Day 4: Classification, support vector machines

Introduction to Machine Learning Summer School
June 18, 2018 - June 29, 2018, Chicago

Instructor: Suriya Gunasekar, TTI Chicago

21 June 2018

TOYOTA TECHNOLOGICAL INSTITUTE AT CHICAGO
THE UNIVERSITY OF CHICAGO
NSF

Topics so far

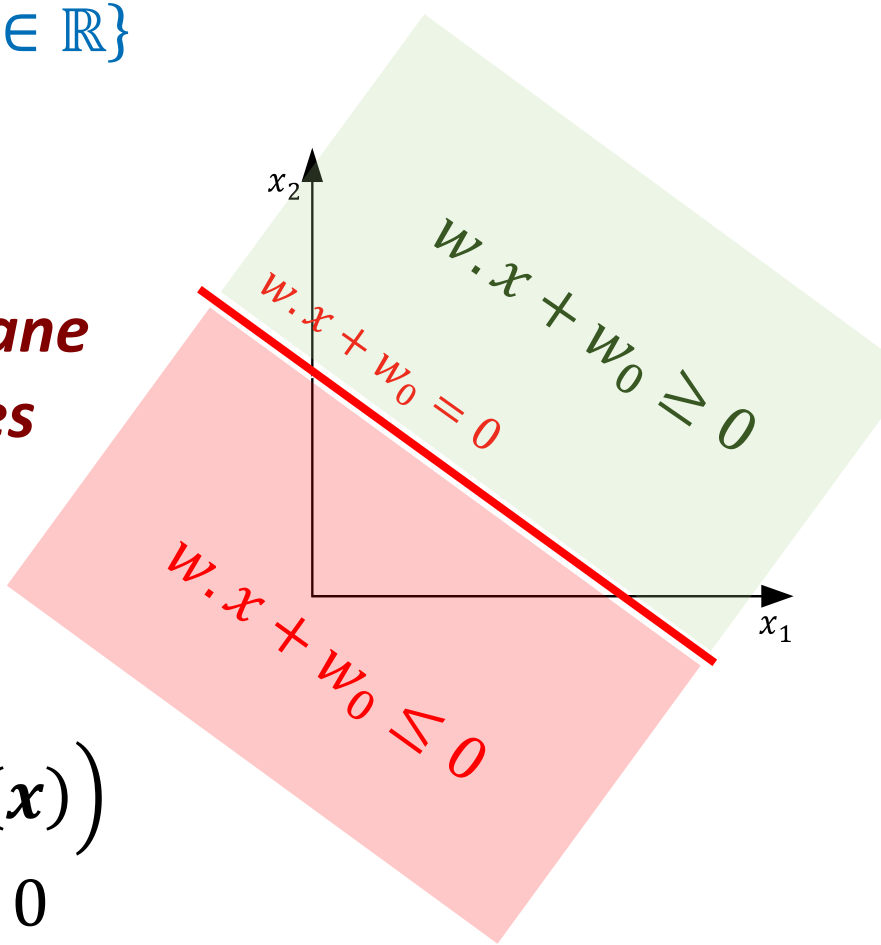
- Supervised learning, linear regression
- Linear regression
  - Overfitting, bias variance trade-off
  - Ridge and lasso regression, gradient descent
- Yesterday
  - Classification, logistic regression
  - Regularization for logistic regression
  - Multi-class classification
- Today
  - Maximum margin classifiers
  - Kernel trick

Classification

- Supervised learning: estimate a mapping f from input $x \in \mathcal{X}$ to output $y \in \mathcal{Y}$
  - **Regression** $\mathcal{Y} = \mathbb{R}$ or other continuous variables
  - **Classification** $\mathcal{Y}$ takes discrete set of values
    - Examples:
      - $\mathcal{Y} = \{\text{spam}, \text{nospam}\}$,
      - digits (not values) $\mathcal{Y} = \{0,1,2, \ldots, 9\}$
- Many successful applications of ML in vision, speech, NLP, healthcare

Parametric classifiers

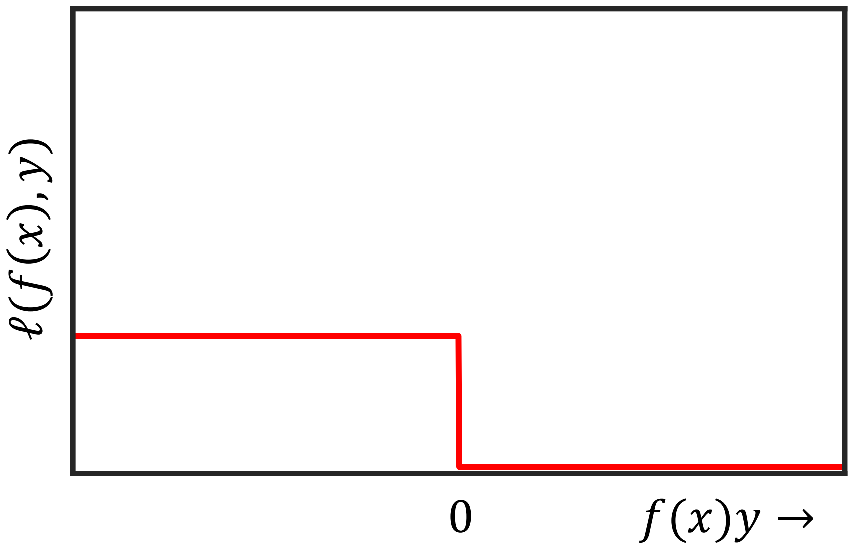
- $\mathcal{H} = \{\boldsymbol{x} \to \boldsymbol{w}.\boldsymbol{x} + w_0 : \boldsymbol{w} \in \mathbb{R}^d, w_0 \in \mathbb{R}\}$
- $\hat{y}(\boldsymbol{x}) = \mathrm{sign}(\widehat{\boldsymbol{w}}.\boldsymbol{x} + \widehat{w}_0)$
- $\widehat{\boldsymbol{w}}.\boldsymbol{x} + \widehat{w}_0 = 0$ (linear) decision boundary or separating ***hyperplane*** separates $\mathbb{R}^d$ into two ***halfspaces*** *(regions)*
  $\widehat{\boldsymbol{w}}.\boldsymbol{x} + \widehat{w}_0 > 0$ gets label 1 and
  $\widehat{\boldsymbol{w}}.\boldsymbol{x} + \widehat{w}_0 < 0$ gets label -1
- more generally, $\hat{y}(\boldsymbol{x}) = \mathrm{sign}\left(\hat{f}(\boldsymbol{x})\right)$
  → decision boundary is $\hat{f}(\boldsymbol{x}) = 0$



Surrogate Losses

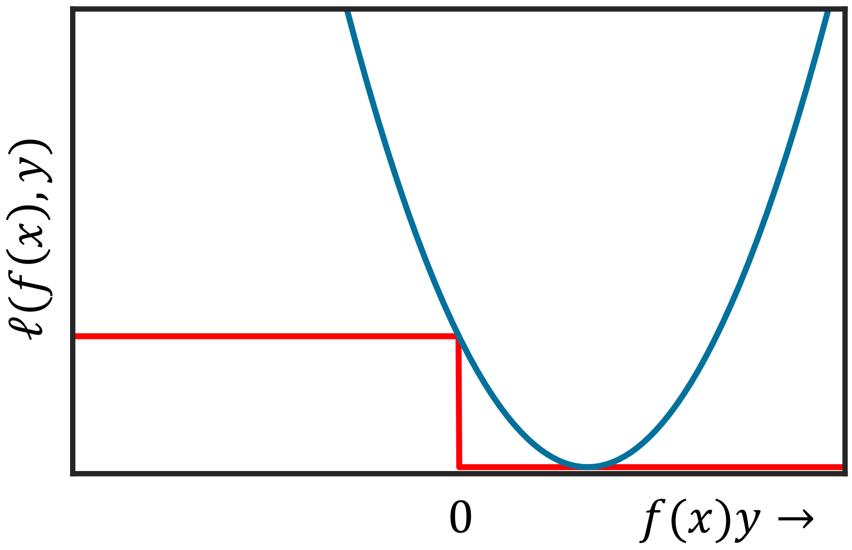
- The correct loss to use is 0-1 loss ***after*** thresholding

$$\begin{aligned}\ell^{01}(f(x), y) &= \mathbf{1}\big[\mathrm{sign}\big(f(x)\big) \neq y\big] \\ &= \mathbf{1}[\mathrm{sign}(f(x)y) < 0]\end{aligned}$$



