 0$$

$$w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$$

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}) + b = \sum_{n=1}^{N} a_n t_n K(\mathbf{x}, \mathbf{x}_n) + b$$

example of nonlinear map ϕ

- Let $\mathbf{x} = [x_1, x_2]^{\mathrm{T}}$,$\mathbf{y} = [y_1, y_2]^{\mathrm{T}}$

- Let $\phi(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2]^{\mathrm{T}}$

$\phi(x)^T\phi(y) = 1 + 2x_1y_1 + 2x_2y_2 + 2x_1x_2y_1y_2 + x_1^2y_1^2 + x_2^2y_2^2$

- $\phi(\mathbf{x})^{\mathrm{T}}\phi(\mathbf{y}) = \left(\mathbf{x}^{\mathrm{T}}\mathbf{y} + 1\right)^2 =: K(\mathbf{x}, \mathbf{y})$ $= (x_1y_1 + x_2y_2 + 1)^2$

use K instead of ϕ

- In practice, instead of starting with ϕ, we can directly start with K.
- Popular choices of K:
    - polynomial: $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^{\mathrm{T}}\mathbf{y} + 1)^q$
    - radial basis function (RBF): $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$
    - sigmoidal: $K(\mathbf{x}, \mathbf{y}) = \tanh(2\mathbf{x}^T\mathbf{y} + 1)$

food for thought

1. In general, what K can be chosen?
2. If e.g., $K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$, what ϕ is behind it?

kernel PCA

kernel PCA

- Kernel methods are not just for SVMs.
- Suppose we want to do dimension reduction using principal component analysis (PCA) for some data that do not show a clear linear structure.
- Nevertheless, after applying a nonlinear function ϕ, the transformed data show a clear linear low-dimensional structure.
- We can apply PCA to $\{\phi(\mathbf{x}_n)\}_{n=1}^{N}$
- The question is how to avoid defining ϕ and use K instead.

$$\{x_n\}_{n=1}^N \xrightarrow{\text{nonlinear}} \{\phi(x_n)\}_{n=1}^N \xrightarrow{\text{apply PCA}} ?$$

Centralization: $\mu = \frac{1}{N}\sum_{n=1}^N \phi(x_n)$

Data matrix

$$\Phi = \begin{bmatrix} (\phi(x_1) - \mu)^T \\ (\phi(x_2) - \mu)^T \\ \vdots \\ (\phi(x_N) - \mu)^T \end{bmatrix}$$

PCA on $\{\phi(x_n)\}_{n=1}^N$

$\Leftrightarrow$ Singular Value Decomposition (SVD) of Φ

$\Leftrightarrow$ Eigenproblem of $\Phi\Phi^T$.

The (n,m)-th entry of $\Phi\Phi^T$ is given by

$$(\phi(x_n) - \mu)^T(\phi(x_m) - \mu)$$

$$= \phi(x_n)^T\phi(x_m) - \mu^T\phi(x_m) - \mu^T\phi(x_n) + \mu^T\mu$$

$$= \phi(x_n)^T\phi(x_m) - \frac{1}{N}\sum_{l=1}^N \phi(x_l)^T\phi(x_m) - \frac{1}{N}\sum_{l=1}^N \phi(x_l)^T\phi(x_n) + \frac{1}{N^2}\sum_{l=1}^N\sum_{k=1}^N \phi(x_l)^T\phi(x_k)$$

Let K be the Gram matrix whose (n,m)-th entry is given by $K(n,m) = \phi(x_n)^T \phi(x_m)$

Let 😺 $:= \Phi \Phi^T$. Then

😺 (n,m)

$$= \phi(x_n)^T \phi(x_m) - \frac{1}{N} \sum_{l=1}^{N} \phi(x_l)^T \phi(x_m) - \frac{1}{N} \sum_{l=1}^{N} \phi(x_l)^T \phi(x_n) + \frac{1}{N^2} \sum_{l=1}^{N} \sum_{k=1}^{N} \phi(x_l)^T \phi(x_k)$$

$$= K(n,m) - \overset{\text{"column sum"}}{\frac{1}{N} \sum_{l=1}^{N} K(l,m)} - \overset{\text{"row sum"}}{\frac{1}{N} \sum_{l=1}^{N} K(n,l)} + \underset{\text{"column \& row sum"}}{\frac{1}{N^2} \sum_{l=1}^{N} \sum_{k=1}^{N} K(l,k)}$$

Consider $\mathbb{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$. Then $\mathbb{1}\mathbb{1}^T = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$

and $(\mathbb{1}\mathbb{1}^T K)(n,m) = \sum_{l=1}^{N} \underbrace{(\mathbb{1}\mathbb{1}^T)(n,l)}_{1} K(l,m)$

$= \sum_{l=1}^{N} K(l,m) =$ "column sum".

