MTH 221

Fundamentals of Machine Learning

Batuhan Bardak

Lecture 7: Support Vector Machines

Date: 14.11.2023

Announcements

- Midterm date
  - December 1, 2023 from 18:40 to 20:40
  - Informatics Institute, II-02 and II-03
  - You are responsible for the topics we have covered and discussed in class until the exam (including the content shared by the guests in the Special Guest lecture and this paper as well https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf)

Announcements

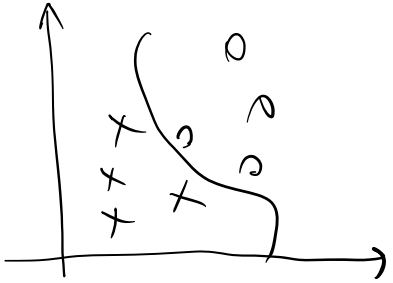
- Special Guest Lecture, On November 21, 2023
  - Barkın Sarıtaş, Senior Sales Strategy, Planning & Operations Manager @ Github
    - Transforming Data into Decisions: The Anatomy of a Corporate Analytics Project
  - Yunuscan Koçak, Machine Learning Scientist @ Booking
    - From Jupyter to Production: A Pragmatic Guide to Deploying Machine Learning Models
  - Salim Tütüncü, Senior Solutions Architect @ AWS
    - Generative AI: How to Leverage AWS ML Services to Drive Value

Announcements

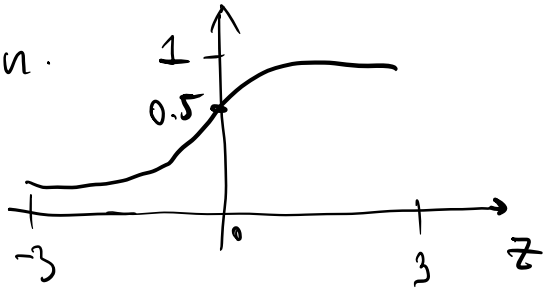
- Project Progress Report Deadline
  - December 3, 2023 by 23:59
  - The outline should be
    - EDA
    - Experimental Reesults
    - Future Work
    - At least 3 pages long in PDF Format
    - Send to TA while keeping me in the CC.

Recap Logistic Regression

* classification.



sigmoid function.



$$g(z) = \frac{1}{1+e^{-z}}$$

$$f_{\vec{w},b}(\vec{x})$$

$$z = \vec{w}\vec{x} + b.$$

$$g(z) = \frac{1}{1+e^{-z}}$$

Recap Gradient Descent

Linear —
New —
Logistic.

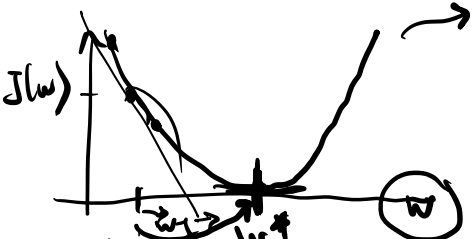
$$J(w,b) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 \rightarrow \text{linear reg.}$$

$$\times$$

$$J(\vec{w},b) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log\left(f_{\vec{w},b}(\vec{x}^{(i)}) \right) + \left(1 - y^{(i)} \right) \log\left(1 - f_{\vec{w},b}(\vec{x}^{(i)}) \right) \right]$$

1 0

- find the best $\vec{w}$
- w



→ convex

$$\min_{w} J(w,b)$$

Repeat until converge {

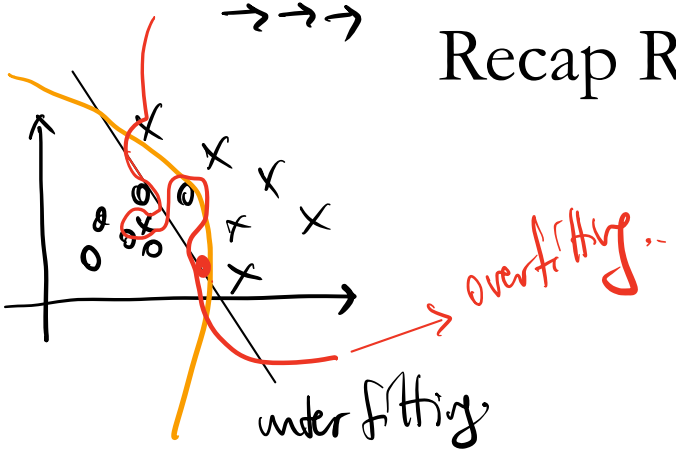
$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w},b)$$

w_j

}

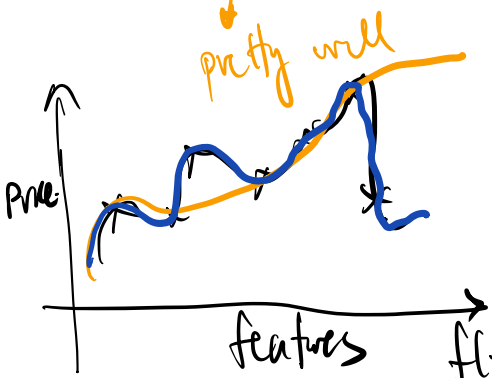
↓ learning rate

Recap Regularization



- collect more training data.
- select features ↘ 1 100

  feature selection algorithms.

