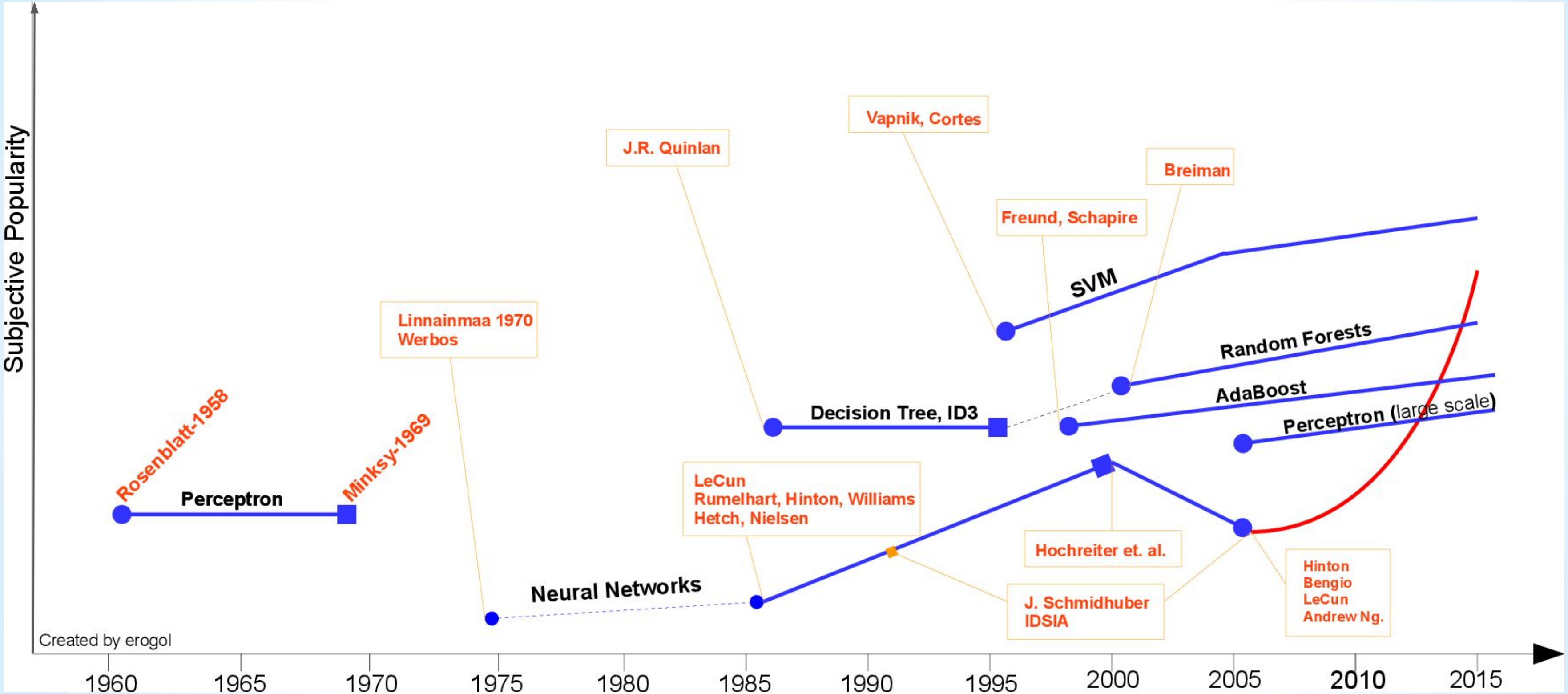
Support Vector Machines

Lior Sidi & Noa Lubin

Y-DATA SCHOOL OF DATA SCIENCE

Yandex

Models History

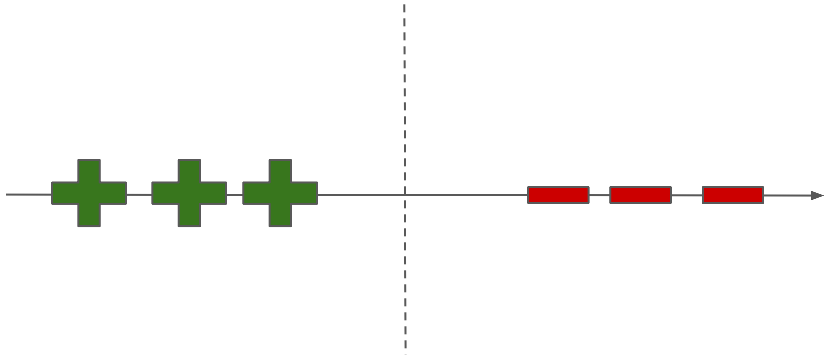


Motivation

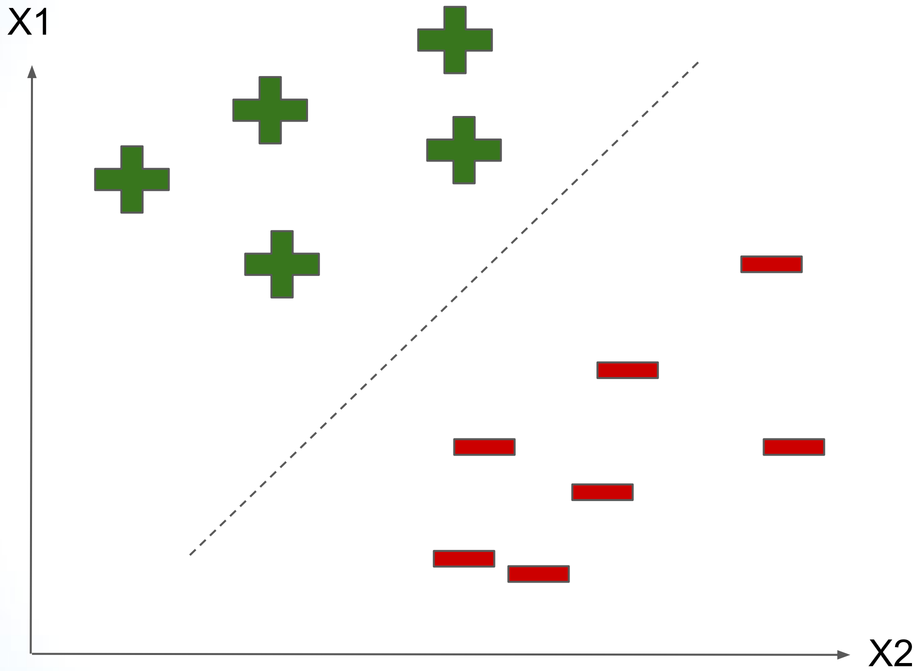
Good for difficult problems with limited data (<10K data points)

- Face Detection
- Text Classification
- Protein Fold and Remote Homology Detection
- Handwriting Recognition