Surrogate Losses

- The correct loss to use is 0-1 loss ***after*** thresholding
$$\ell^{01}(f(x), y) = \mathbf{1}\big[\mathrm{sign}\big(f(x)\big) \neq y\big]$$
$$= \mathbf{1}[\mathrm{sign}(f(x)y) < 0]$$
- Linear regression uses $\ell^{LS}(f(x), y) = (f(x) - y)^2$



Surrogate Losses

- Hard to optimize over ℓ^{01}, find another loss $\ell(f(x), y)$
  - Convex (for any fixed y) → easier to minimize
  - An upper bound of ℓ^{01} → small $\ell \Rightarrow$ small ℓ^{01}
- Satisfied by squared loss

→but has "large" loss even when $\ell^{01}(f(x), y) = 0$

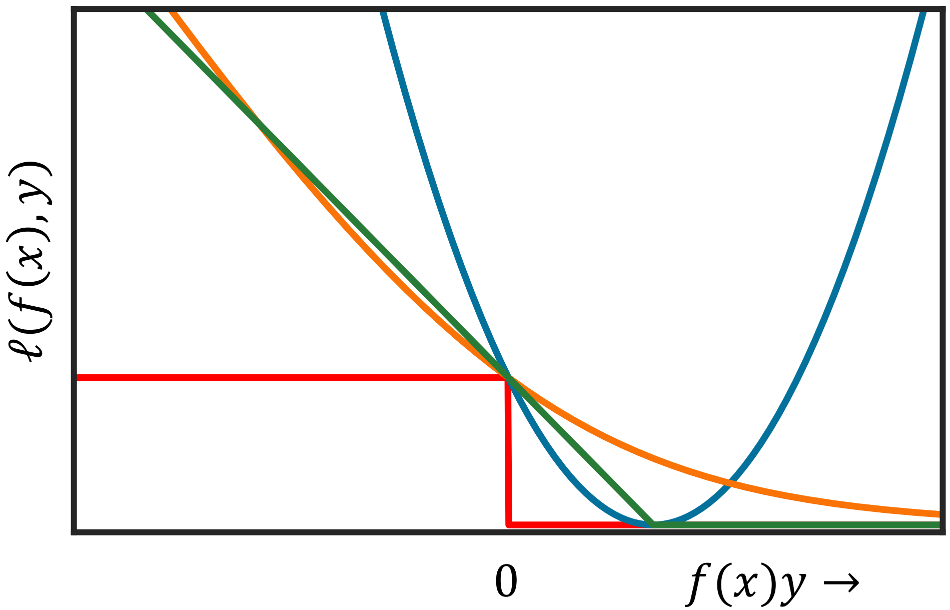
- Two more surrogate losses in in this course

○ Logistic loss

$\ell^{\log}(f(x), y) = \log\big(1 + \exp(-f(x)y$

○ Hinge loss

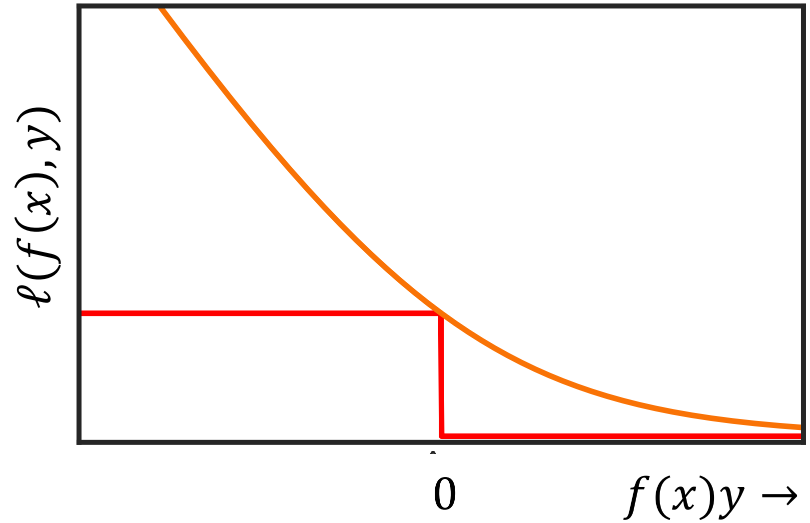
$\ell^{hinge}(f(x), y) = \max(0{,}1 - f(x)y$



Logistic regression: ERM on surrogate loss

Logistic loss

$$\ell(f(x), y) = \log\big(1 + \exp(-f(x)y)\big)$$



- $S = \{(\boldsymbol{x}^{(i)}, y^{(i)}) : i = 1,2,\ldots,N\}, \ \ \mathcal{X} = \mathbb{R}^d, \ \ \mathcal{Y} = \{-1,1\}$
- Linear model $f(\boldsymbol{x}) = f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}.\boldsymbol{x} + w_0$
- Minimize training loss

$$\widehat{\boldsymbol{w}}, \widehat{w}_0 = \underset{\boldsymbol{w}, w_0}{\operatorname{argmin}} \sum_i \log\Big(1 + \exp\big(-(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0)y^{(i)}\big)\Big)$$

- Output classifier $\hat{y}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}.\boldsymbol{x} + w_0)$

Logistic Regression

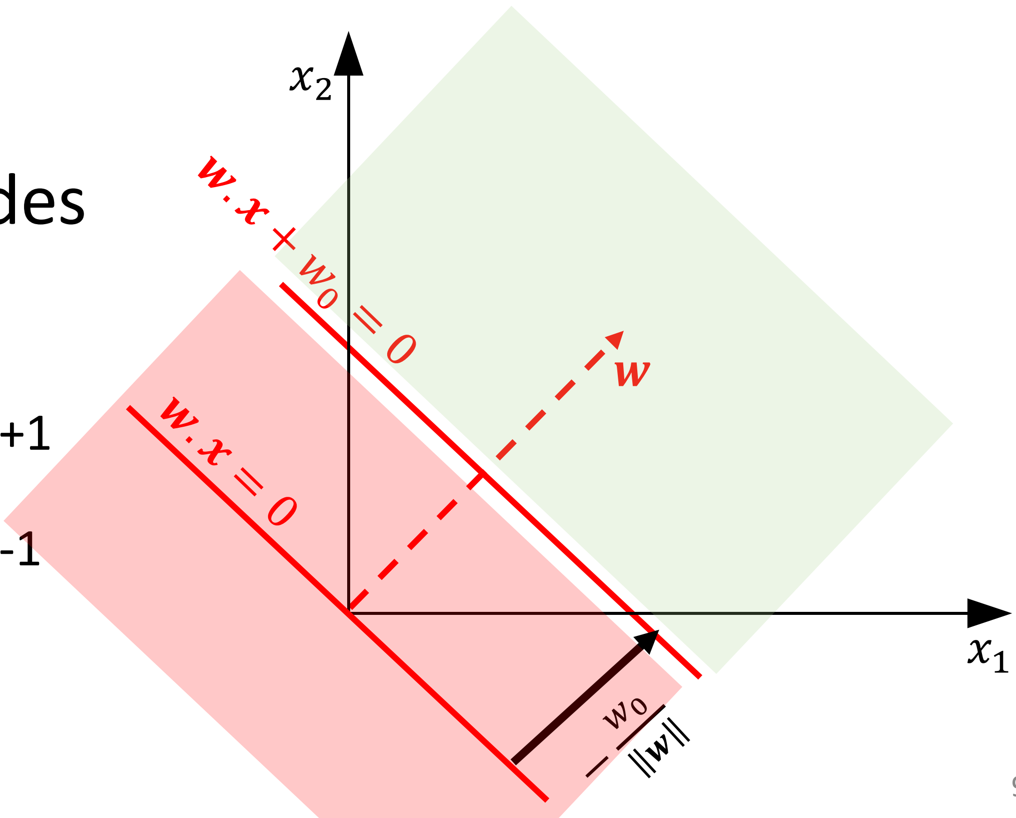
$$\hat{\boldsymbol{w}}, \hat{w}_0 = \underset{\boldsymbol{w}, w_0}{\operatorname{argmin}} \sum_i \log\left(1 + \exp(-(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0)y^{(i)})\right)$$