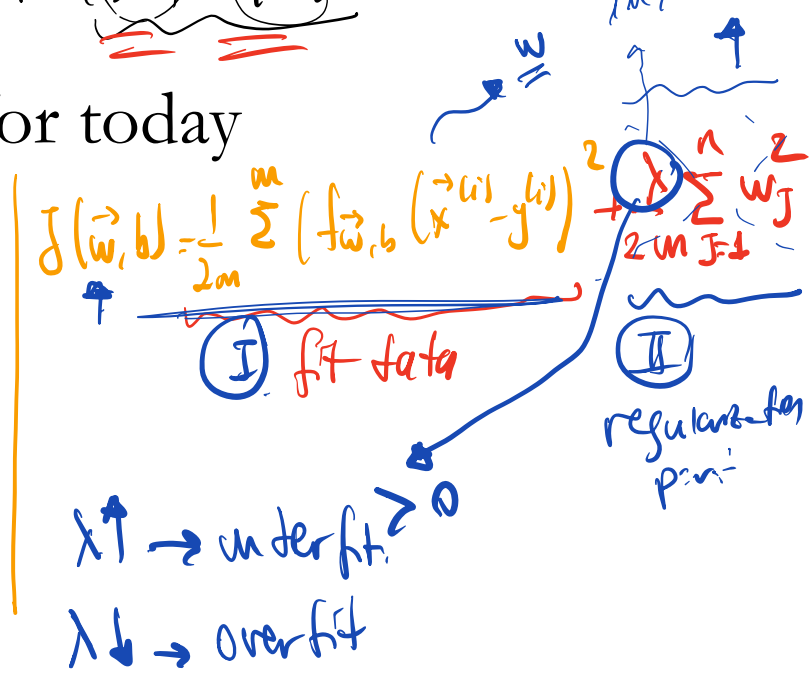


- Regularization.
  → Reduce the size of parameters w_j

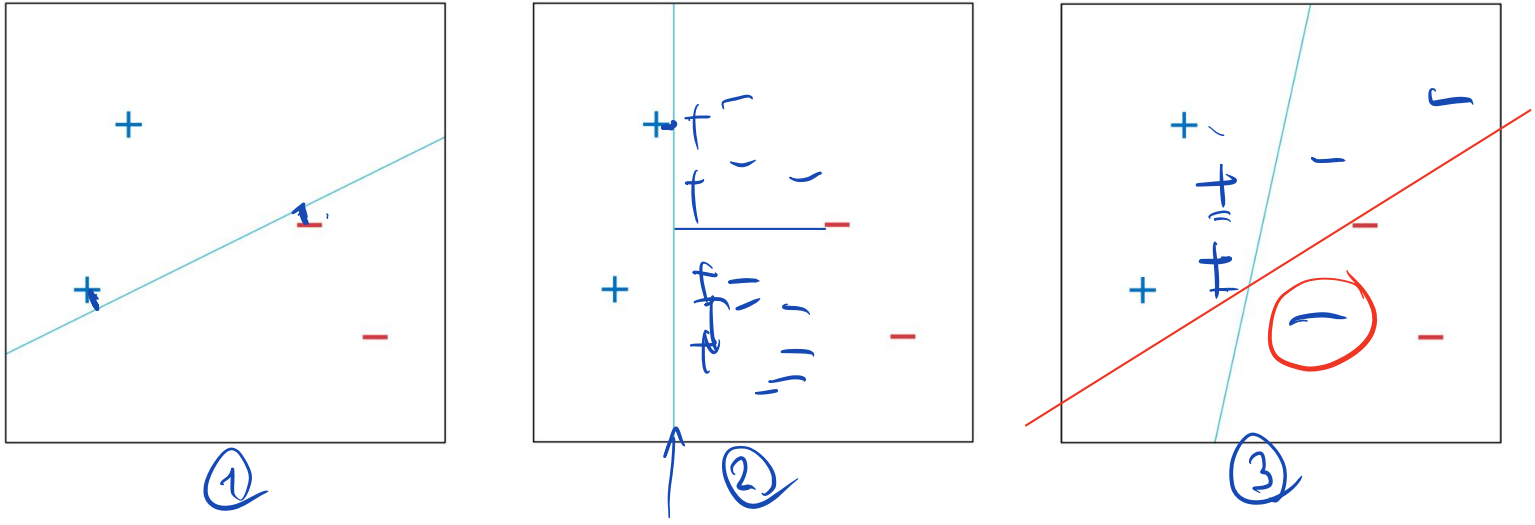
$f(x) = 28x - 385x^2 + 39x^3 - 174x^4 + 100$ ... 0

Plan for today

- SVM for linearly separable classes
- SVM for linearly inseparable classes
- SVM for nonlinear decision boundaries
  - Kernel functions

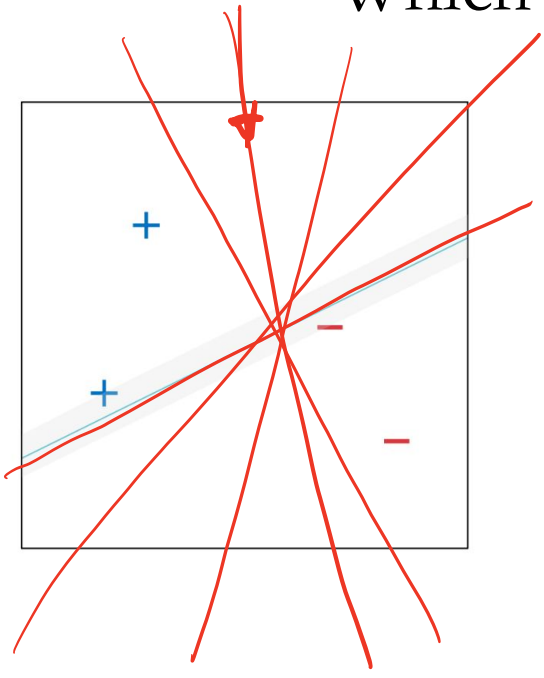


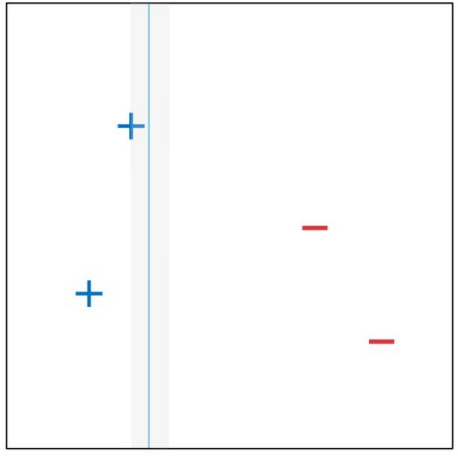
Which linear separator is best?

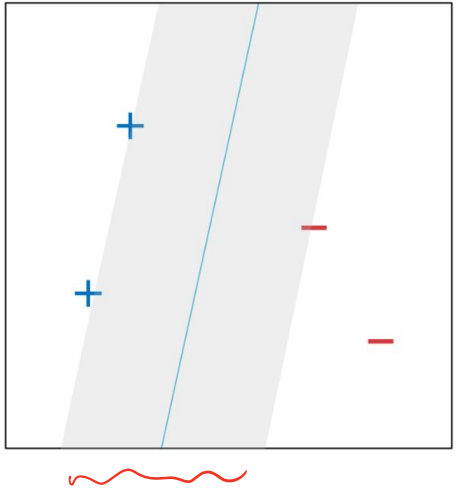


Slide Credit: Henry Chai

Which linear separator is best?