Linear Classification - 1 dim



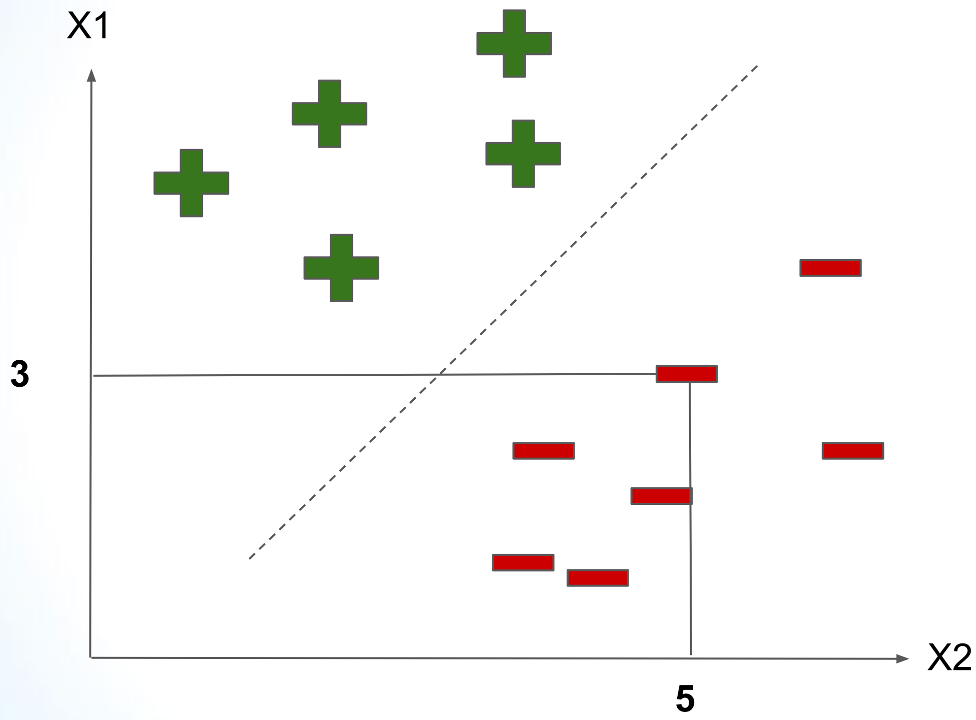
Linear Classification - 2 dim



w1*x1 = w2*x2 + b

2*x1 = 4*x2 - 3
2*x1 - 4*x2 + 3 = 0

Linear Classification - 2 dim

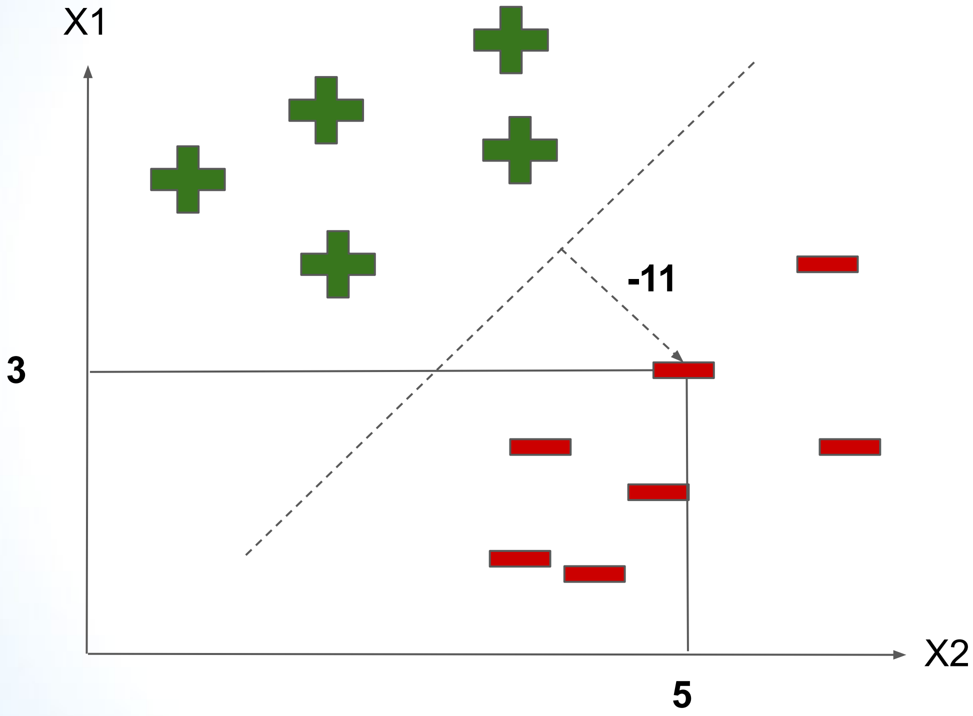


w1*x1 = w2*x2 + b

2*x1 = 4*x2 - 3
2*x1 - 4*x2 + 3 = 0

2*3 - 4*5 + 3 =

Linear Classification - 2 dim

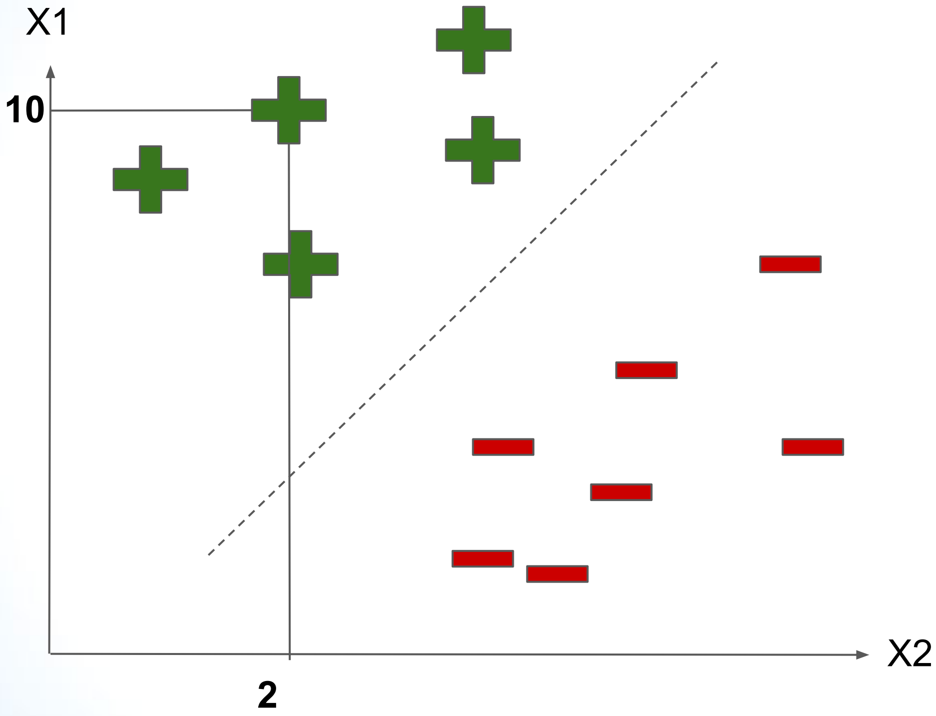


w1*x1 = w2*x2 + b

2*x1 = 4*x2 - 3
2*x1 - 4*x2 + 3 = 0

2*3 - 4*5 + 3 = 6 - 20 + 3 = -11

Linear Classification - 2 dim

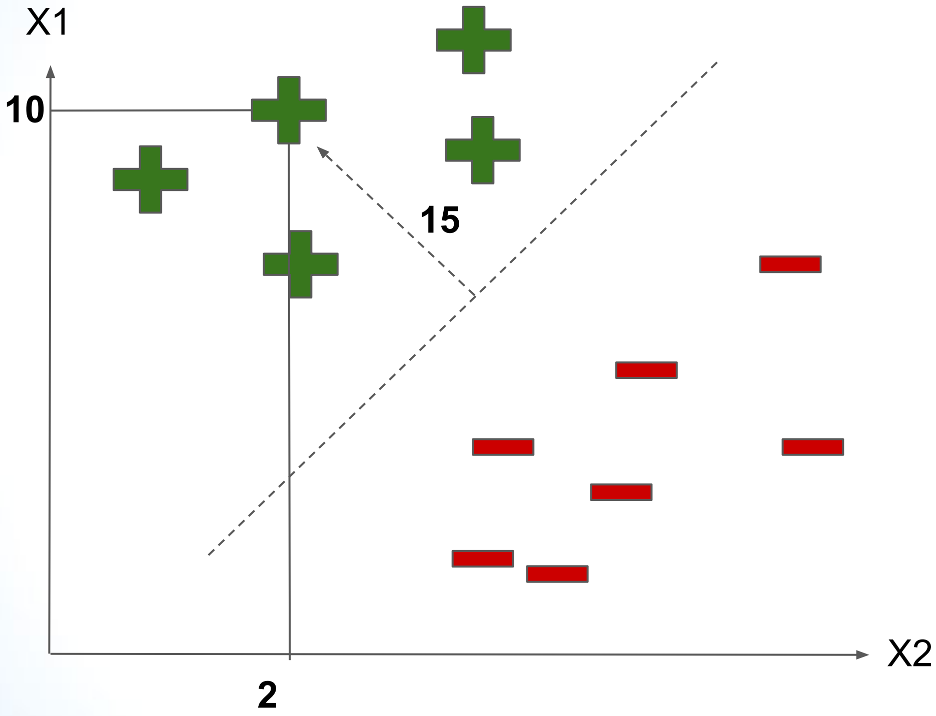


$w1*x1 = w2*x2 + b$

$2*x1 = 4*x2 - 3$
$2*x1 - 4*x2 + 3 = 0$

$2*3 - 4*5 + 3 = 6 - 20 + 3 = -11$
$2*10 - 4*2 + 3 = 20 - 8 + 3 = 15$

Linear Classification - 2 dim

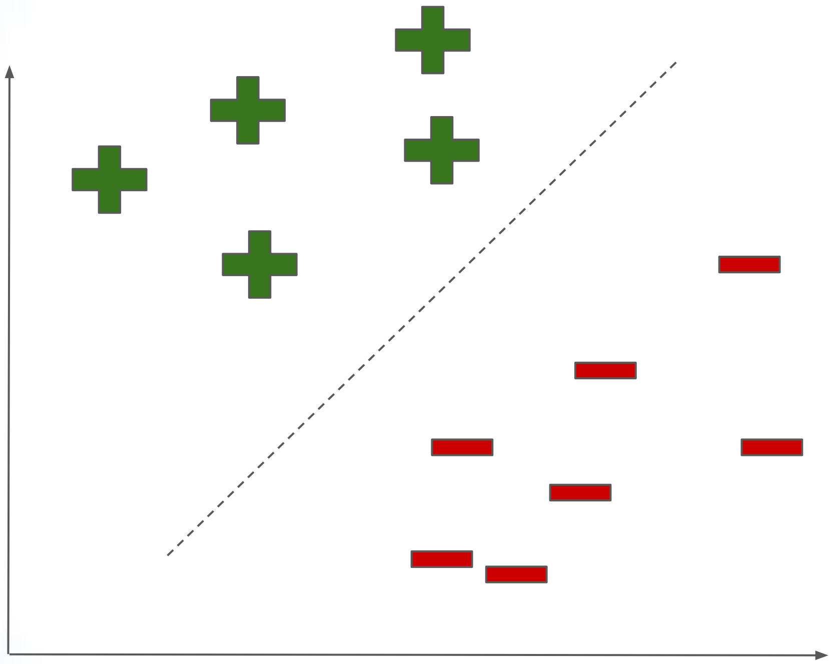


$w1*x1 = w2*x2 + b$

$2*x1 = 4*x2 - 3$
$2*x1 - 4*x2 + 3 = 0$

$2*3 - 4*5 + 3 = 6 - 20 + 3 = -11$
$2*10 - 4*2 + 3 = 20 - 8 + 3 = 15$

Linear Classification - 2 dim

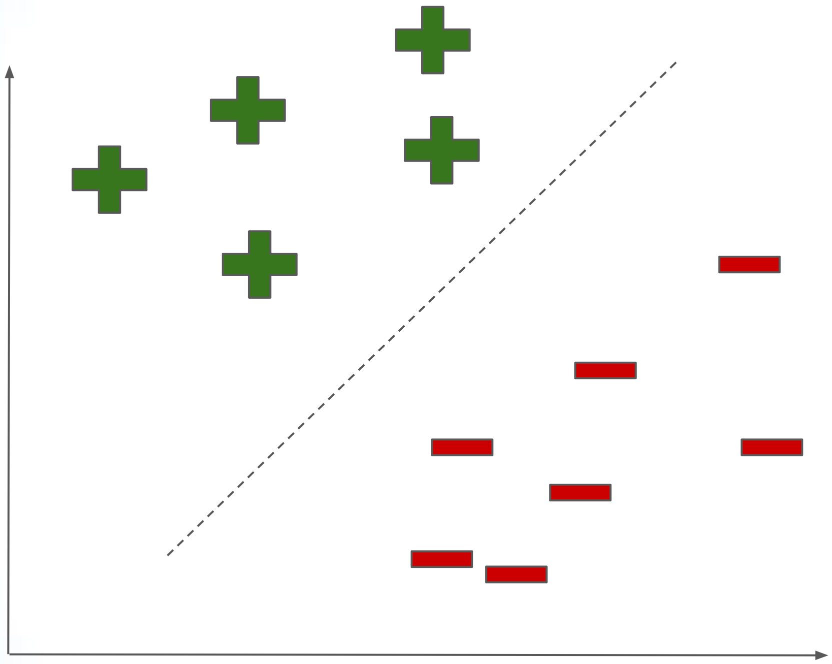


w1*x1 = w2*x2 + b

f(X,W) = w1*x1 + w2*x2 =

For simplicity We are going to eliminate the bias term Which can be added as a vectors of ones

Linear Classification - 2 dim

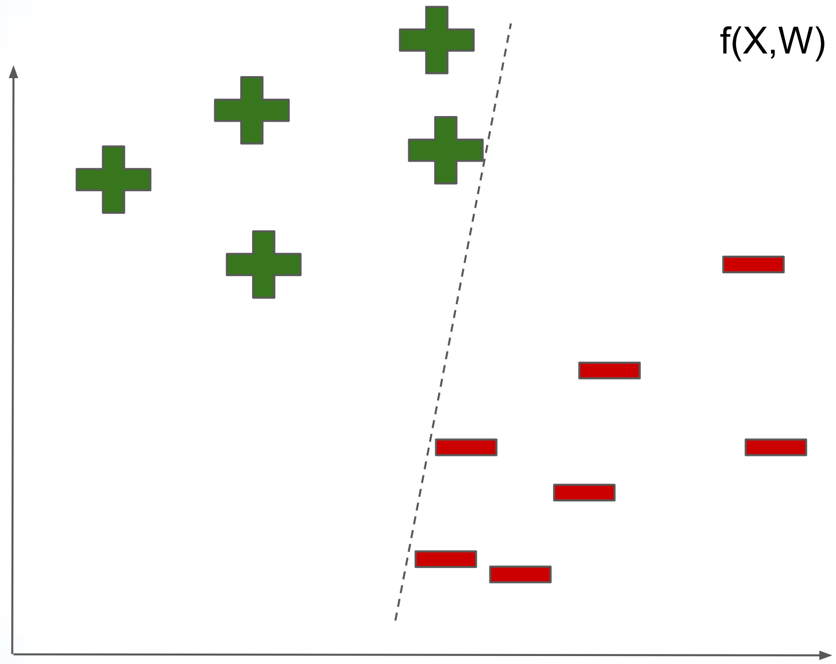


w1*x1 = w2*x2 + b

f(X,W) = w1*x1 + w2*x2 =
= ΣWX = W^{t}X = 0
=> sign(WX)