- Convex optimization problem
- Can solve using gradient descent
- Can also add usual regularization: ℓ_2, ℓ_1

Linear decision boundaries

$$\hat{y}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}.\boldsymbol{x} + w_0)$$

- $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 = 0\}$ is a hyperplane in $\mathbb{R}^d$
  - decision boundary
  - $\boldsymbol{w}$ is direction of normal
  - w_0 is the offset
- $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 = 0\}$ divides $\mathbb{R}^d$ into two halfspaces (regions)
  - $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 \geq 0\}$ get label +1 and $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 < 0\}$ get label -1



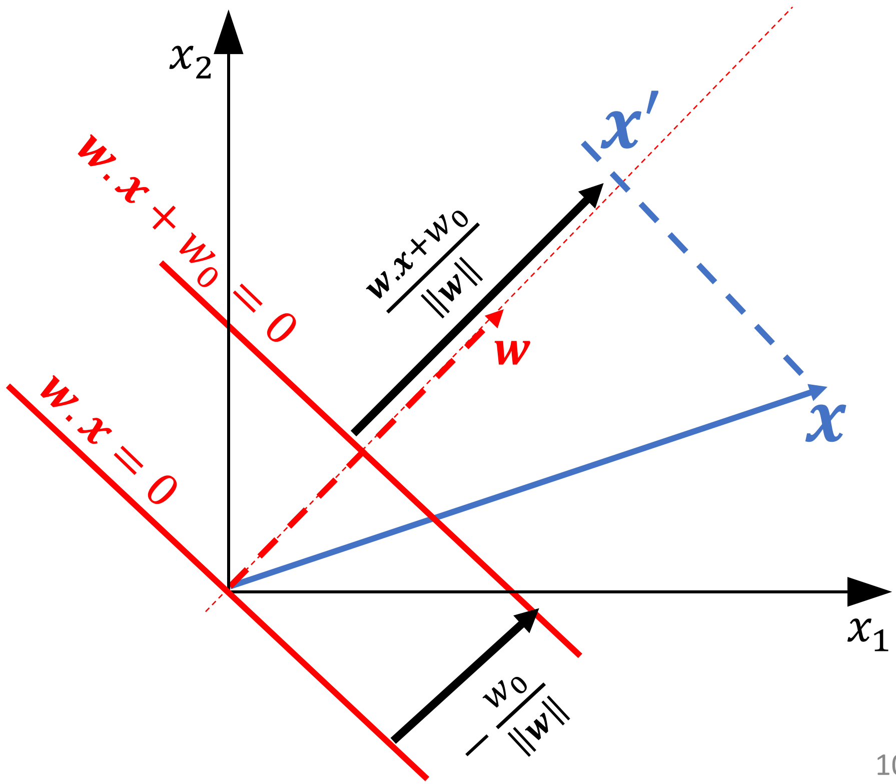
Linear decision boundaries

$$\hat{y}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}.\boldsymbol{x} + w_0)$$

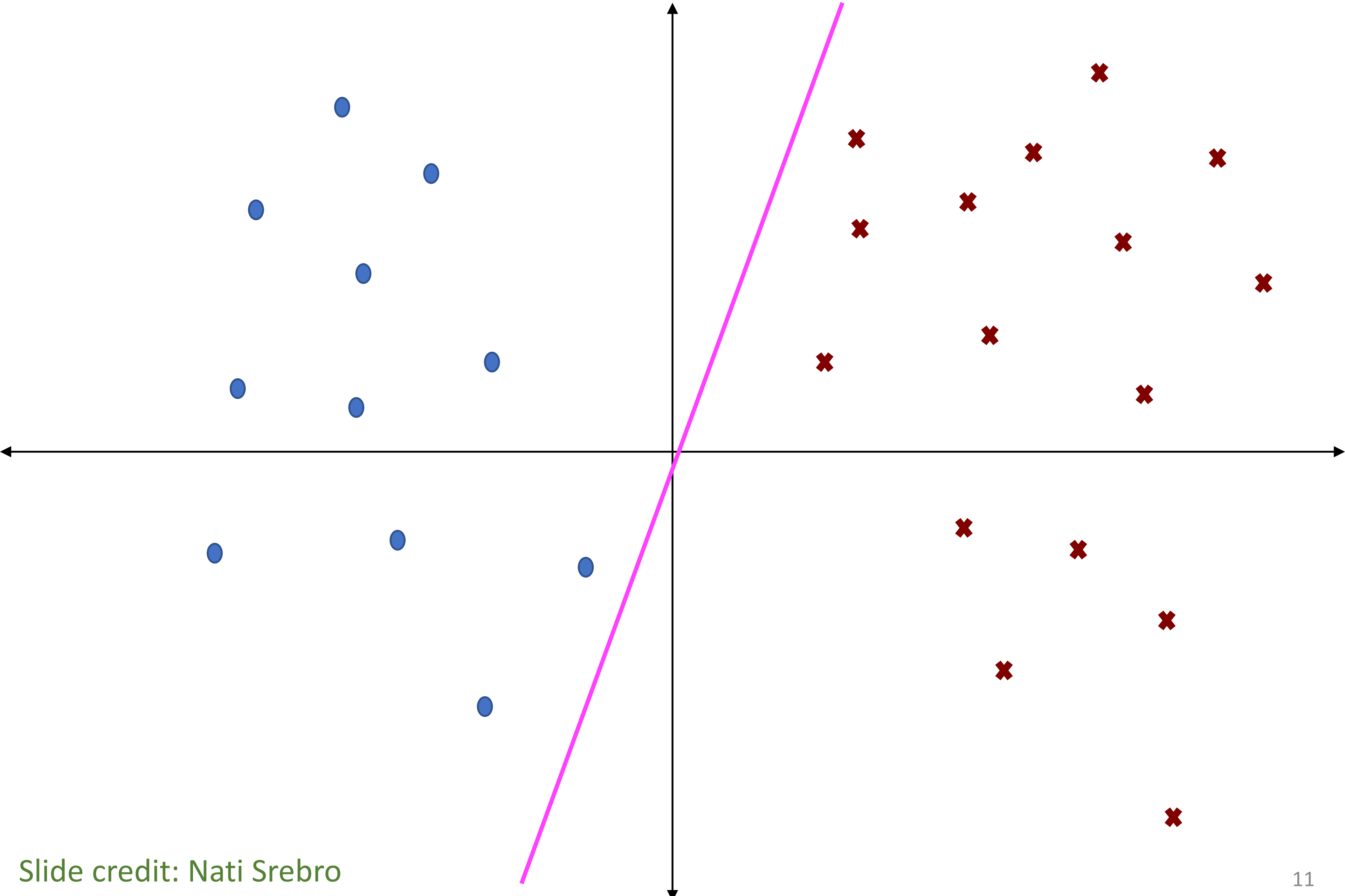
- $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 = 0\}$ is a hyperplane in $\mathbb{R}^d$
  - decision boundary
  - $\boldsymbol{w}$ is direction of normal
  - w_0 is the offset
- $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 = 0\}$ divides $\mathbb{R}^d$ into two halfspaces (regions)
  - $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 \geq 0\}$ get label +1 and $\{\boldsymbol{x}: \boldsymbol{w}.\boldsymbol{x} + w_0 < 0\}$ get label -1