Slide Credit: Henry Chai

SVM

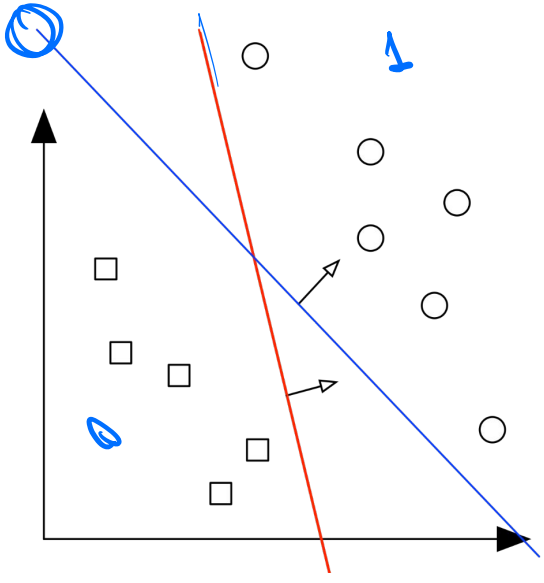
- The Support Vector Machine (SVM) is a linear classifier that can be viewed as an extension of the Perceptron.
- The Perceptron guaranteed that you find a hyperplane if it exists. The SVM finds the **maximum margin** separating hyperplane.

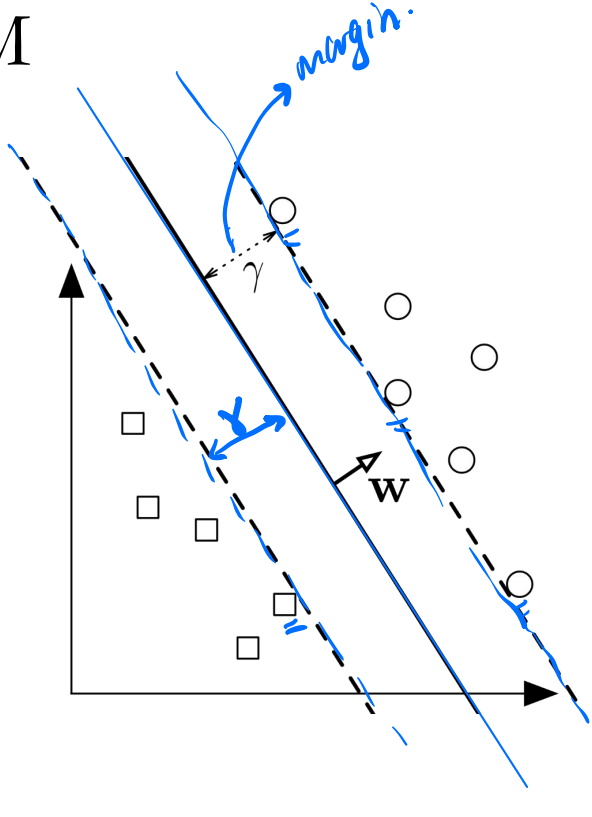
Maximal Margin Linear Separators

- The margin of a linear separator is the distance between it and the nearest training data point.
- Questions:
  - How can we efficiently find a maximal-margin linear separator?
  - Why are linear separators with larger margins better?
  - What can we do if the data is not linearly separable?

Slide Credit: Henry Chai

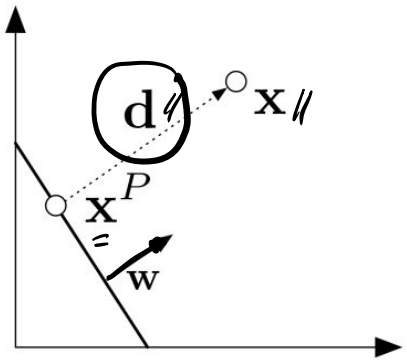
SVM





Geometric Margin

Hyperplane defined by (w, b), i.e.,
$\{x : w^\top x + b = 0\}$



Fact 1. $x - x^P$ is parallel to w:

$$x - x^p = \alpha w$$

Fact 2. x^p is on the hyperplane:

$$w^\top x^P + b = 0$$

Fact 1 + fact 2 implies:

$$w^\top(x - \alpha w) + b = 0 \rightarrow \alpha = (w^\top x + b)/\|w\|_2^2$$

Final step:

$$d = \|x - x^p\|_2 = \|\alpha w\|_2 = \frac{|w^\top x + b|}{\|w\|_2}$$

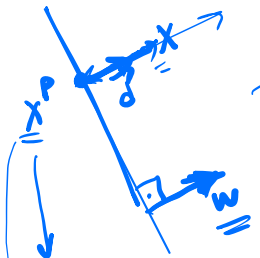
Slide Credit: Cornell, CS5785-2020

→ hyperplane

Geometric Margin

$H_{w,b} = \{x : w^T x + b = 0\}$

find the distance btw point "x" and hyperplane.



$\vec{x}_p = \vec{x} - \vec{d}$

$w^T x_p + b = 0$

$= w^T(\vec{x} - \vec{d}) + b$

$= w^T(\vec{x} - \alpha\vec{w}) + b = 0$

* "d" must be the rescaled version of $\vec{w}$.