For simplicity We write
W^{t}X as WX

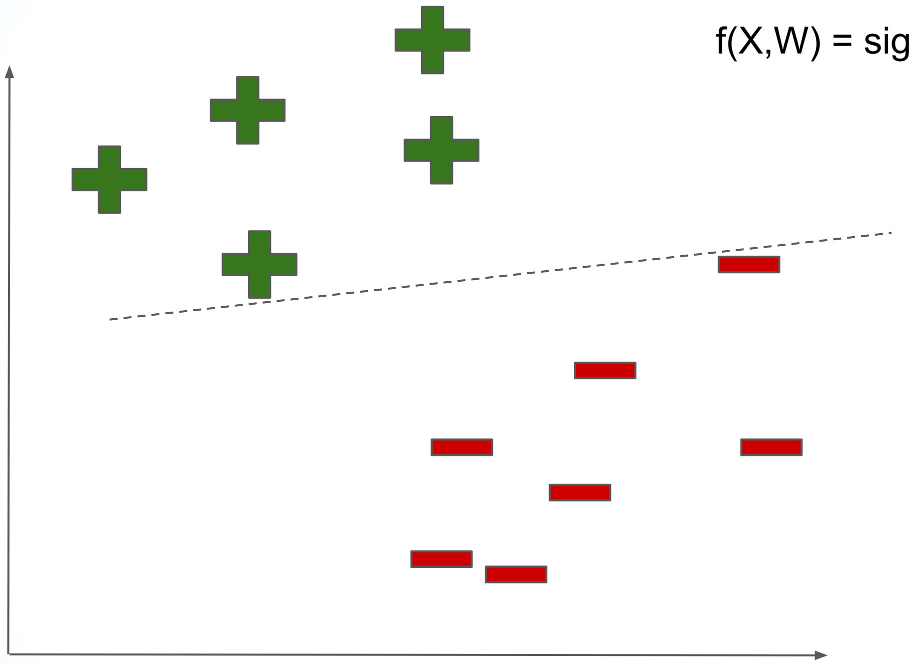
Linear Classification



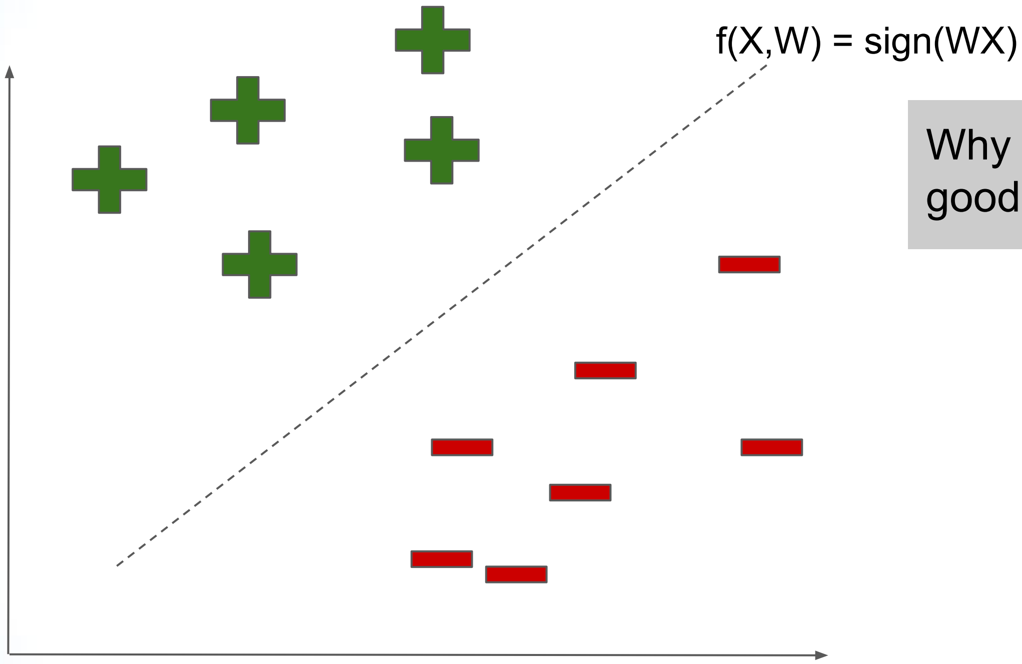
f(X,W) = sign(WX)

Linear Classification

$f(X,W) = \text{sign}(WX)$

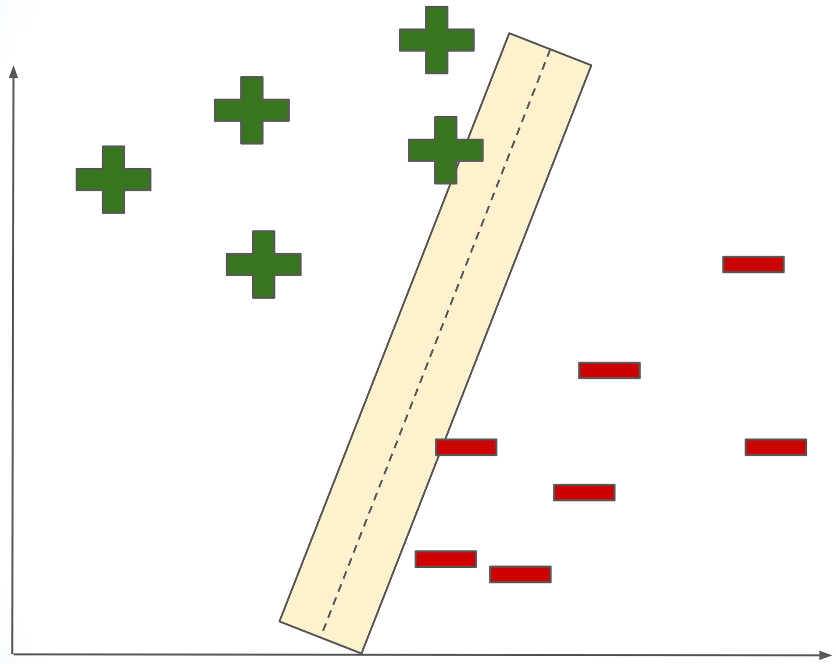


Linear Classification



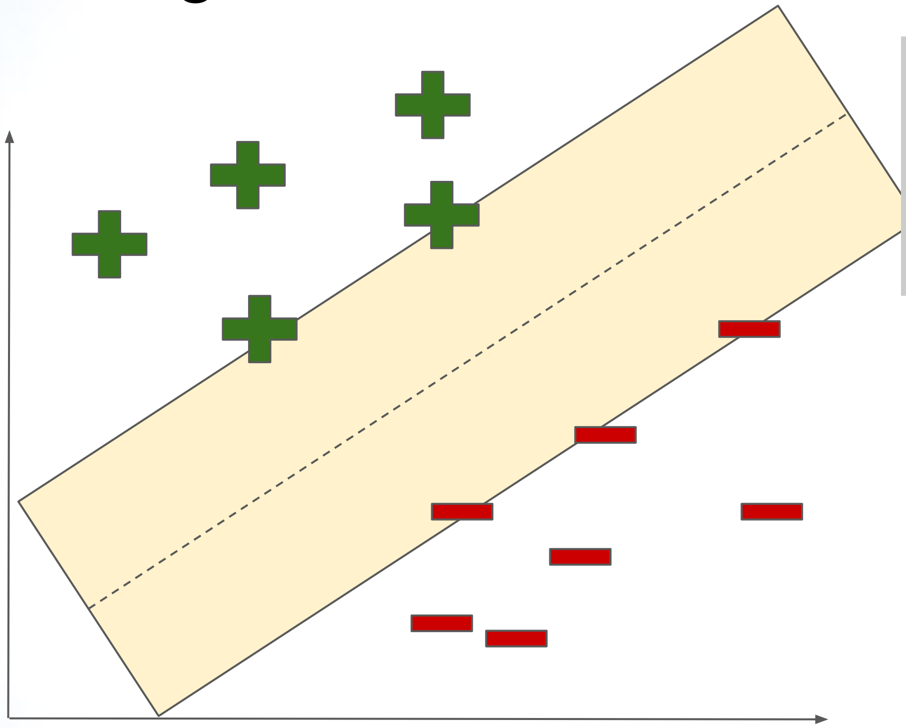
Why is this seems as a good separator?

Classifier Margin



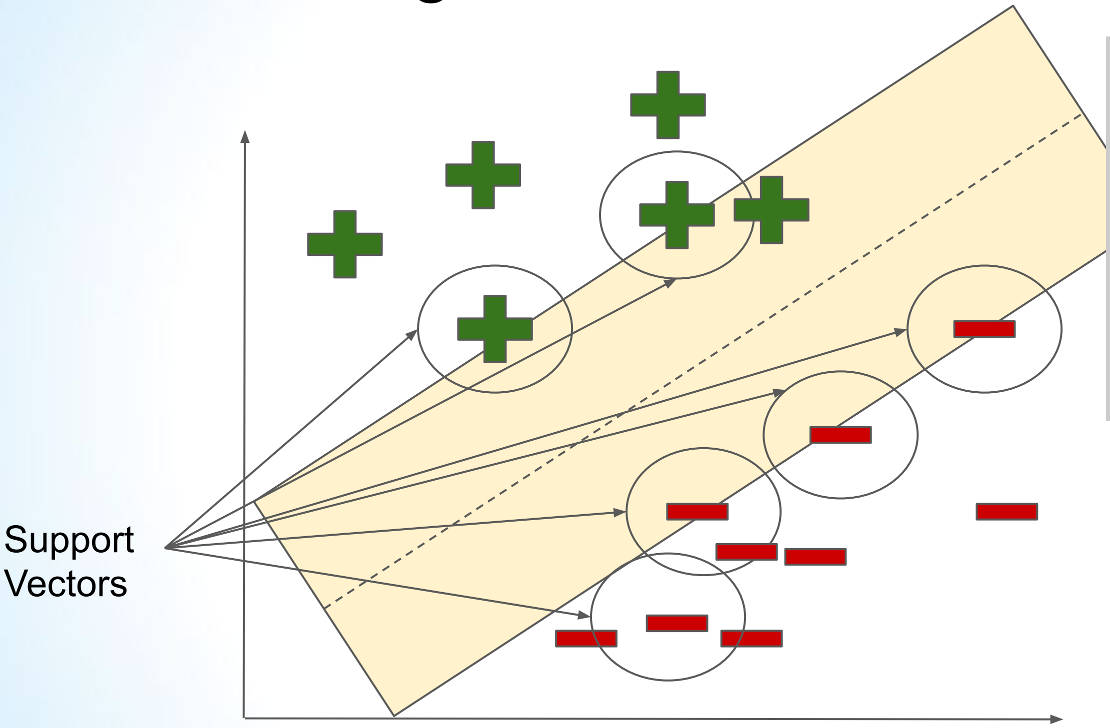
A margin in linear classifiers is the boundary width the touches the datapoint