Similarly, $(K\mathbb{1}\mathbb{1}^T)(n,m) = \sum_{l=1}^{N} K(n,l) =$ "row sum".

Therefore,

$$\boxed{\Phi\Phi^T = K - \frac{1}{N}\mathbb{1}\mathbb{1}^T K - \frac{1}{N} K\mathbb{1}\mathbb{1}^T + \frac{1}{N^2}\mathbb{1}\mathbb{1}^T K\mathbb{1}\mathbb{1}^T}$$

kernel regression

kernel regression

- In ridge regression, suppose our model is $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})$:

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

- Taking gradient w.r.t. $\mathbf{w}$ yields:

$$\frac{\partial J(w)}{\partial w} = \sum_{n=1}^{N}\left(w^{T}\phi(x_n) - t_n\right)\phi(x_n) + \lambda w \overset{\text{set}}{=\!=} 0$$

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}\phi(\mathbf{x}_n) = \sum_{n=1}^{N} a_n\phi(\mathbf{x}_n) = \mathbf{\Phi}^{\mathrm{T}}\mathbf{a}$$

Let $a_n = -\frac{1}{\lambda}\left(w^{T}\phi(x_n) - t_n\right)$

kernel regression

- In ridge regression, suppose our model is $y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x})$:

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

- Taking gradient w.r.t. $\mathbf{w}$ yields: (Cont'd from previous slide)

Let $\Phi = \begin{bmatrix} \phi(x_1)^T \\ \phi(x_2)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix}$, $a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}\phi(\mathbf{x}_n) = \boxed{\sum_{n=1}^{N} a_n\phi(\mathbf{x}_n) = \mathbf{\Phi}^{\mathrm{T}}\mathbf{a}}$$

kernel regression

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

- With $\mathbf{w} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{a}$

$$\sum_{n=1}^{N}\left(w^T\phi(x_n) - t_n\right)^2 = \begin{bmatrix} w^T\phi(x_1) - t_1 \\ w^T\phi(x_2) - t_2 \\ \vdots \\ w^T\phi(x_N) - t_N \end{bmatrix}^T \begin{bmatrix} w^T\phi(x_1) - t_1 \\ w^T\phi(x_2) - t_2 \\ \vdots \\ w^T\phi(x_N) - t_N \end{bmatrix}$$

$$= (\Phi w - t)^T(\Phi w - t) \qquad \text{where } t = (t_1, t_2, \cdots, t_N)^T$$

$$= (\Phi\Phi^T a - t)^T(\Phi\Phi^T a - t)$$

Hence, $$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\mathbf{\Phi}^{\mathrm{T}}\mathbf{a}$$

kernel regression

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{a}$$

$$K_{nm} = \phi(\mathbf{x}_n)^{\mathrm{T}}\phi(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m)$$

"Gram matrix"

$K = \Phi \Phi^T$

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a}$$

kernel regression

- From $\boxed{a_n = -\frac{1}{\lambda}\left\{\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) - t_n\right\} \qquad \mathbf{w} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{a}}$, we have

- Therefore,

$$\mathbf{a} = (\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\mathbf{t}$$

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{\Phi}\phi(\mathbf{x}) = \mathbf{k}(\mathbf{x})^{\mathrm{T}}(\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\mathbf{t}$$

Questions?

Reference

- *Kernel methods:*
  - *[Al] Ch.13.5-13.7*
  - *[Bi] Ch.7.1, 6.1-6.2*
- *Kernel PCA:*
  - *[HaTF] Ch.14.5.4*
- *Gaussian processes:*
  - *[Bi] Ch.6.4.1-6.4.2, 6.4.5*
  - *[HaTF] Ch.5.8.1-5.8.2*