$\vec{d} = \alpha \vec{w}$

$\alpha = \frac{w^T x + b}{w^T w}$

- length → norm of d ⇒ $\|x\|_2 = \sqrt{x_1^2 + x_2^2}$

⇒ $\|d\|_2 = \sqrt{d^T d}$

$= \sqrt{\alpha^2 w^T w}$

$= \alpha \sqrt{w^T w}$

$= \frac{w^T x + b}{w^T w} \sqrt{w^T w}$

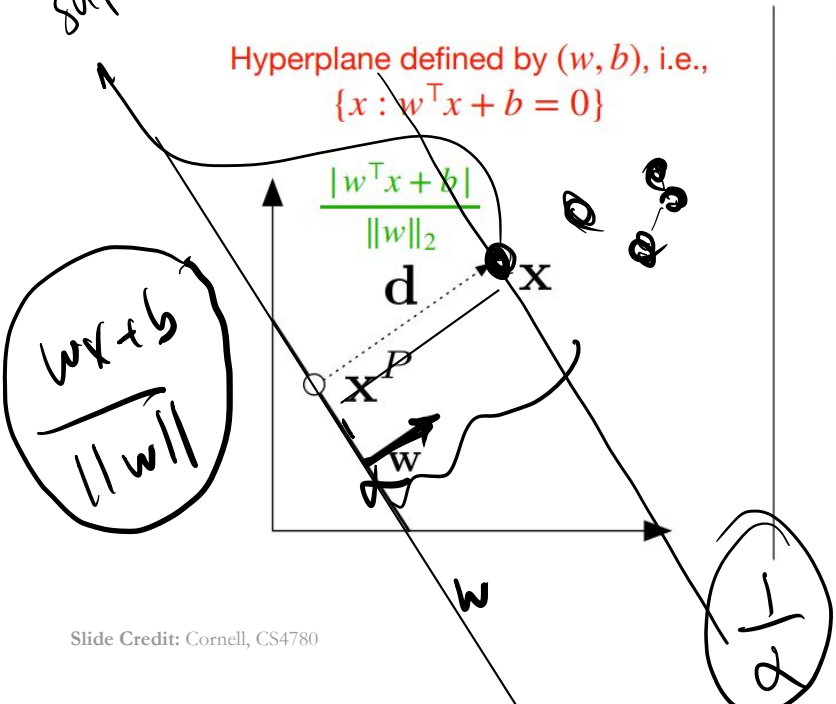
$= \frac{w^T x + b}{\sqrt{w^T w}}$

$= \frac{w^T x + b}{\|w\|_2}$

d

distance of a point x to the hyperplane

Geometric Margin is Scale Invariant



We scale (w, b) by a constant $\gamma \in \mathbb{R}^+$

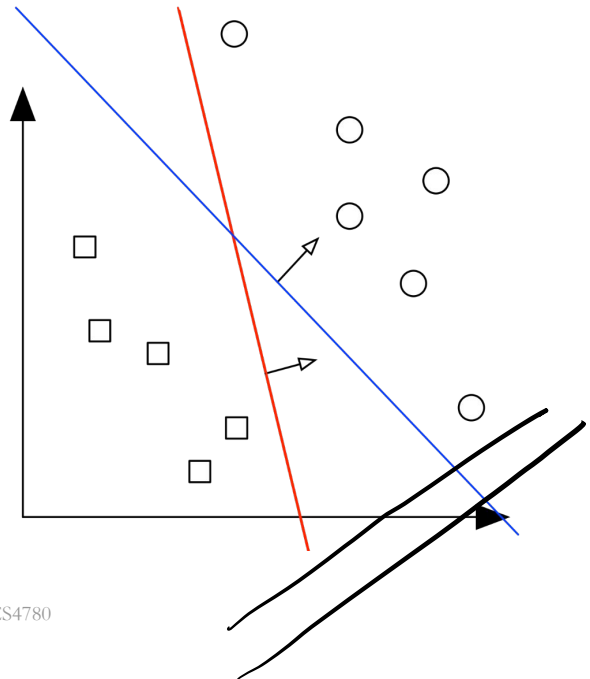
Q: is the hyperplane defined by $(\gamma w, \gamma b)$ different?

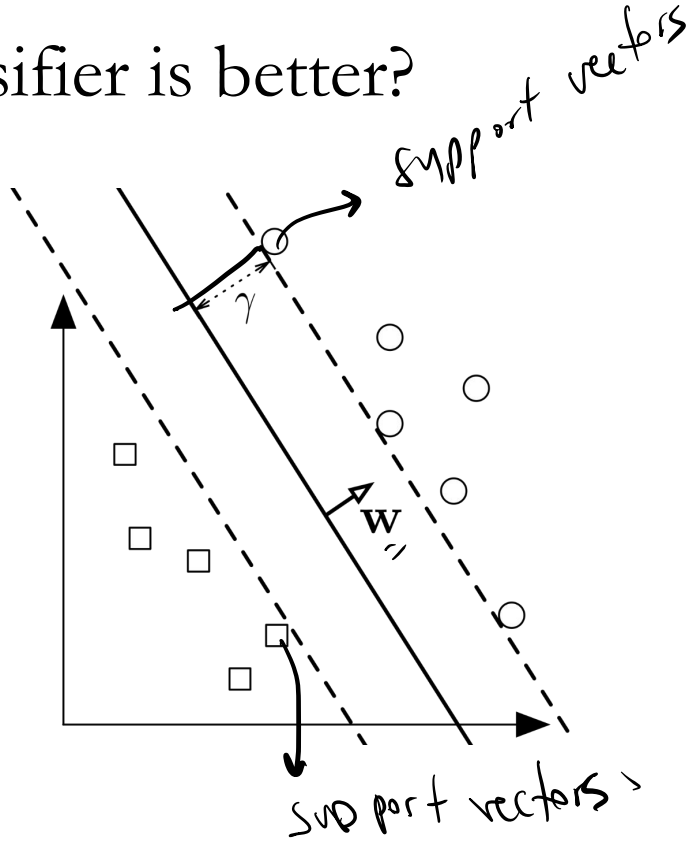
Q: does the margin change?

Hyperplane & Geometric margin are **scale invariant**!

Slide Credit: Cornell, CS4780

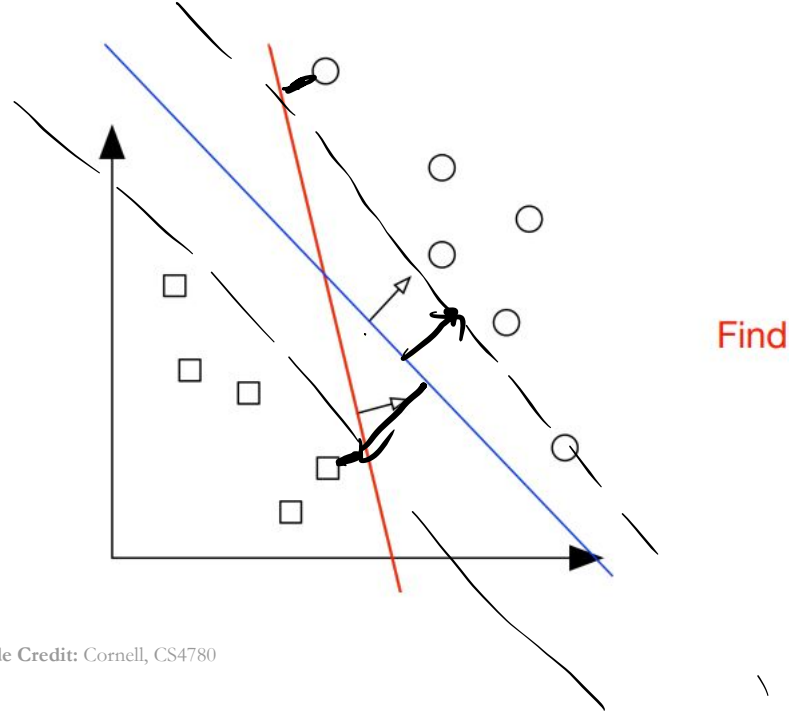
Which linear classifier is better?