Maximum margin



A maximum margin in linear classifiers is the Max boundary width the touches the datapoint

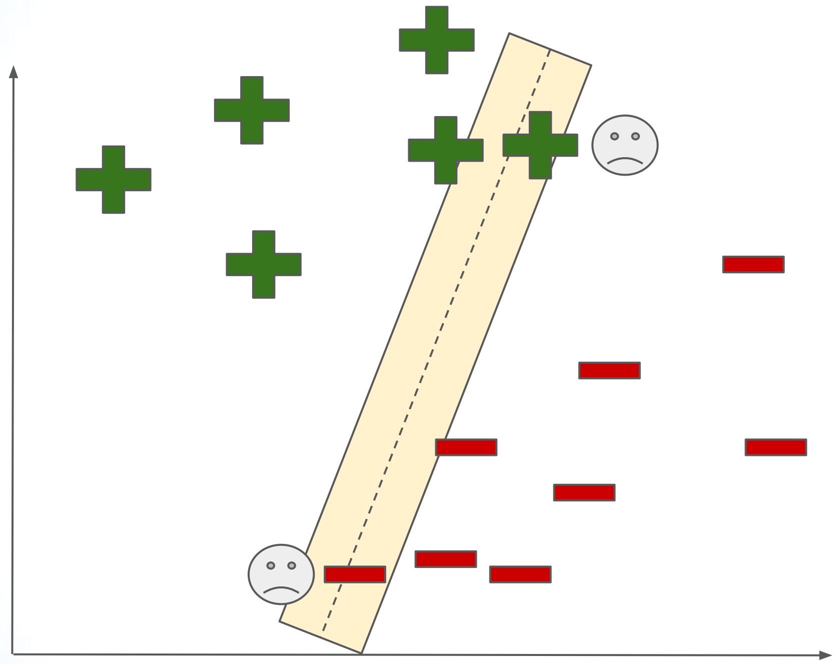
Maximum Margin



A maximum margin in linear classifiers is the Max boundary width the touches the datapoint

The points on the margins are called Support Vectors

Classifier Margin

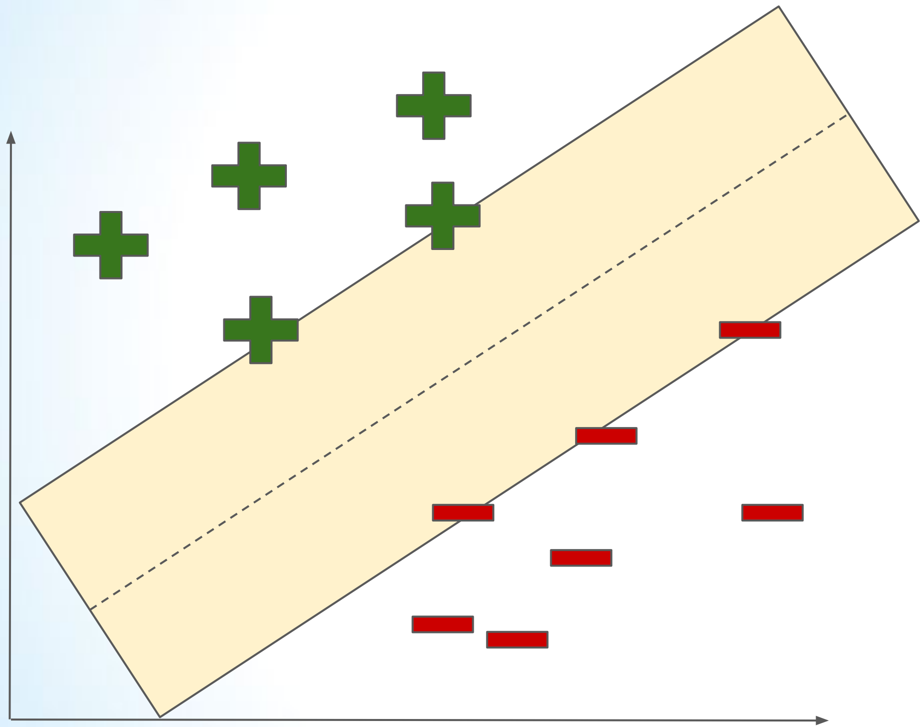


A maximum margin in linear classifiers is the Max boundary width the touches the datapoint

The points on the margins are called Support Vectors

Allows a more flexibility around the decision boundary

Classifier Margin



A maximum margin in linear classifiers is the Max boundary width the touches the datapoint

The points on the margins are called Support Vectors

Allows a more flexibility around the decision boundary

VC dimension **can show that the maximum margin is a good approach to linearly separable problems.**

VC Dimension - Vapnik-Chervonenkis (60-90)

Explain learning from a statistical view

VC-dim measure the **capacity** of a learner

Complexity

Expressive power

Richness

Y-DATA SCHOOL OF DATA SCIENCE

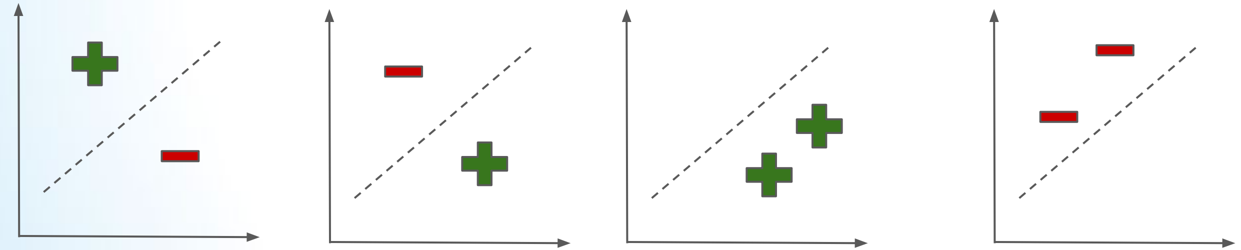
Yandex

VC Dimension - Vapnik-Chervonenkis (60-90)

VC-dim is the maximum number of points the learner can **Shatter**

A learner can **Shatter** points if all y’s can achieve Zero error on the train set

Perceptron / logistic regression for 2 points:



1. Lets try 3 points   And 4..

2. What is the VC-dim of the learner? **dim + 1**   3. What happened with 3 points same line..

VC Dimension - Vapnik-Chervonenkis (60-90)

VC-dim can predict the probabilistic upper bound of the test error (!)

N: The training-set size

$$\Pr\left(\text{test error} \leqslant \text{training error} + \sqrt{\frac{1}{N}\left[D\left(\log\left(\frac{2N}{D}\right)+1\right)-\log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta,$$

Valid When N >> D

$$0 < \eta \leqslant 1$$

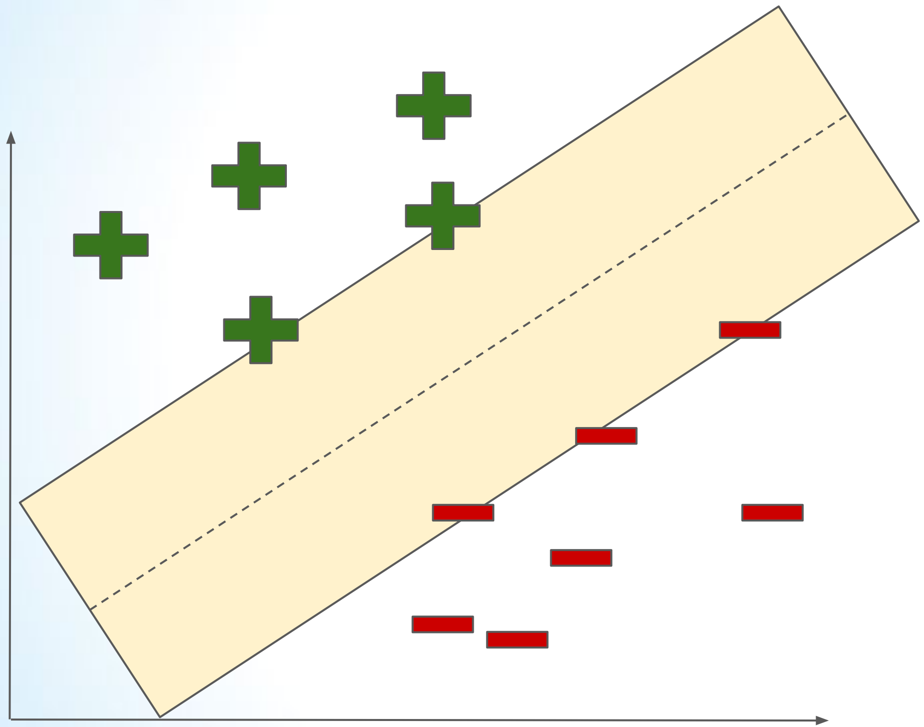
D: The VC-dim of a learner

$$D_{svm} = 1 + \frac{1}{margin^2}$$

In the linear separable case

Higher margin, lower D, lower test error

Classifier Margin



A maximum margin in linear classifiers is the Max boundary width the touches the datapoint

The points on the margins are called Support Vectors

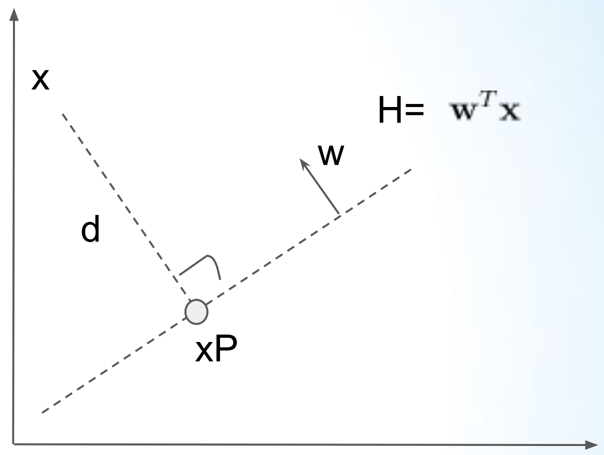
Allows a more flexibility around the decision boundary

VC dimension **can show that the maximum margin is a good approach to linearly separable problems.**

Hard SVM - the separable case

What is the distance between point X and the hyperplane?