Maps $\boldsymbol{x}$ to a 1D coordinate

$$x' = \frac{\boldsymbol{w}.\boldsymbol{x} + w_0}{\|\boldsymbol{w}\|}$$

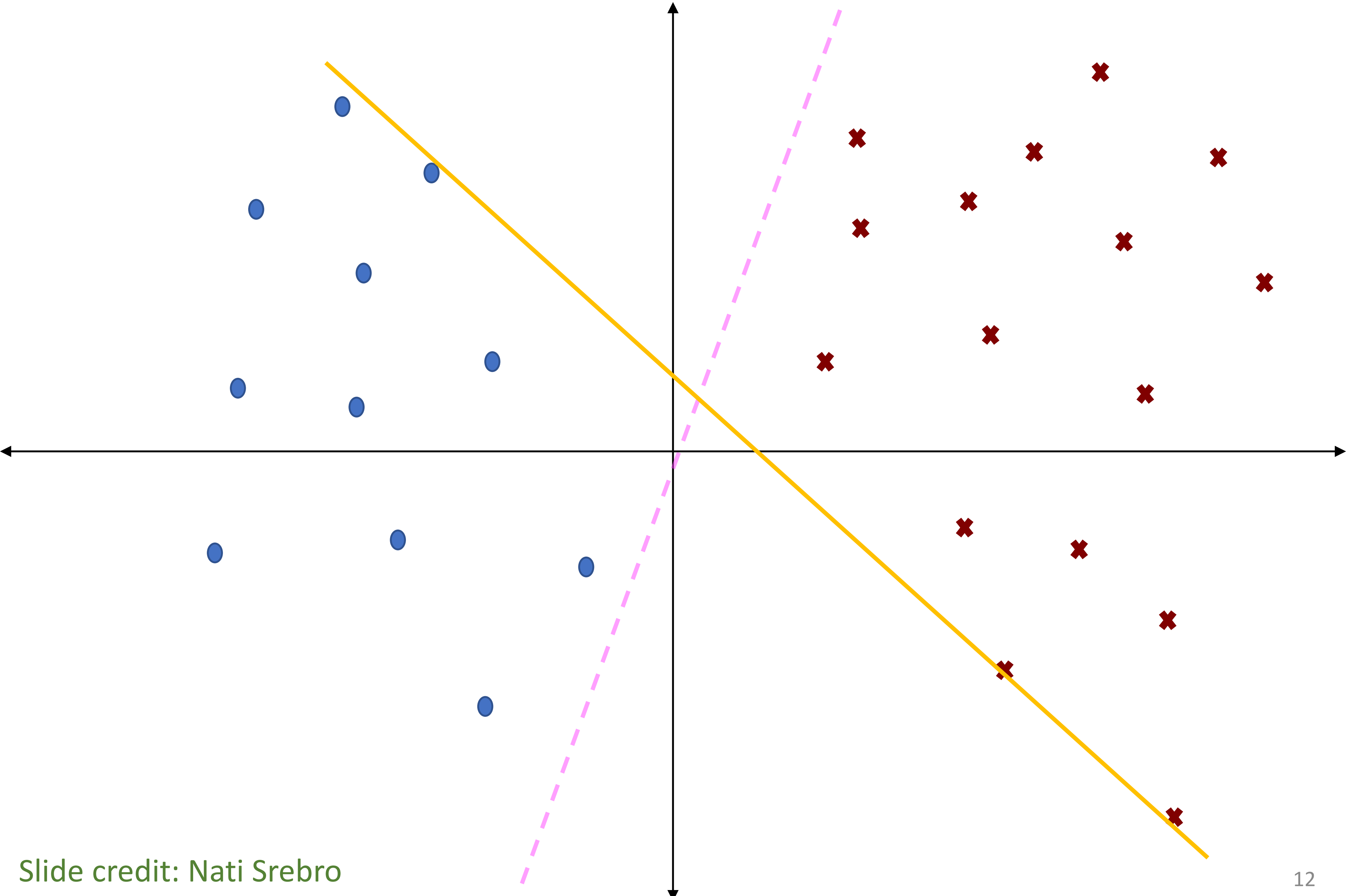


Linear separators in 2D



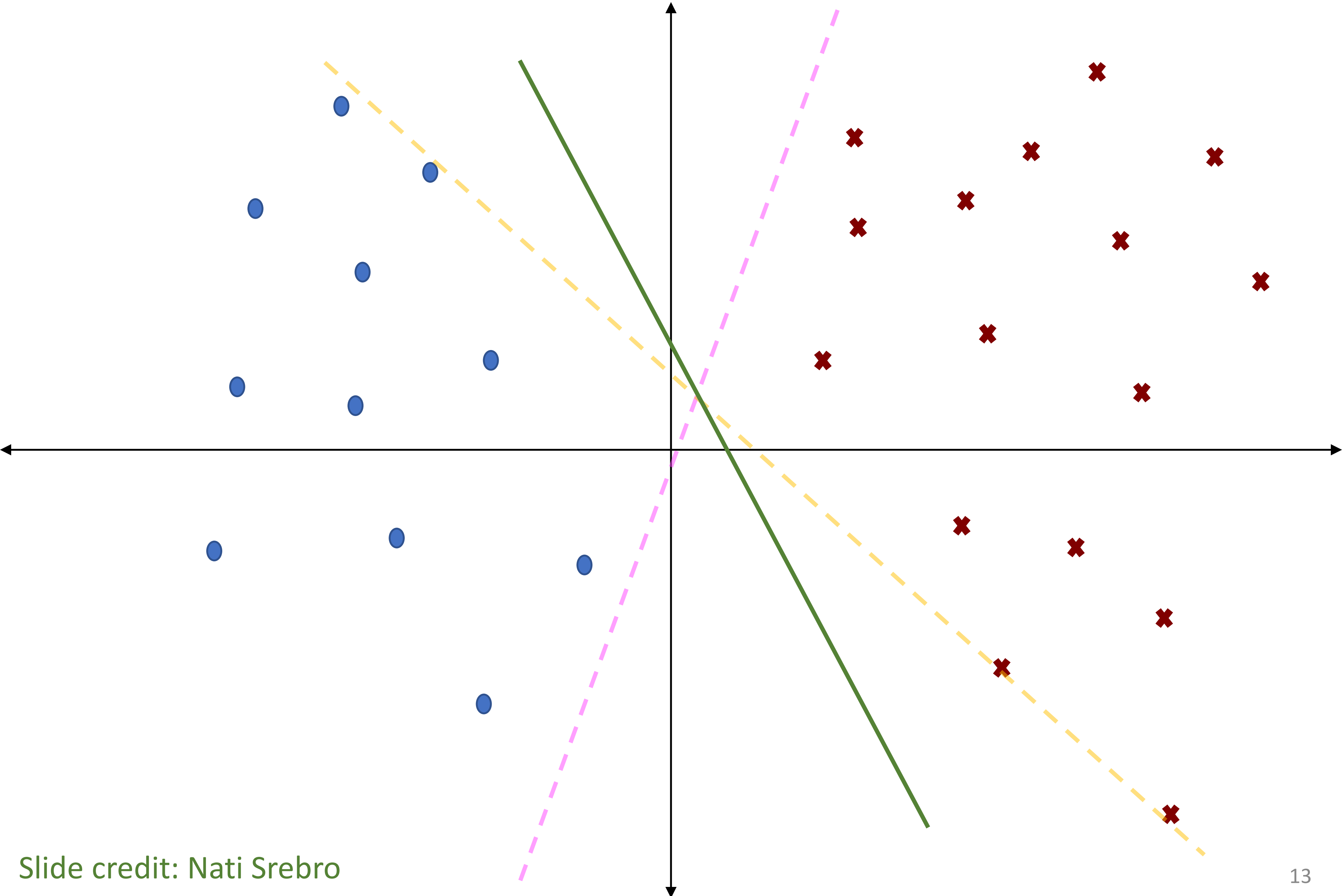
Slide credit: Nati Srebro

Linear separators in 2D



Slide credit: Nati Srebro

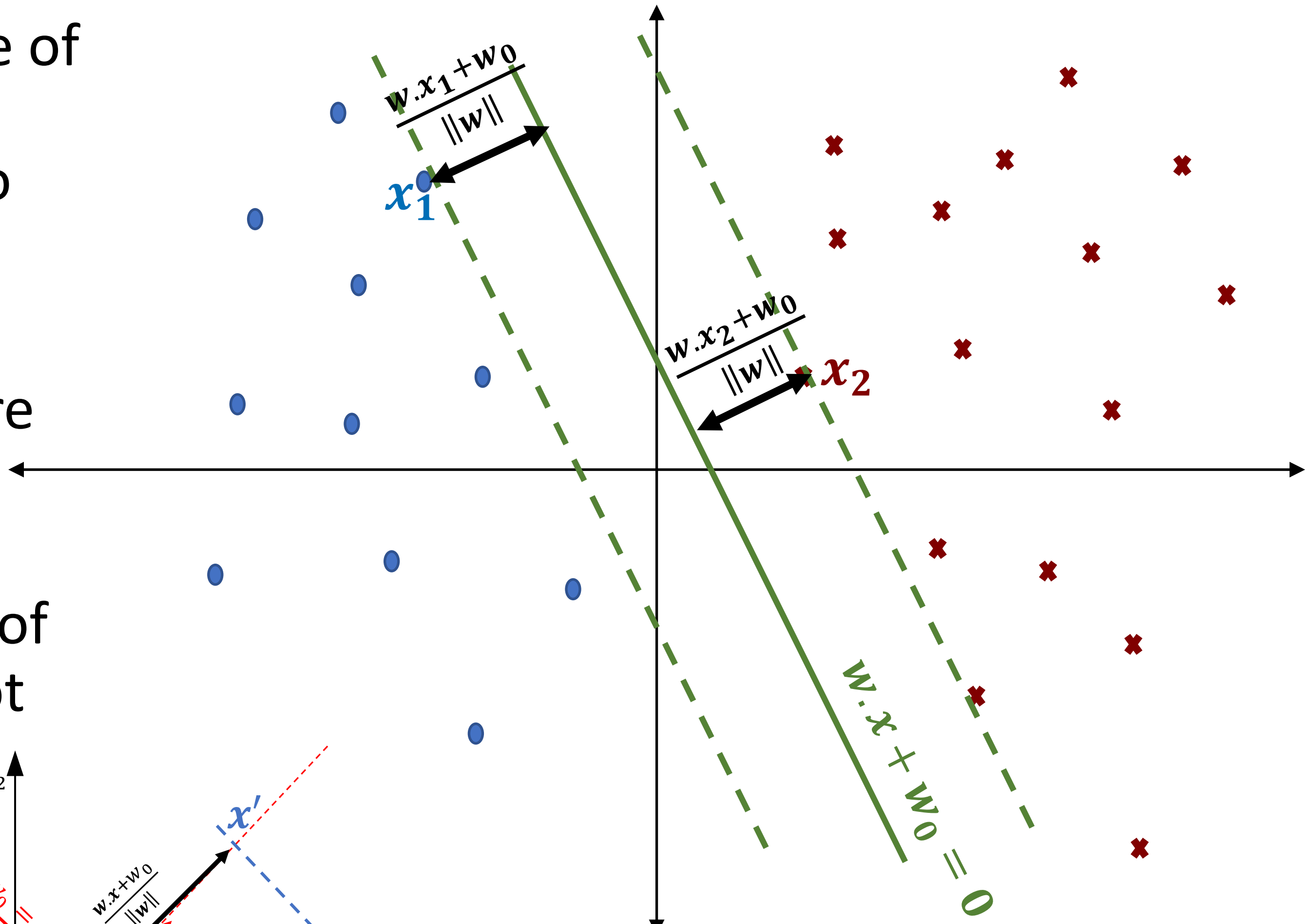