Slide Credit: Cornell, CS4780

Max Margin Classifier

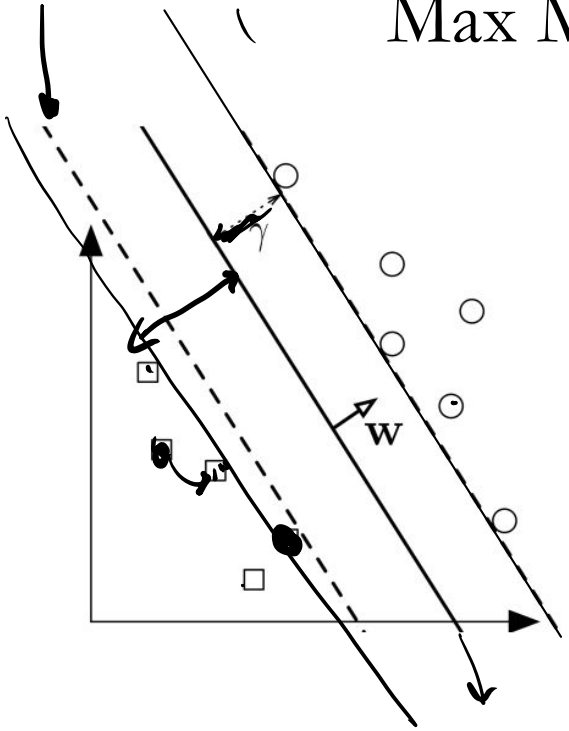


The Goal of SVM:

Find a hyperplane that has the largest Geometric margin

Slide Credit: Cornell, CS4780

Max Margin Classifier



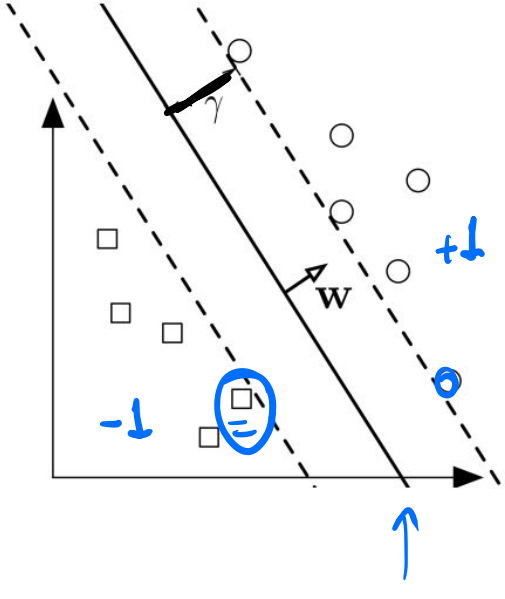
Given a linearly separable dataset $\{x_i, y_i\}_{i=1}^n$, the minimum geometric margin is defined as

$$\gamma(w,b) := \min_{x_i \in \mathcal{D}} \frac{|x_i^\top w + b|}{\|w\|_2}$$

Goal: we want to find (w, b) s.t. it separates the data, and maximizes $\gamma(w,b)$

$$\max_{w,b} \gamma(w,b)$$

Max Margin Classifier



We want to find (w, b) s.t. it separates the data, and maximizes $\gamma(w, b)$

Constraint

$$\max_{w,b} \gamma(w, b)$$

$$\text{s.t.} \forall i,\ y_i(w^\top x_i + b) \geq 0$$

Plug in the def of $\gamma(w, b)$:

$$\max_{w,b} \frac{1}{\|w\|_2} \min_{x_i} |w^\top x_i + b|$$

$$\text{s.t.} \forall i,\ y_i(w^\top x_i + b) \geq 0$$

Slide Credit: Cornell, CS4780

SVM for separable data: Max Margin Classifier

We want to find (w, b) s.t. it separates the data, and maximize $\gamma(w, b)$

$$\min_{w,b} w^T w$$

$$\text{s.t.} \; \forall i, \; y_i(w^\top x_i + b) \geq 0$$

$$\min_i |w^\top x_i + b| = 1$$

Recall that margin & hyperplane is scale invariant

For any (w, b), we can always scale it by some constant to have

$$\min_{x_i} |w^\top x_i + b| = 1$$

Without loss of generality, let's just focus on such (w, b) pairs with $\min_{x_i} |w^\top x_i + b| = 1$

SVM for separable data: Max Margin Classifier

$$\min_{w,b} \|w\|_2^2$$

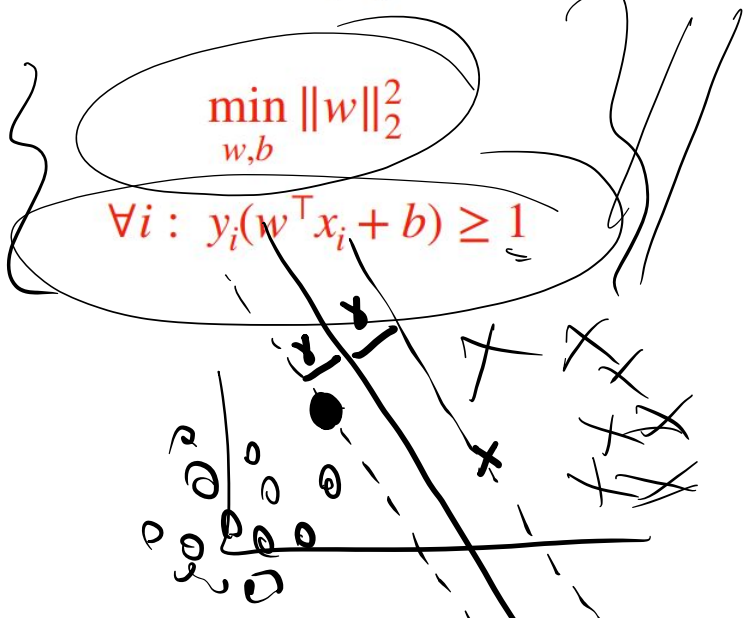
$$\text{s.t.} \; \forall i, \; y_i(w^\top x_i + b) \geq 0$$