- x is a point
- d vector from H to x with minimum length
- xP is the projection of x on H



Y-DATA SCHOOL OF DATA SCIENCE

Yandex

Defining the Margin

$$\mathbf{w}^T\mathbf{x}^P = 0 \qquad \mathbf{x}^P = \mathbf{x} - \mathbf{d} \qquad \mathbf{d} = \alpha\mathbf{w} \quad \alpha \in \mathbb{R}$$

XP is a point on H

d is parallel to w

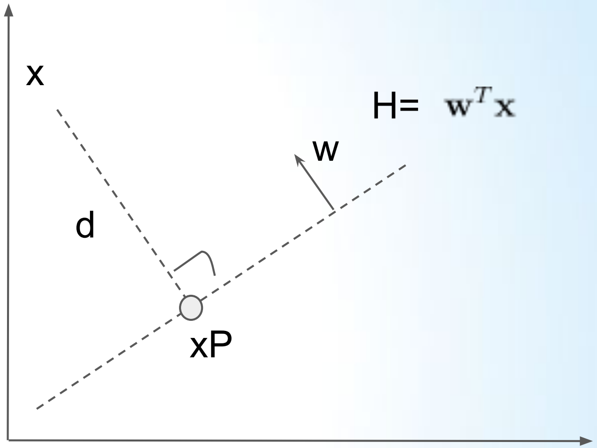
$$\mathbf{w}^T\mathbf{x}^P = \mathbf{w}^T(\mathbf{x} - \mathbf{d}) = \mathbf{w}^T(\mathbf{x} - \alpha\mathbf{w}) = 0$$

$$\alpha = \frac{\mathbf{w}^T\mathbf{x}}{\mathbf{w}^T\mathbf{w}}$$

The length of d: $\|\mathbf{d}\|_2 = \sqrt{\mathbf{d}^T\mathbf{d}} = \sqrt{\alpha^2\mathbf{w}^T\mathbf{w}} = \frac{|\mathbf{w}^T\mathbf{x}|}{\sqrt{\mathbf{w}^T\mathbf{w}}} = \frac{|\mathbf{w}^T\mathbf{x}|}{\|\mathbf{w}\|_2}$

Margin of H with respect to D: $\gamma(\mathbf{w}) = \min_{\mathbf{x}\in D} \frac{|\mathbf{w}^T\mathbf{x}|}{\|\mathbf{w}\|_2}$

By definition, the margin and hyperplane are scale invariant: $\gamma(\beta\mathbf{w}) = \gamma(\mathbf{w}), \forall\beta \neq 0$



Y-DATA SCHOOL OF DATA SCIENCE

Yandex

Defining the Maximum Margin

$$\underbrace{\max_{\mathbf{w}} \gamma(\mathbf{w}, b)}_{maximize\ margin} \text{ such that } \underbrace{\forall i\ y_i(\mathbf{w}^T x_i) \geq 0}_{separating\ hyperplane}$$

$$\underbrace{\max_{\mathbf{w}} \underbrace{\frac{1}{\|\mathbf{w}\|_2} \min_{\mathbf{x}_i \in D} \left|\mathbf{w}^T \mathbf{x}_i\right|}_{\gamma(\mathbf{w})}}_{maximize\ margin} \quad s.t. \quad \underbrace{\forall i\ y_i(\mathbf{w}^T x_i) \geq 0}_{separating\ hyperplane}$$

Because the hyperplane is scale invariant, we can fix the scale of w anyway we want. Let's be clever about it, and choose it such that

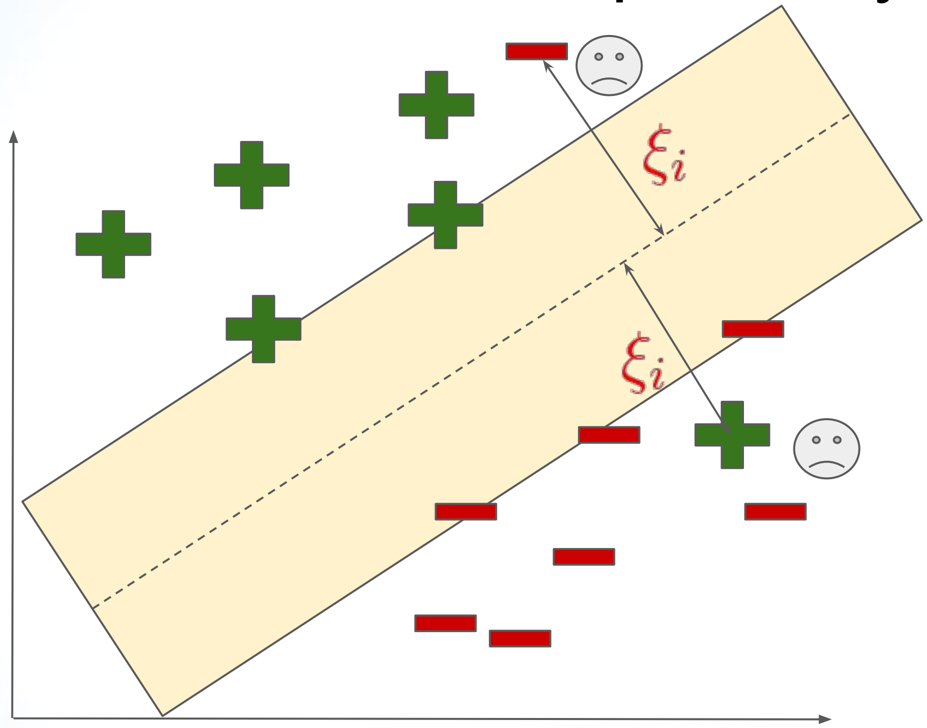
$$\min_{\mathbf{x} \in D} \left|\mathbf{w}^T \mathbf{x}\right| = 1.$$

$$\max_{\mathbf{w}} \frac{1}{\|\mathbf{w}\|_2} \cdot 1 = \min_{\mathbf{w}} \|\mathbf{w}\|_2 = \min_{\mathbf{w}} \mathbf{w}^\top \mathbf{w} \qquad \text{s.t.} \quad \forall i\ y_i(\mathbf{w}^T \mathbf{x}_i) \geq 1$$

$$\boxed{min_w \|w\|^2 \ \ s.t \ \ \forall i, y_i w x_i \geq 1}$$

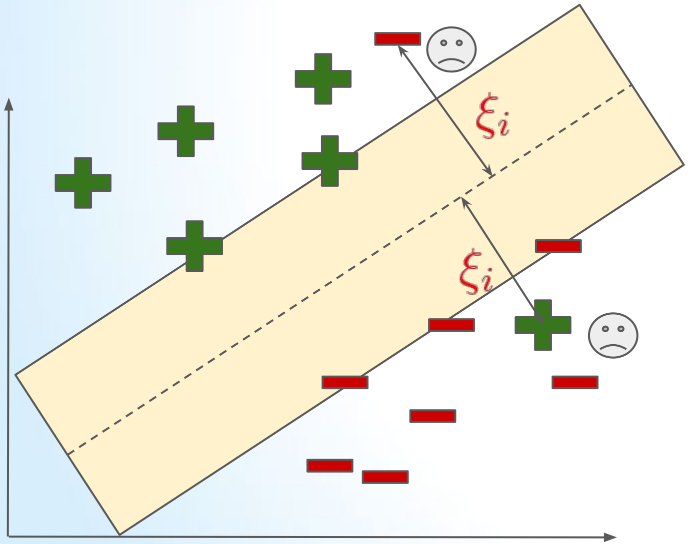
*Assumes full separability

What if there is no linear separability?



Soft SVM - Supporting error data points

$$argmin_{w,\xi}\left(\lambda\left\|w\right\|^2+C\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)$$



$$s.t\ \forall y_i(wx_i) >= 1-\xi_i$$

Soft SVM - Supporting error data points

$$argmin_{w,\xi}\left(\lambda\left\|w\right\|^2 + C\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)$$

$$s.t\ \forall y_i(wx_i) >= 1 - \xi_i$$

$$argmin_{w,\xi}\left(\lambda\left\|w\right\|^2 + \underbrace{hinge(wx,y)}_{max\left\{0,1-ywx\right\}}\right)$$

Soft SVM - Supporting error data points

$$argmin_{w,\xi}\left(\lambda\left\|w\right\|^2 + C\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)$$

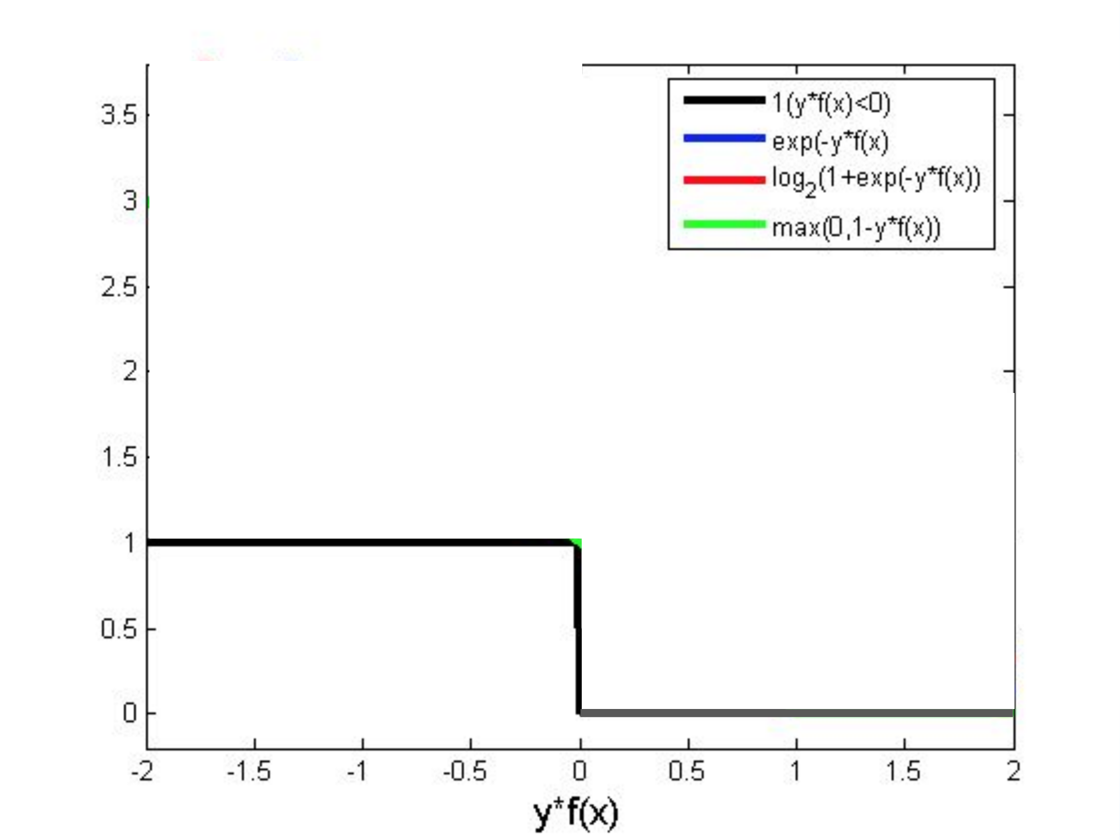
$$s.t\ \forall y_i(wx_i) >= 1-\xi_i$$

$$argmin_{w,\xi}\left(\lambda\left\|w\right\|^2 + \underbrace{hinge(wx,y)}_{max\left\{0,1-ywx\right\}}\right)$$

regularization objective

But came from problem definition

Draw me a function



Soft SVM - Supporting error data points

$$argmin_{w,\xi}\left(\lambda\left\|w\right\|^2 + C\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)$$

$$s.t\ \forall y_i(wx_i) >= 1 - \xi_i$$

$$argmin_{w,\xi}\left(\lambda\left\|w\right\|^2 + \underbrace{hinge(wx,y)}_{max\{0,1-ywx\}}\right)$$

regularization objective

But came from problem definition

How we can solve it?