Linear separators in 2D

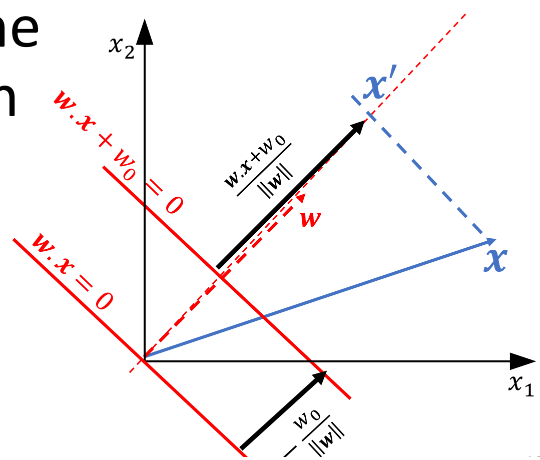


Slide credit: Nati Srebro

Margin of a classifier

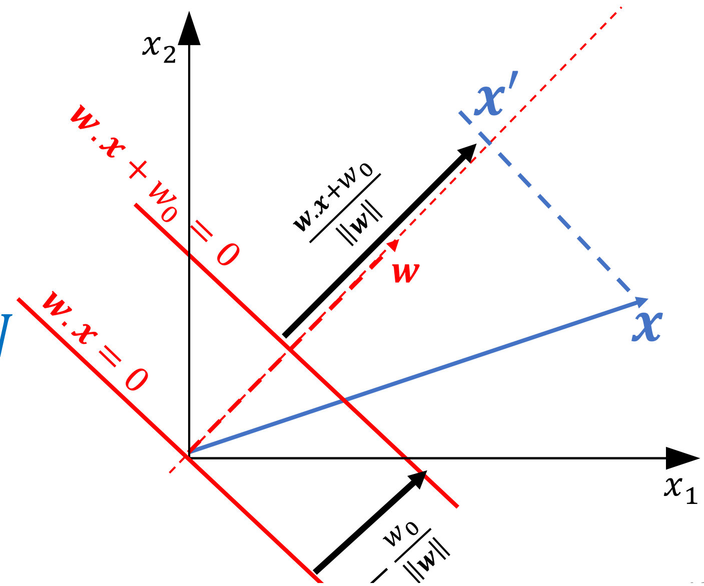
- **Margin:** distance of the closest instance point to the linear hyperplane
- Large margins are more stable
  - small perturbations of the data do not change the prediction