$$\min_i |w^\top x_i + b| = 1$$

We can further simplify the constraint

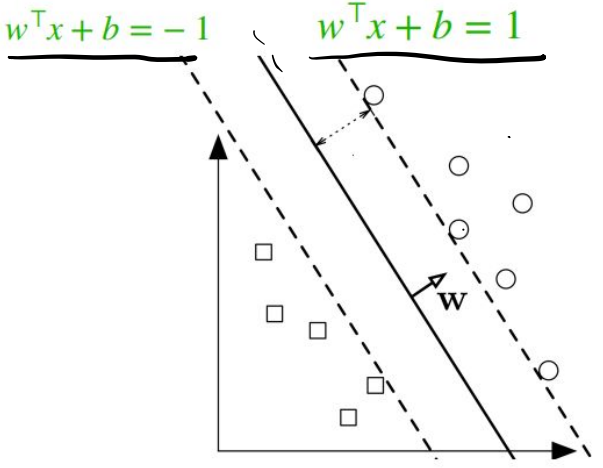
$$\min_{w,b} \|w\|_2^2$$

$$\forall i : \; y_i(w^\top x_i + b) \geq 1$$



Slide Credit: Cornell, CS4780

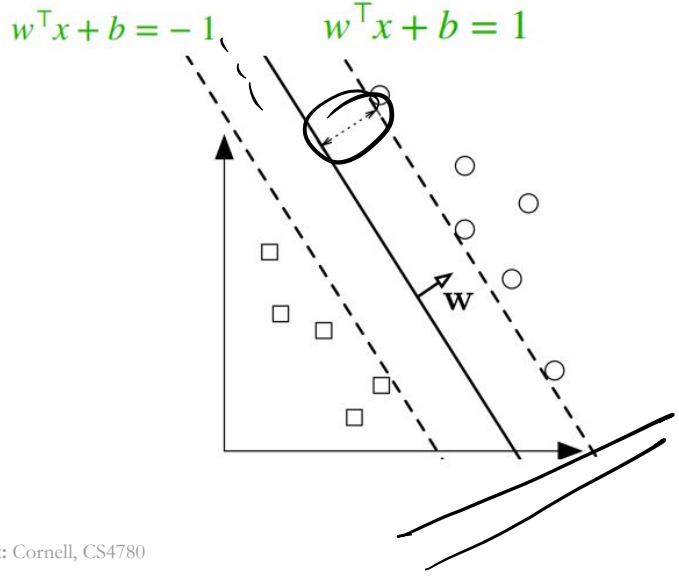
Max Margin Classifier



for the optimal (w, b) pair, points x_i such that $y_i(w^\top x_i + b) = 1$ are called **support vectors**

↳ Support vectors

Support Vectors



for the optimal (w, b) pair, points x_i such that $y_i(w^\top x_i + b) = 1$ are called **support vectors**

Slide Credit: Cornell, CS4780

SVM for non-separable data

If data is not linearly separable, then **there is no** (w, b) can satisfy $\forall i: \; y_i(w^\top x_i + b) \geq 1$

= infeasible.

Idea: introducing slack variables to relax the constraint, i.e., find (w, b, ξ_i), s.t,

$$\forall i: \; y_i(w^\top x_i + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \forall i$$

soft constraints.

Q: does this always has a feasible solution?

SVM for non-separable data

Idea: introducing slack variables to relax the constraint, i.e., find (w, b, ξ_i), st,

$$\forall i: \; y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

We still want our margin to be somewhat large, i.e., we want slack variables to be as small as possible

$$\min_{w,b,\xi} \|w\|_2^2 + c \sum_{i=1}^{n} \xi_i$$

Penalizing large slacks

$$\forall i: \; y_i(w^\top x_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0$$

SVM for non-separable data

$$\min_{w,b,\xi_i} \|w\|_2^2 + c \sum_{i=1}^{n} \xi_i$$

Penalizing large slacks

$$\forall i: \; y_i(w^\top x_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0$$

We can turn this constrained opt to a unconstraint opt w/ a single objective.

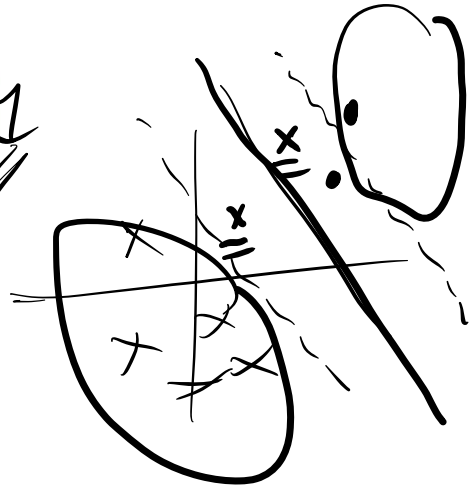
Q: For any fixed (w, b) pair, how to set ξ_i, such that the obj is minimized?

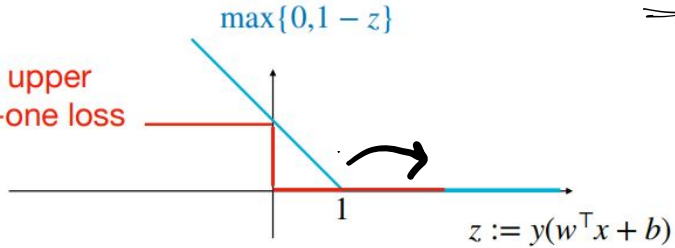
A: set $\xi_i = \max\{0, 1 - y_i(w^\top x_i + b)\}$

SVM for non-separable data

$$\min_{w,b} \|w\|_2^2 + c \sum_{i=1}^{n} \max\left\{0, 1 - y_i(w^\top x_i + b)\right\}$$

Hinge loss





Hinge loss starts penalizing when functional margin falls below 1

Slide Credit: Cornell, CS4780

SVM for non-separable data

$$\min_{w,b} \|w\|_2^2 + c \sum_{i=1}^{n} \max\left\{0, 1 - y_i(w^\top x_i + b)\right\}$$