Gradient descent

Quadratic Programing

```
SGD for solving Soft-SVM
```

goal: Solve $\operatorname{argmin}_{\mathbf{w}} \left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}_i \rangle\} \right)$

parameter: T

initialize: $\boldsymbol{\theta}^{(1)} = \mathbf{0}$

for $t = 1, \ldots, T$

  Let $\mathbf{w}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\theta}^{(t)}$

  Choose i uniformly at random from $[m]$

  If $(y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle < 1)$

    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + y_i \mathbf{x}_i$

  Else

    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$

output: $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$

https://leon.bottou.org/publications/pdf/lin-2006.pdf

Homework

1. **Implementing SVM called Pegasos (2011) - 55 (+ 10 bonus)**
   1. Implement Class - 35
   2. Test - 10
   3. Analyze param - 5
   4. Analyze learning - 5
   5. Mini-batch bonus - 10*
2. **The effect of imbalance on SVM** - **15**
3. **Practical SVM in scikit-learn & hypertune - 10**
4. **Using different Kernels - 20**

Another Way to Solve SVM

And other tricks

From Primal to Dual

Primal

$$min\ \ f_0(x)$$

$$s.t\ \ f_i(x) \le 0 \quad \forall i = 1..m$$

Original SVM definition

From Primal to Dual

Primal

$$min\ \ f_0(x)$$

$$s.t\ \ f_i(x) \leq 0 \quad \forall i = 1..m$$

Original SVM definition

Dual

$$L_{\alpha \geq 0}(x, \alpha) = f_0(x) + \sum_{i=1} \alpha_i f_i(x)$$

$$g(\alpha) = min_x\ \ L(x, \alpha)$$

Dual definition that solvable with **linear solvers**

Y-DATA SCHOOL OF DATA SCIENCE

Yandex

From Primal to Dual

Primal

$$\min f_0(x)$$

$$s.t\ f_i(x) \leq 0 \quad \forall i = 1..m$$

Dual

$$\underset{\alpha \geq 0}{L(x, \alpha)} = f_0(x) + \sum_{i=1} \alpha_i f_i(x)$$

$$g(\alpha) = \min_x\ L(x, \alpha)$$

Original SVM definition

Dual definition that solvable with **linear solvers**

From Primal to Dual

Primal

$$min\ \ f_0(x)$$

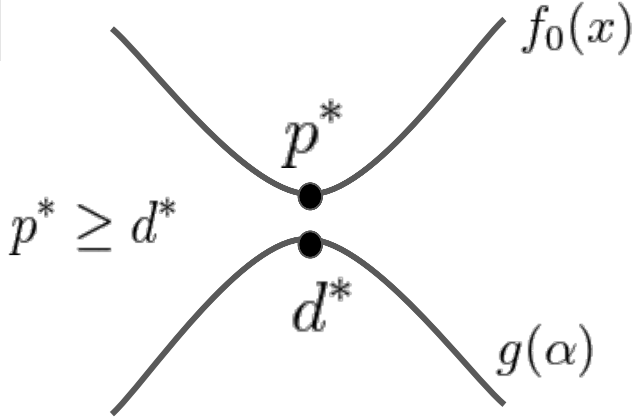
$$s.t\ \ f_i(x) \leq 0 \quad \forall i = 1..m$$

Dual

$$\underset{\alpha \geq 0}{L(x,\alpha)} = f_0(x) + \sum_{i=1} \alpha_i f_i(x)$$

$$g(\alpha) = min_x\ \ L(x,\alpha)$$

Original SVM definition



Dual definition that solvable with **linear solvers**

From Primal to Dual

Primal

$$min \;\; f_0(x)$$

$$s.t \;\; f_i(x) \leq 0 \quad \forall i = 1..m$$

Dual

$$\underset{\alpha \geq 0}{L(x, \alpha)} = f_0(x) + \sum_{i=1}^{m} \alpha f_i(x)$$

$$g(x) = min_x \;\; L(x, \alpha)$$

SVM definition

$$min \;\; \|w\|^2$$

$$s.t \quad y_i w x_i \geq 1 \;\; \forall i = 1..m$$

From Primal to Dual

Primal

$$min\ f_0(x)$$

$$s.t\ f_i(x) \leq 0 \quad \forall i = 1..m$$

SVM definition

$$min\ \|w\|^2$$

$$s.t \quad y_i w x_i \geq 1 \ \ \forall i = 1..m$$

Modify

$$min\ \frac{1}{2}\|w\|^2$$

$$s.t \quad 1 - y_i w x_i \leq 0 \ \ \forall i = 1..m$$

Dual

$$\underset{\alpha \geq 0}{L(x,\alpha)} = f_0(x) + \sum_{i=1}^{m} \alpha f_i(x)$$

$$g(x) = min_x\ L(x,\alpha)$$

Y-DATA SCHOOL OF DATA SCIENCE

Yandex

From Primal to Dual

Primal

$$min\ \ f_0(x)$$

$$s.t\ \ f_i(x) \leq 0 \quad \forall i = 1..m$$

SVM definition

$$min\ \ \|w\|^2$$

$$s.t \quad y_i w x_i \geq 1 \ \ \forall i = 1..m$$

Modify

$$min\ \frac{1}{2}\|w\|^2$$

$$s.t \quad 1 - y_i w x_i \leq 0 \ \ \forall i = 1..m$$

Dual

$$\underset{\alpha \geq 0}{L(x, \alpha)} = f_0(x) + \sum_{i=1}^{m} \alpha f_i(x)$$

$$g(x) = min_x\ \ L(x, \alpha)$$

$$\underset{\alpha \geq 0}{L(x, \alpha)} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i w x_i - 1)$$

$$g(\alpha) = min_x\ \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i w x_i - 1)$$

From Dual to Primal

$$\underset{\alpha \geq 0}{L(x, \alpha)} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i w x_i - 1)$$

$$g(\alpha) = min_x \ \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i w x_i - 1)$$

From Dual to Primal

$$\underset{\alpha\geq 0}{L(x,\alpha)} = \frac{1}{2}\left\|w\right\|^2 - \sum_{i=1}^{m}\alpha_i(y_i w x_i - 1)$$

$$g(\alpha) = min_x \ \frac{1}{2}\left\|w\right\|^2 - \sum_{i=1}^{m}\alpha_i(y_i w x_i - 1)$$

$$\frac{\partial L}{\partial w} = 0$$

From Dual to Primal

$$\underset{\alpha \geq 0}{L(x, \alpha)} = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i w x_i - 1)$$

$$g(\alpha) = min_x \ \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y_i w x_i - 1)$$

$$\frac{\partial L}{\partial w} = 0$$

$$w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0$$

$$w* = \sum_{i=1}^{m} \alpha_i y_i x_i$$

$$\alpha_i \geq 0$$

Lagrange coefficients for each sample in the training set

Optimization Definition	$$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l x_k x_l$$	All possible pairs in the training set
Constraints	$$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$$	

Optimization Definition	$$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l x_k x_l$$
Constraints	$$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$$
Back to Primal	$$w = \sum_{k=1}^{R} \alpha_k y_k x_k$$

Optimization Definition	$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l x_k x_l$
Constraints	$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$
Back to Primal	$w = \sum_{k=1}^{R} \alpha_k y_k x_k$
Predict	$f(x, w) = sign(w, x)$

Optimization Definition

$$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l x_k x_l$$

Constraints

$$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$$

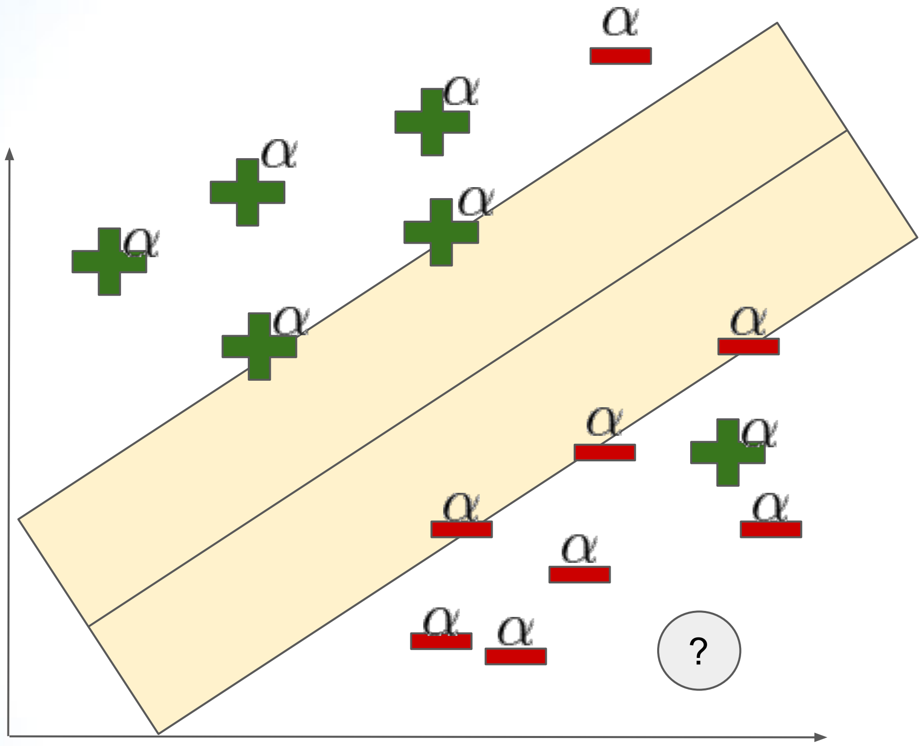
Back to Primal

$$w = \sum_{k=1}^{R} \alpha_k y_k x_k$$

Predict

$$f(x, w) = sign(w, x) \Longrightarrow f(x, w) = sign(\sum_{k=1}^{R} \alpha_k y_k x_k x)$$