Maximum margin classifier

- $S = \{(\boldsymbol{x}^{(i)}, y^{(i)}) : i = 1,2, \ldots, N\}$
  binary classes $\mathcal{Y} = \{-1,1\}$
- Assume data is “linearly separable”
  - $\exists \boldsymbol{w}, w_0$ such that for all $i = 1,2, \ldots, N$
    $y^{(i)} = \mathrm{sign}(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0)$
    $\Rightarrow y^{(i)}(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0) > 0$



- **Maximum margin** separator given by

$$\hat{\boldsymbol{w}}, \hat{w}_0 = \underset{\boldsymbol{w} \in \mathbb{R}^d, w_0}{\mathrm{argmax}} \underbrace{\min_i}_{\text{smallest margin}} \underbrace{\frac{y^{(i)}(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0)}{\|\boldsymbol{w}\|}}_{\text{margin of sample } i}$$

Maximum margin classifier

$$\hat{\boldsymbol{w}}, \hat{w}_0 = \underset{\boldsymbol{w}\in\mathbb{R}^d, w_0\in\mathbb{R}}{\operatorname{argmax}} \quad \min_i \frac{y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})} + w_0\right)}{\|\boldsymbol{w}\|}$$

- Claim 1: If $\hat{\boldsymbol{w}}, \hat{w}_0$ is a solution,
  then for any $\gamma > 0$, $\gamma\hat{\boldsymbol{w}}, \gamma\hat{w}_0$ is also a solution
- Option 1: We can fix $\|\boldsymbol{w}\| = 1$ to get

$$\hat{\boldsymbol{w}}, \hat{w}_0 = \underset{\|\boldsymbol{w}\|=1, w_0}{\operatorname{arg}max} \quad \min_i y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})} + w_0\right)$$

Maximum margin classifier

$$\widehat{\boldsymbol{w}}, \widehat{w}_0 = \underset{\boldsymbol{w}\in\mathbb{R}^d, w_0\in\mathbb{R}}{\operatorname{argmax}} \quad \min_i \frac{y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0\right)}{\|\boldsymbol{w}\|}$$

- Claim 1: If $\widehat{\boldsymbol{w}}, \widehat{w}_0$ is a solution,
  then for any $\gamma > 0$, $\gamma\widehat{\boldsymbol{w}}, \gamma\widehat{w}_0$ is also a solution
- Option 1: we can fix $\|\boldsymbol{w}\| = 1$ to get

$$\widehat{\boldsymbol{w}}, \widehat{w}_0 = \underset{\|\boldsymbol{w}\|=1}{\operatorname{arg}max} \quad \min_i y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0\right)$$

- Option 2: we can also fix $\min_i y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0\right) = 1$
  - margin now is $\frac{1}{\|\boldsymbol{w}\|}$
  - Instead of “increasing margin” we can “reduce norm”

Max-margin classifier equivalent formulation

- Solve: $\widetilde{\boldsymbol{w}}, \widetilde{w}_0 = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2$

  s.t. $\forall i, \quad y^{(i)}(\boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})} + w_0) \geq 1$

Hard margin Support Vector Machine (SVM)

- **Claim 2:** Equivalent to previous slide

  → $\frac{\widetilde{\boldsymbol{w}}}{\|\widetilde{\boldsymbol{w}}\|}, \frac{\widetilde{w}_0}{\|\widetilde{\boldsymbol{w}}\|}$ is solution for

$$\widehat{\boldsymbol{w}}, \widehat{w}_0 = \max_{\|\boldsymbol{w}\|=1} \min_i \; y^{(i)}(\boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})} + w_0)$$

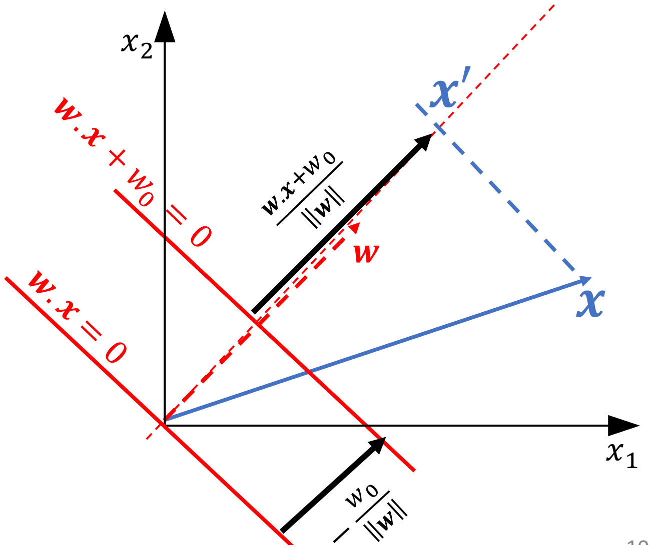
- Proof:

1. Let $\min_i \; y^{(i)}\left(\widehat{\boldsymbol{w}}.\boldsymbol{x}^{(\boldsymbol{i})} + \widehat{w}_0\right) = \hat{\gamma}$, then $\min_i y^{(i)}\left(\frac{\widehat{\boldsymbol{w}}}{\hat{\gamma}}.\boldsymbol{x}^{(\boldsymbol{i})} + \frac{\widehat{w}_0}{\hat{\gamma}}\right) \geq 1$
2. $\Rightarrow \|\widetilde{\boldsymbol{w}}\| \leq \left\|\frac{\widehat{\boldsymbol{w}}}{\hat{\gamma}}\right\| = \frac{1}{\hat{\gamma}}$
3. $\min_i \; y^{(i)}\left(\frac{\widetilde{\boldsymbol{w}}}{\|\widetilde{\boldsymbol{w}}\|}.x^{(i)} + \frac{\widetilde{w}_0}{\|\widetilde{\boldsymbol{w}}\|}\right) = \min_i \frac{y^{(i)}\left(\widetilde{\boldsymbol{w}}.\boldsymbol{x}^{(\boldsymbol{i})} + \widetilde{w}_0\right)}{\|\widetilde{\boldsymbol{w}}\|} \geq \frac{1}{\|\widetilde{\boldsymbol{w}}\|} \geq \hat{\gamma}$

Maximum margin classifier formulations

- Original formulation

$$\hat{\boldsymbol{w}}, \hat{w}_0 = \underset{\boldsymbol{w}\in\mathbb{R}^d, w_0\in\mathbb{R}}{\operatorname{argmax}} \quad \min_i \frac{y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0\right)}{\|\boldsymbol{w}\|}$$



- Fixing $\|\boldsymbol{w}\| = 1$

$$\hat{\boldsymbol{w}}, \hat{w}_0 = \underset{\boldsymbol{w}, w_0}{\operatorname{argmax}} \quad \min_i y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0\right) \ \text{ s.t. } \|\boldsymbol{w}\| = 1$$

- Fixing $\min_i \ y^{(i)}\left(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0\right) = 1$

$$\widetilde{\boldsymbol{w}}, \widetilde{w}_0 = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \ \text{ s.t. } \forall i, y^{(i)}(\boldsymbol{w}.\boldsymbol{x}^{(i)} + w_0) \geq 1$$

Henceforth, w_0 will be absorbed into $\boldsymbol{w}$ by adding an additonal feature of ‘1’ to $\boldsymbol{x}$

Margin and norm

- $\text{margin}(\boldsymbol{w}) = \min_i \frac{y^{(i)}\ \boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})}}{\|\boldsymbol{w}\|}$
- Remember in regression: small norm solutions have low complexity!
  - Is this true for maximum margin classifiers?
  - what about classification with logistic loss $\sum_i \log(1 + \exp(-y^{(i)}\ \boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})}))$?
  - how to do capacity control in maximum margin classifier learning?
- Some places $\min_i y^{(i)}\ \boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})}$ referred as margin → implicitly assumes normalization
  - $\min_i y^{(i)}\ \boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})}$ is meaningless without knowing what $\|\boldsymbol{w}\|$ is!