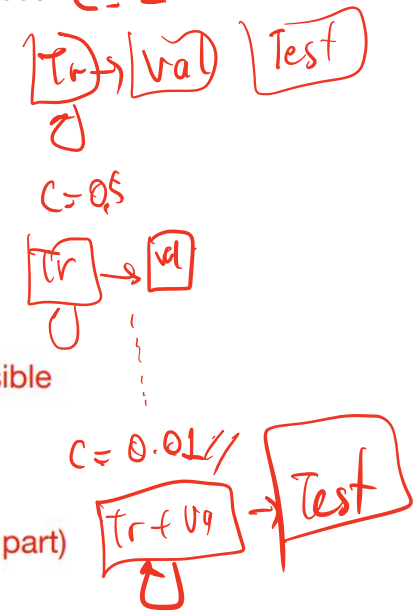
Trades off $\|w\|_2^2$ and functional margins over data

When $c \to +\infty$:

forcing $y_i(w^\top x_i + b) \geq 1$ for as many data points as possible

When $c \to 0^+$:

The solution $w \to \mathbf{0}$ (i.e., we do not care about hinge loss part)



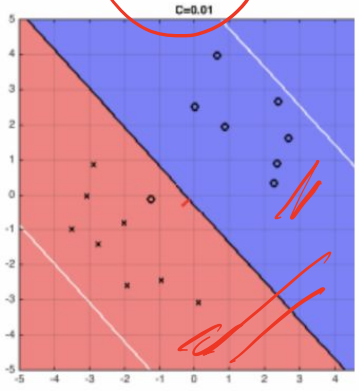
Slide Credit: Cornell, CS4780

SVM for non-separable data

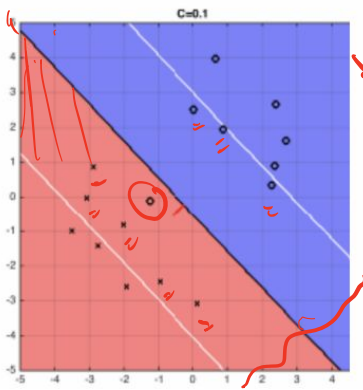
$$\min_{w,b} \|w\|_2^2 + c \sum_{i=1}^{n} \max\left\{0, 1 - y_i(w^\top x_i + b)\right\}$$

Trades off $\|w\|_2^2$ and functional margins over data

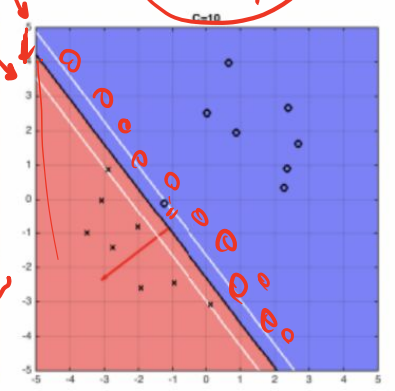
C = 0.01



C = 1



C =10



Slide Credit: Cornell, CS4780

Summary

1. SVM for linearly separable data

$$\min_{w,b} \|w\|_2^2$$

$$\forall i: \; y_i(w^\top x_i + b) \geq 1$$

2. SVM for non-separable data

$$\min_{w,b} \|w\|_2^2 + c \sum_{i=1}^{n} \max\left\{0, 1 - y_i(w^\top x_i + b)\right\}$$

Hinge loss

Slide Credit: Cornell, CS4780

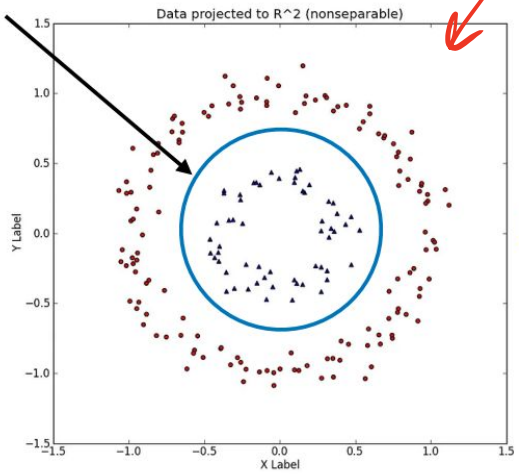
Kernel SVMs

- Go from linear models to more powerful nonlinear ones.
- Keep convexity (ease of optimization).
- Generalize the concept of feature engineering.

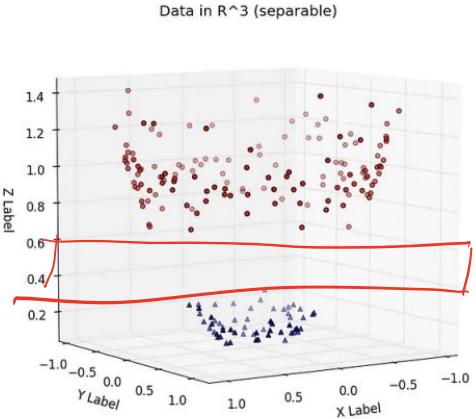
Slide Credit: W4995, Andreas Müller

Kernels

Goal: Non-linear decision boundary



Our approach



ϕ

$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$

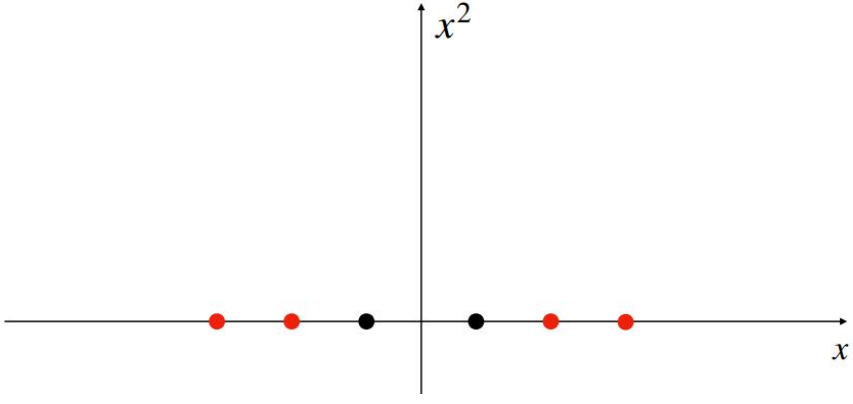
kernel trick

Slide Credit: Cornell, CS4780

Feature mapping

Define $\phi(\mathbf{x}) \in \mathbb{R}^m$ as a feature mapping (often $m > d$)

Ex 1: $\mathbf{x} \in \mathbb{R}$, $\phi(\mathbf{x}) = [x, x^2]^\top \in \mathbb{R}^2$



Feature mapping

Define $\phi(\mathbf{x}) \in \mathbb{R}^m$ as a feature mapping (often $m > d$)

Ex 2: quadratic feature mapping ϕ

$$\mathbf{x} = [x_1, x_2]^\top,$$

$$\phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2]^\top$$

Feature mapping

Define $\phi(\mathbf{x}) \in \mathbb{R}^m$ as a feature mapping (often $m > d$)

Ex 2: cubic feature mapping ϕ

$$\mathbf{x} = [x_1, x_2]^\top,$$

$$\phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, x_1^3, x_2^3, x_1 x_2^2, x_1^2 x_2]^\top$$

Q: in general, for $\mathbf{x} \in \mathbb{R}^d$, and a p-th order polynomial feature ϕ, what's the dimension of $\phi(\mathbf{x})$?

at least $\binom{d}{p}$

Dim of $\phi(\mathbf{x})$ can be very large!