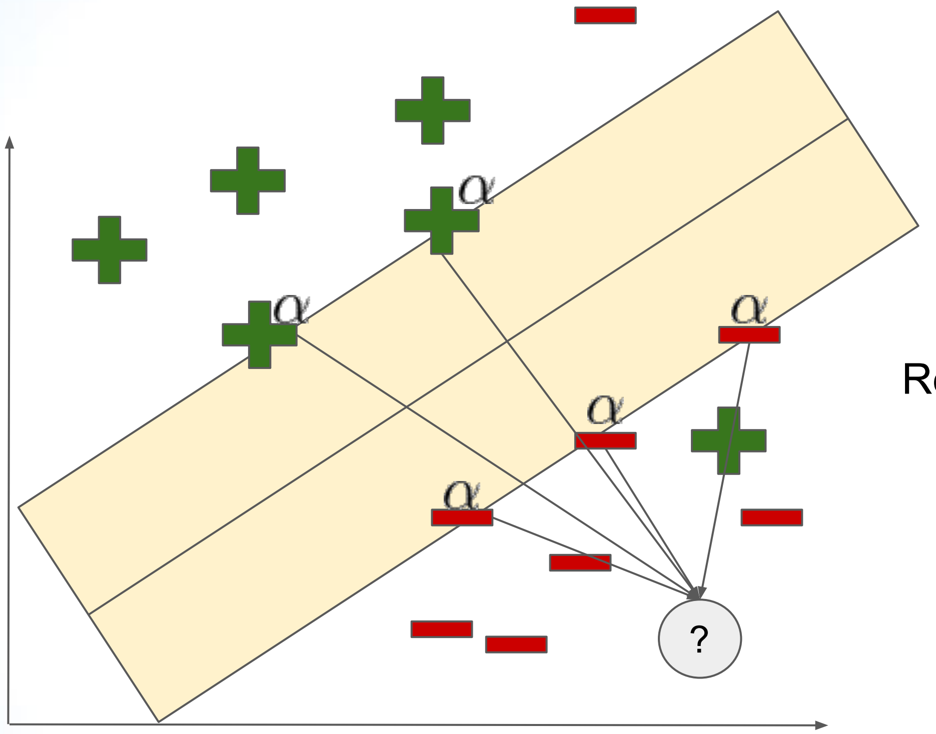
Applying



$$f(x, w) = sign(\sum_{k=1}^{R} \alpha_k y_k x_k x)$$

- What is the alpha values on the margin?
- Outside the margin?

Applying

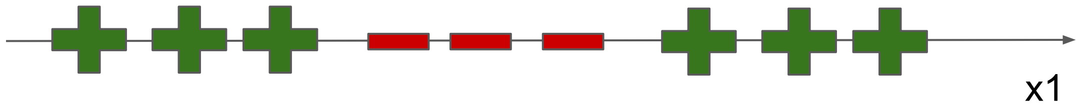


$$f(x, w) = sign(\sum_{k=1}^{R} \alpha_k y_k x_k x)$$

Reminds you something?

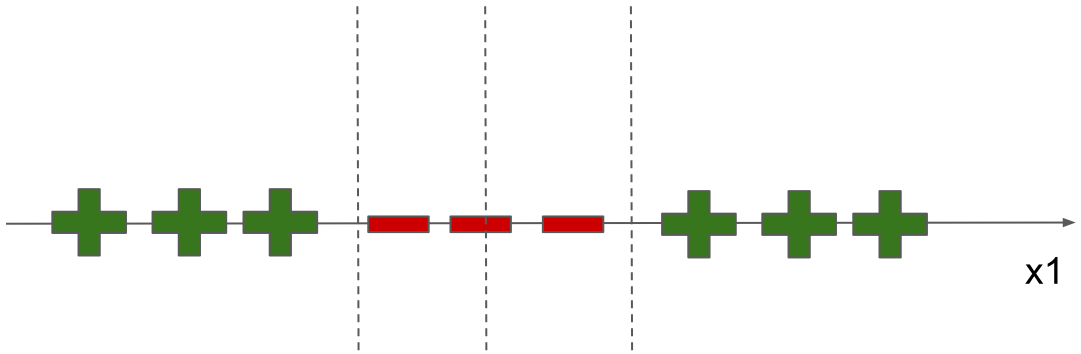
Dealing with non-linearity

How to separate with linear classifier?



Dealing with non-linearity

How to separate with linear classifier?



Y-DATA SCHOOL OF DATA SCIENCE

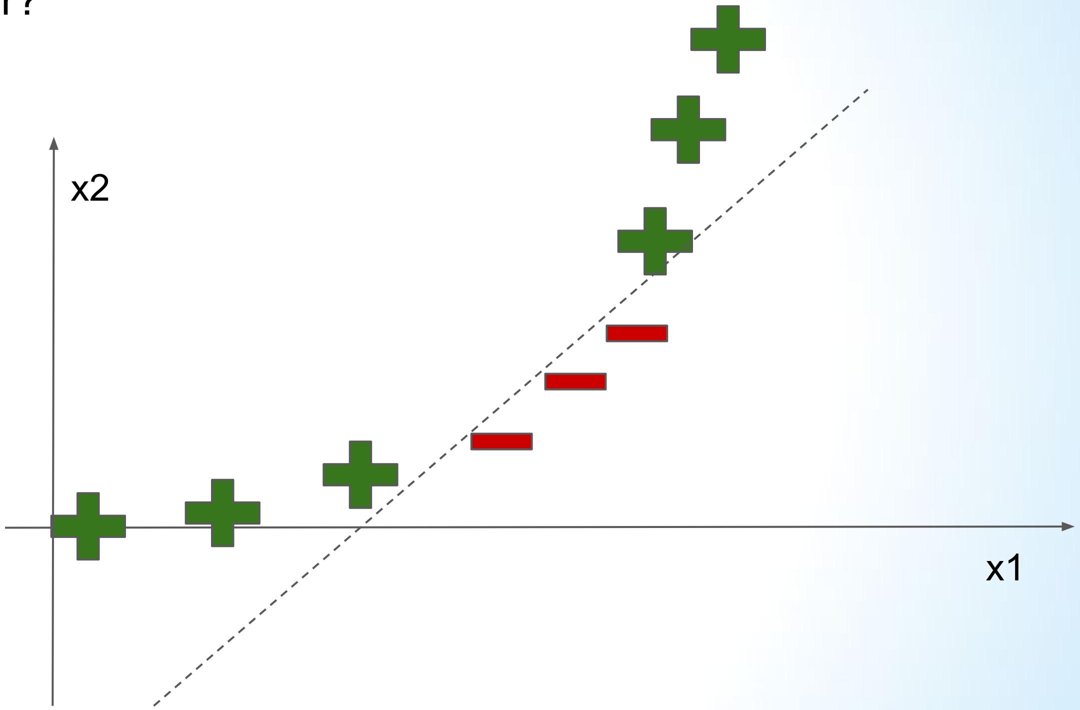
Yandex

Dealing with non-linearity

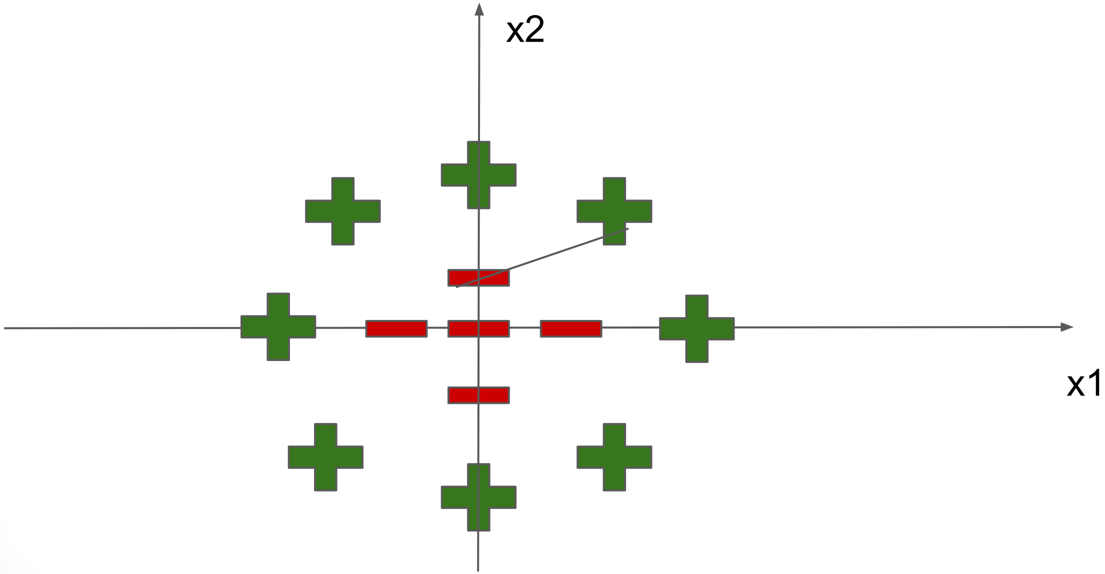
How to separate with linear classifier?