Solutions of hard margin SVM

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \qquad s.t., \qquad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \quad \forall i$$

- Theorem: $\hat{\boldsymbol{w}} = span\{\boldsymbol{x}^{(\boldsymbol{i})}: i = 1,2,\dots,N\}$

i.e., $\exists\{\hat{\beta}_i: i = 1,2,\dots,N\}$ such that $\hat{\boldsymbol{w}} = \sum_i \hat{\beta}_i\ \boldsymbol{x}^{(\boldsymbol{i})}$

Solutions of hard margin SVM

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \qquad s.t., \qquad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \quad \forall i$$

- Theorem: $\hat{\boldsymbol{w}} = span\{\boldsymbol{x}^{(\boldsymbol{i})}: i = 1,2,\dots,N\}$

i.e., $\exists\{\hat{\beta}_i: i = 1,2,\dots,N\}$ such that $\hat{\boldsymbol{w}} = \sum_i \hat{\beta}_i\ \boldsymbol{x}^{(\boldsymbol{i})}$

- Denote $\mathcal{S} = span\{\boldsymbol{x}^{(\boldsymbol{i})}: i = 1,2,\dots,N\}$ and
$$\mathcal{S}^{\perp} = \{\boldsymbol{z} \in \mathbb{R}^d: \forall i, \boldsymbol{z}.\boldsymbol{x}_{\boldsymbol{i}} = 0\}$$
    - For any $\boldsymbol{z} \in \mathbb{R}^d, \boldsymbol{z} = \boldsymbol{z}_{\mathcal{S}} + \boldsymbol{z}_{\mathcal{S}^{\perp}}$ s.t. $\boldsymbol{z}_{\mathcal{S}} \in \mathcal{S}$ and $\boldsymbol{z}_{\mathcal{S}^{\perp}} \in \mathcal{S}^{\perp}$
    - $\|\boldsymbol{z}\|^2 = \|\boldsymbol{z}_{\mathcal{S}}\|^2 + \left\|\boldsymbol{z}_{\mathcal{S}^{\perp}}\right\|^2$

Solutions of hard margin SVM

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \quad s.t., \quad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \quad \forall i$$

- Theorem: $\hat{\boldsymbol{w}} = span\{\boldsymbol{x}^{(\boldsymbol{i})}: i = 1,2, \dots, N\}$

i.e., $\exists\{\hat{\beta}_i: i = 1,2, \dots, N\}$ such that $\hat{\boldsymbol{w}} = \sum_i \hat{\beta}_i\ \boldsymbol{x}^{(\boldsymbol{i})}$

- Denote $\mathcal{S} = span\{\boldsymbol{x}^{(\boldsymbol{i})}: i = 1,2, \dots, N\}$ and

$$\mathcal{S}^{\perp} = \{\boldsymbol{z} \in \mathbb{R}^d: \forall i, \boldsymbol{z}.\boldsymbol{x}_{\boldsymbol{i}} = 0\}$$

  - For any $\boldsymbol{z} \in \mathbb{R}^d, \boldsymbol{z} = \boldsymbol{z}_{\mathcal{S}} + \boldsymbol{z}_{\mathcal{S}^{\perp}}$ s.t. $\boldsymbol{z}_{\mathcal{S}} \in \mathcal{S}$ and $\boldsymbol{z}_{\mathcal{S}^{\perp}} \in \mathcal{S}^{\perp}$
  - $\|\boldsymbol{z}\|^2 = \|\boldsymbol{z}_{\mathcal{S}}\|^2 + \left\|\boldsymbol{z}_{\mathcal{S}^{\perp}}\right\|^2$

- Three step proof:

1. Decompose $\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}_{\mathcal{S}} + \hat{\boldsymbol{w}}_{\mathcal{S}^{\perp}}$.
2. $\min_i y^{(i)}\, \hat{\boldsymbol{w}}.\boldsymbol{x}^{(\boldsymbol{i})} \geq 1 \Rightarrow \min_i y^{(i)}\, \hat{\boldsymbol{w}}_{\mathcal{S}}.\boldsymbol{x}^{(\boldsymbol{i})} \geq 1$
   (because $\hat{\boldsymbol{w}}_{\mathcal{S}^{\perp}}.\boldsymbol{x}^{(\boldsymbol{i})} = 0\ \forall i$)
3. if $\hat{\boldsymbol{w}}_{\mathcal{S}^{\perp}} \neq 0$, then $\|\hat{\boldsymbol{w}}_{\mathcal{S}}\| < \|\hat{\boldsymbol{w}}\|$

Representer Theorem

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \qquad s.t., \qquad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \quad \forall i$$

- Theorem: $\hat{\boldsymbol{w}} = span\{\boldsymbol{x}^{(i)}: i = 1,2,\ldots,N\}$ i.e.,

$\exists\{\hat{\beta}_i: i = 1,2,\ldots,N\}$ such that $\hat{\boldsymbol{w}} = \sum_i \hat{\beta}_i\ \boldsymbol{x}^{(i)}$

  - Special case of representor theorem

- Theorem (ext): additionally, $\{\hat{\beta}_i\}$ also stisfies $\hat{\beta}_i = 0$ for all i such that $y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} > 1$
- Proof?: (animation next slide)