Slide Credit: Cornell, CS4780

Feature mapping

Define $\phi(\mathbf{x}) \in \mathbb{R}^m$ as a feature mapping (often $m > d$)

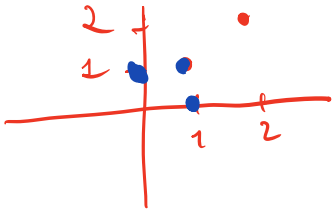
Ex 2: cubic feature mapping ϕ

$$\mathbf{x} = [x_1, x_2]^\top,$$

$$\phi(\mathbf{x}) = [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, x_1^3, x_2^3, x_1 x_2^2, x_1^2 x_2]^\top$$

class = 1 : (1, 1) (2, 2)

class = 2 : (1, 0) (0, 1)



Slide Credit: Cornell, CS4780

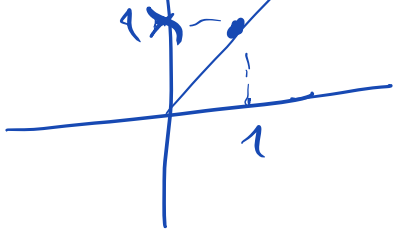
Q: in general, for $\mathbf{x} \in \mathbb{R}^d$, and a p-th order polynomial feature ϕ, what's the dimension of $\phi(\mathbf{x})$?

at least $\binom{d}{p}$

Dim of $\phi(\mathbf{x})$ can be very large!

$$K(x, x') = (1 + x \cdot x')^2$$

$z = (z_1, z_2, z_3) = (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$



Example of Kernels

Point (1,1) becomes:

$(1^2, \sqrt{2}\cdot 1\cdot 1, 1^2) = (1, \sqrt{2}, 1)$

$(1, 0)$

$(1^2, \sqrt{2}\cdot 1\cdot 0, 0^2) = (1, 0, 0)$

$$k_{\text{linear}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$$

$$k_{\text{poly}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

$$k_{\text{rbf}}(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$$

$$k_{\text{sigmoid}}(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \mathbf{x}^T \mathbf{x}' + r)$$

$$k_{\cap}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{p} \min(x_i, x_i')$$

- If k and k' are kernels, so are $k + k', kk', ck', \ldots$

Slide Credit: W4995, Andreas Müller

Polynomial Kernel vs Features

$$k_{\text{poly}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

Primal vs Dual Optimization

Explicit polynomials $\rightarrow$ compute on `n_samples * n_features ** d`
Kernel trick $\rightarrow$ compute on kernel matrix of shape `n_samples * n_samples`

For a single feature:

$$(x^2, \sqrt{2}x, 1)^T (x'^2, \sqrt{2}x', 1) = x^2 x'^2 + 2xx' + 1 = (xx' + 1)^2$$

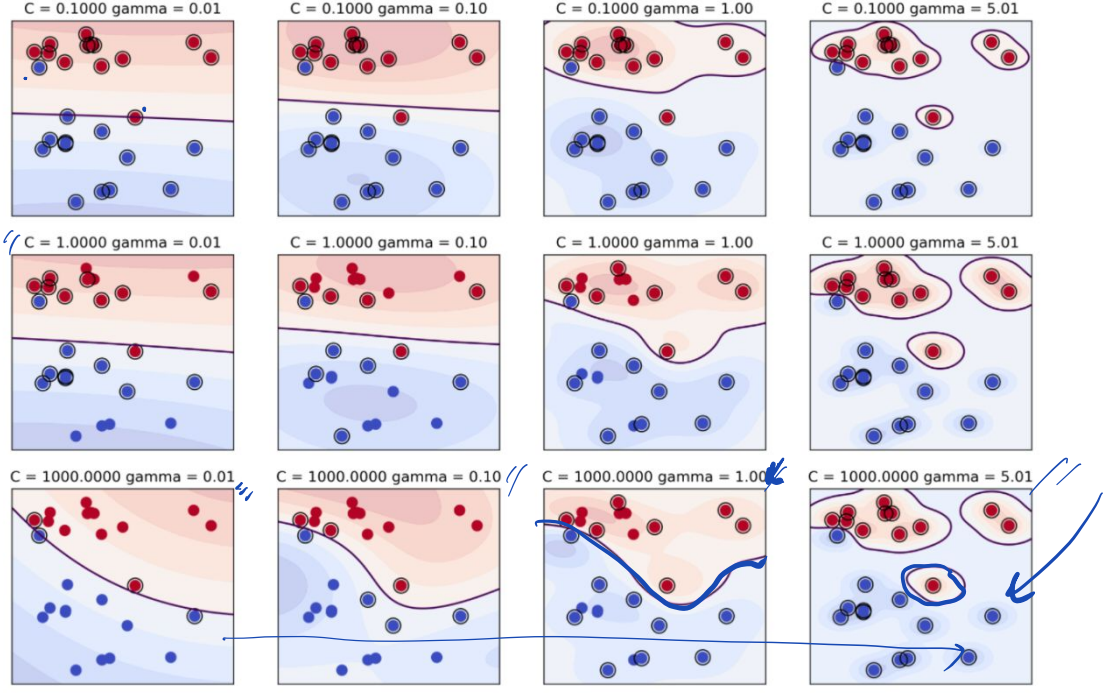
Slide Credit: W4995, Andreas Müller

Kernels in Practice

- Dual coefficients less interpretable
- Long runtime for “large” datasets (100k samples)
- Real power in infinite-dimensional spaces: rbf!
- Rbf is “universal kernel” - can learn (aka overfit) anything.

Slide Credit: W4995, Andreas Müller

Parameters for RBF Kernels



Slide Credit: W4995, A

Next Class:

Decision Trees & Ensemble Learning

- Guest Lecture
- Review / hands on example
- midterm
-