Use non-linear transformation

$$\phi(X) : \mathbb{R}^k \rightarrow \mathbb{R}^n | n > k$$

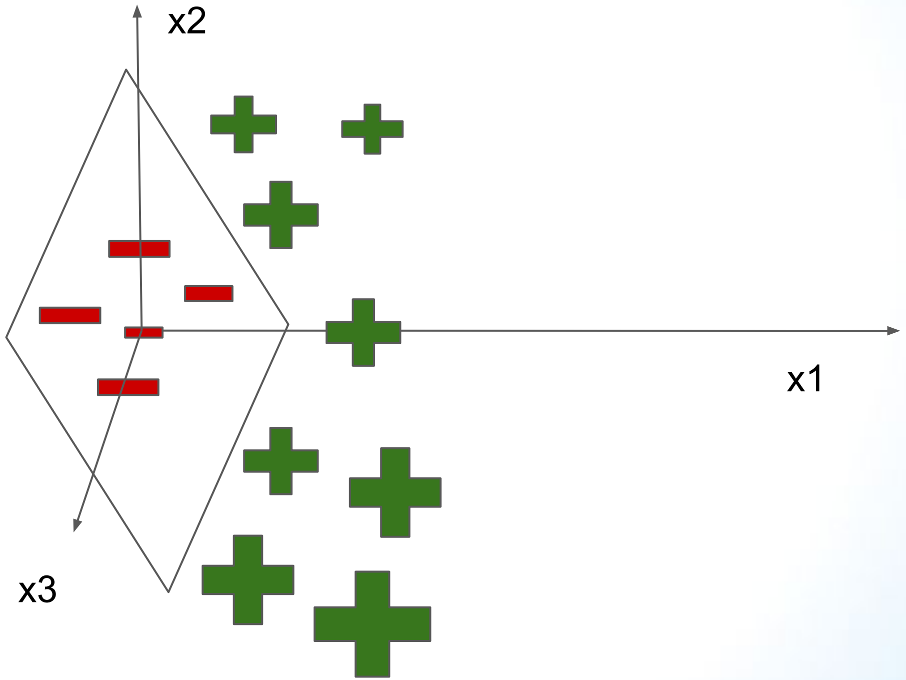


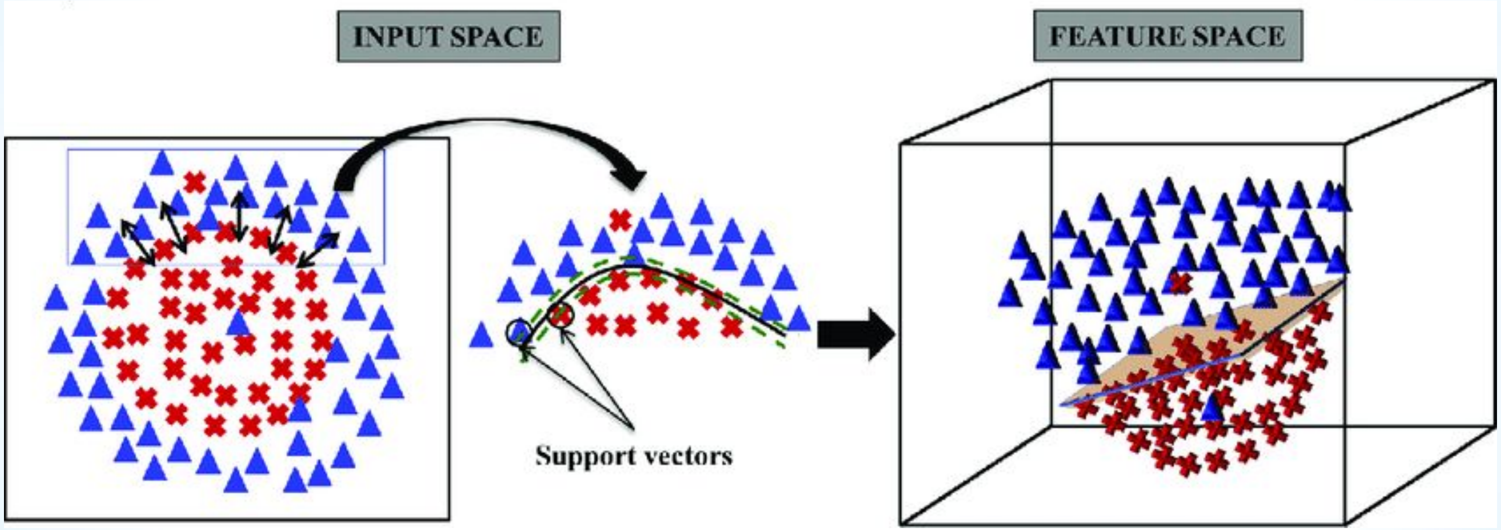
Dealing with non-linearity



Dealing with non-linearity

$$\phi(X) : \mathbb{R}^k \rightarrow \mathbb{R}^n | n > k$$




INPUT SPACE
FEATURE SPACE
Support vectors

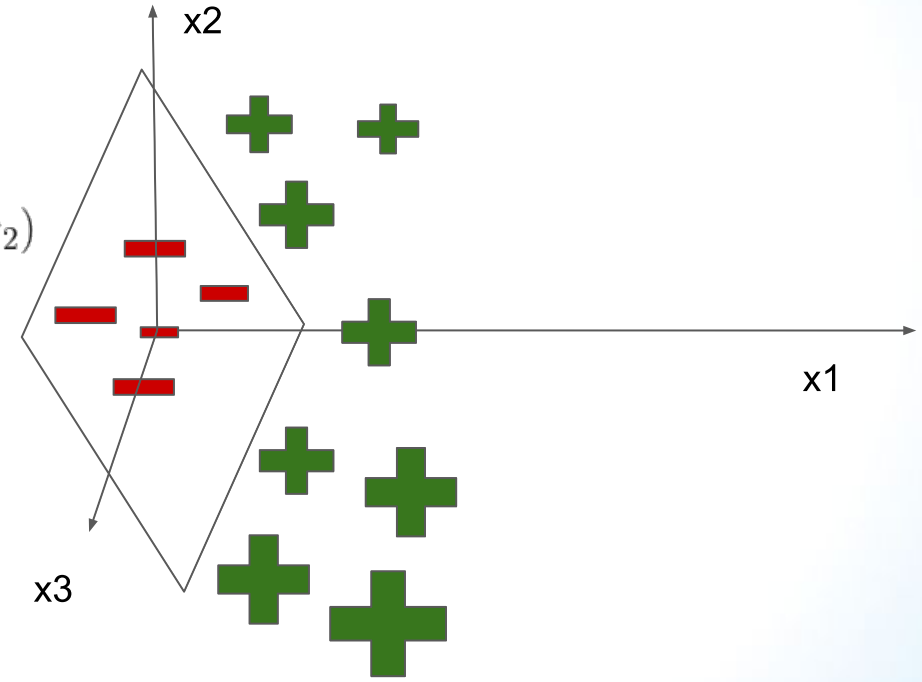
Dealing with non-linearity

How to separate with linear classifier?

$$\phi(X) : \mathbb{R}^k \rightarrow \mathbb{R}^n | n > k$$

$$(x_1, x_2) \mapsto \theta(X) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

A capital X = a datapoint



The Kernel Trick

To support this transformation We would need to compute all these features for each sample

The Kernel Trick

To support this transformation We would need to compute all these features for each sample

The Solution: the Kernel trick!
Use a dot product in feature space can be computed as kernel function

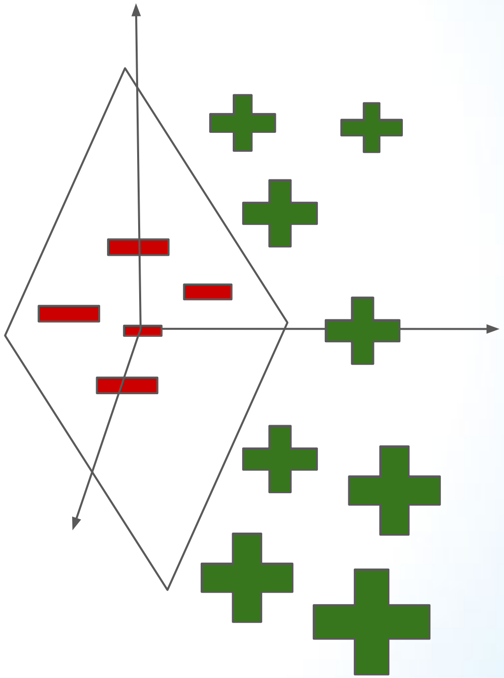
$$K(x_i, x_j) = \phi(x_i)\phi(x_j)$$

The Kernel Trick

$$(x_1, x_2) \mapsto \theta(X) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$K(X_i, X_j) = \phi(X_i)\phi(X_j) = (X_iX_j)^2$$

$$= (X_{i_1}^2, X_{i_2}^2, \sqrt{2}X_{i_1}X_{i_2})(X_{j_1}^2, X_{j_2}^2, \sqrt{2}X_{j_1}X_{j_2})^T =$$
$$(X_{i_1}X_{j_1} + X_{i_2}X_{j_2})^2 = (X_iX_j)^2$$



The Kernel Trick

We would need to compute all these features!

The Solution: the Kernel trick!
Use a dot product in feature space can be computed as kernel function

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

Where do we have dot product in SVM?

Optimization Definition

$$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l x_k x_l$$

Constraints

$$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$$

Back to Primal

$$w = \sum_{k=1}^{R} \alpha_k y_k x_k$$

Predict

$$f(x, w) = sign(w, x) \qquad f(x, w) = sign(\sum_{k=1}^{R} \alpha_k y_k x_k x)$$

Optimization Definition

$$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l \boxed{x_k x_l}$$

Constraints

$$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$$

Back to Primal

$$w = \sum_{k=1}^{R} \alpha_k y_k x_k$$

Predict

$$f(x, w) = sign(w, x)$$

$$f(x, w) = sign(\sum_{k=1}^{R} \alpha_k y_k \boxed{x_k x})$$

Optimization Definition	$$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l \, K(x_k x_l)$$
Constraints	$$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$$
Back to Primal	$$w = \sum_{k=1}^{R} \alpha_k y_k x_k$$
Predict	$$f(x,w) = sign(w,x) \qquad f(x,w) = sign(\sum_{k=1}^{R} \alpha_k y_k x_k x)$$

Optimization Definition

$$max_{\alpha_k} \sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l y_k y_l \, K(x_k x_l)$$

Constraints

$$s.t \quad 0 \leq \alpha_k \leq C \quad \sum_{k=1}^{R} \alpha_k y_k = 0$$

We still need to compute all the kernels pairs, but we don't need to maintain the feature space

Back to Primal

$$w = \sum_{k=1}^{R} \alpha_k y_k x_k$$

Predict

$$f(x, w) = sign(w, x) \qquad f(x, w) = sign(\sum_{k=1}^{R} \alpha_k y_k x_k x)$$

Y-DATA SCHOOL OF DATA SCIENCE

Yandex

Good Kernel Functions for SVM

On top the polynomial kernel function there are more suitable ones:

Radial Basis Function (RBF):

$$K(X_i, X_j) = exp\left(-\frac{(X_i - X_j)^2}{2\sigma^2}\right)$$

Tanh Function (nn):

$$K(X_i, X_j) = tanh\left(\kappa X_i X_j - \delta\right)$$

Good Kernel Functions for SVM

On top the polynomial kernel function there are more suitable ones:

Radial Basis Function (RBF):

$$K(X_i, X_j) = exp\left(-\frac{(X_i - X_j)^2}{2\sigma^2}\right)$$

hypertune

Tanh Function (nn):

$$K(X_i, X_j) = tanh\left(\kappa X_i X_j - \delta\right)$$

SVM Pros & Cons

Good

- Learn many non linear pattern
- Pick "conservative" hypotheses, that are less likely to overfit the data,
- Good for Small datasets with though target pattern
- “Only” 2 params - C, Kernel Function

Y-DATA SCHOOL OF DATA SCIENCE

Yandex

SVM Pros & Cons

Good

- Learn many non linear pattern
- Pick “conservative” hypotheses, that are less likely to overfit the data,
- Good for Small datasets with though target pattern
- “Only” 2 params - C, Kernel Function

Bad

- O(n^2-3) runtime depends on C and the kernel (n number of datapoints)
- O(n^2) memory to compute all the pairwise kernels - 5-10K datapoints
- different values for the Kernel prams

Q&A

VC Dimension - https://www.youtube.com/watch?v=puDzy2XmR5c

https://winvector.github.io/margin/margin.pdf
https://youtu.be/LceLJvKMbBk?t=7311
https://youtu.be/fB47g3QM0sk?t=839

Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for svm. Mathematical programming, 127(1), 3-30. [[pdf](http://www.ee.oulu.fi/research/imag/courses/Vedaldi/ShalevSiSr07.pdf)]

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., & Platt, J. C. (2000). Support vector method for novelty detection. In Advances in neural information processing systems (pp. 582-588). [[pdf](http://papers.nips.cc/paper/1723-support-vector-method-for-novelty-detection.pdf)]

Livni, R., Crammer, K. & Globerson, A.. (2012). A Simple Geometric Interpretation of SVM using Stochastic Adversaries. Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, in PMLR 22:722-730. [[pdf](http://proceedings.mlr.press/v22/livni12/livni12.pdf)]
https://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf
https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote09.html