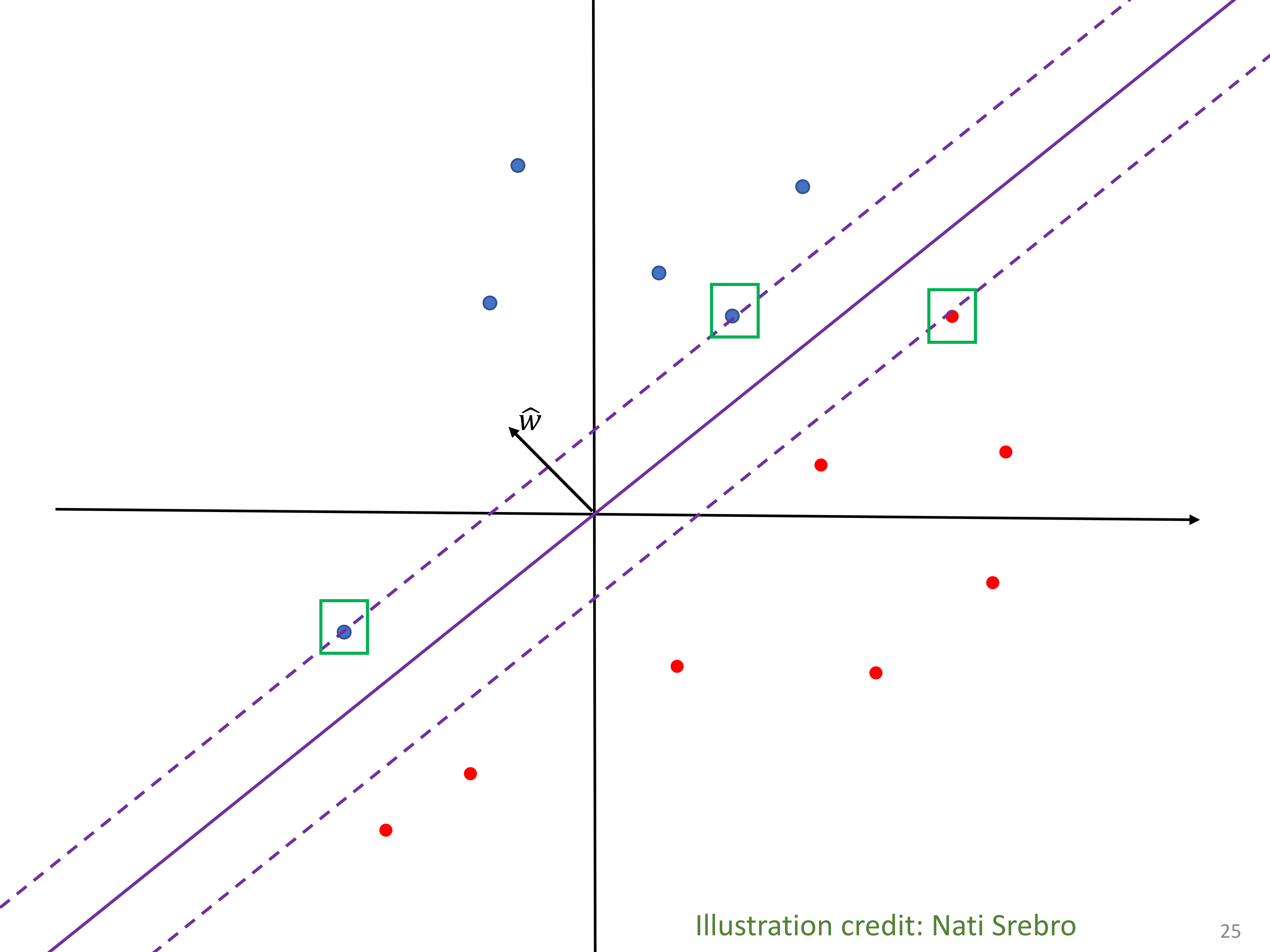


Illustration credit: Nati Srebro

Representer Theorem

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \quad s.t., \quad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \quad \forall i$$

- Theorem: $\exists\{\hat{\beta}_i : i = 1,2,\ldots,N\}$ such that $\widehat{\boldsymbol{w}} = \sum_i \hat{\beta}_i\ \boldsymbol{x}^{(\boldsymbol{i})}$

  $\{\hat{\beta}_i\}$ also satisfies $\hat{\beta}_i = 0$ for all i such that $y^{(i)}\ \widehat{\boldsymbol{w}}.\boldsymbol{x}^{(\boldsymbol{i})} > 1$
- $SV(\widehat{w}) = \{\boldsymbol{i} : y^{(i)}\ \widehat{\boldsymbol{w}}.\boldsymbol{x}^{(\boldsymbol{i})} = \boldsymbol{1}\}$ datapoints closest to $\widehat{w}$
  - called support vectors
  - hence support vector machine

$$\widehat{\boldsymbol{w}} = \sum_{i \in SV(\widehat{w})} \hat{\beta}_i \boldsymbol{x}^{(\boldsymbol{i})}$$

Representer Theorem

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \qquad s.t., \qquad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \quad \forall i$$

- Theorem: $\exists\{\hat{\beta}_i : i = 1,2,\ldots,N\}$ such that $\hat{\boldsymbol{w}} = \sum_i \hat{\beta}_i\ \boldsymbol{x}^{(\boldsymbol{i})}$

$\{\hat{\beta}_i\}$ also satisfies $\hat{\beta}_i = 0$ for all i such that $y^{(i)}\ \hat{\boldsymbol{w}}.\boldsymbol{x}^{(i)} > 1$

- $SV(\hat{w}) = \{\boldsymbol{i} : y^{(i)}\ \hat{\boldsymbol{w}}.\boldsymbol{x}^{(i)} = \boldsymbol{1}\}$ datapoints closest to $\hat{w}$
  - called support vectors
  - hence support vector machine

$$\hat{\boldsymbol{w}} = \sum_{i \in SV(\hat{w})} \hat{\beta}_i \boldsymbol{x}^{(\boldsymbol{i})}$$

How do we get $\hat{w}$?

Optimizing the SVM problem

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \quad s.t., \quad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \quad \forall i$$

1. Can do sub-gradient descent (next class)
2. Special case of quadratic program

$$\min_{\boldsymbol{z}} \frac{\mathbf{1}}{\mathbf{2}}\, \boldsymbol{z}^\top \boldsymbol{P} \boldsymbol{z} + \boldsymbol{q}^\top \boldsymbol{z}$$
$$s.t.\;\; \boldsymbol{G}\boldsymbol{z} \leq \boldsymbol{h}, \boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}$$

- Change of variables $\hat{\boldsymbol{w}} = \sum_{i \in SV(\hat{w})} \hat{\beta}_i \boldsymbol{x}^{(i)}$?
- Change of variables $\hat{\boldsymbol{w}} = \sum_{i=1}^{N} \hat{\beta}_i \boldsymbol{x}^{(i)}$!

Optimizing the SVM problem

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{w}\|^2 \quad s.t., \quad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \;\; \forall i$$

- Change of variables $\boldsymbol{w} = \sum_{i=1}^{N} \beta_i \boldsymbol{x}^{(i)}$!

$$\equiv \min_{\{\beta_i\}} \sum_{i=1}^{N} \sum_{j=1}^{n} \beta_i \beta_j \boldsymbol{x}^{(i)}.\boldsymbol{x}^{(j)} \;\; s.t. \;\; \sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} \beta_j\, y^{(i)} \boldsymbol{x}^{(i)}.\boldsymbol{x}^{(j)} \geq 1 \;\; \forall \boldsymbol{i}$$

$$= \min_{\boldsymbol{\beta} \in \mathbb{R}^{\boldsymbol{N}}} \boldsymbol{\beta}^\top \boldsymbol{G} \boldsymbol{\beta} \;\; s.t.\, y^{(i)} (\boldsymbol{G}\boldsymbol{\beta})_i \geq 1 \quad \forall i$$

- $\boldsymbol{G} \in \mathbb{R}^{N \times N}$ with $G_{ij} = \boldsymbol{x}^{(i)}.\boldsymbol{x}^{(j)}$ is called the gram matrix
- Convex program: quadratic programming

The Kernel

$$\min_{w} \|w\|^2 \quad s.t. \quad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})} \geq 1 \;\; \forall i$$

$$\equiv \min_{\boldsymbol{\beta} \in \mathbb{R}^{\boldsymbol{N}}} \boldsymbol{\beta}^\top \boldsymbol{G} \boldsymbol{\beta} \quad s.t.\, y^{(i)} (\boldsymbol{G}\boldsymbol{\beta})_i \geq 1 \;\; \forall i$$

- Optimization problem depends on $\boldsymbol{x}^{(\boldsymbol{i})}$ only through the values of $G_{ij} = \boldsymbol{x}^{(\boldsymbol{i})}.\boldsymbol{x}^{(\boldsymbol{j})}$ for $i, j \in [N]$.

The Kernel

$$\min_{w} \|w\|^2 \quad s.t. \quad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(i)} \geq 1 \;\; \forall i$$

$$\equiv \min_{\boldsymbol{\beta} \in \mathbb{R}^{\boldsymbol{N}}} \boldsymbol{\beta}^\top \boldsymbol{G} \boldsymbol{\beta} \quad s.t.\, y^{(i)}(\boldsymbol{G}\boldsymbol{\beta})_i \geq 1 \;\; \forall i$$

- Optimization problem depends on $\boldsymbol{x}^{(i)}$ only through the values of $G_{ij} = \boldsymbol{x}^{(i)}.\boldsymbol{x}^{(j)}$ for $i, j \in [N]$.
- What about prediction?

$$\widehat{\boldsymbol{w}}.\boldsymbol{x} = \sum_i \beta_i\, \boldsymbol{x}^{(i)}.\boldsymbol{x}$$

The Kernel

$$\min_{w} \|w\|^2 \quad s.t. \quad y^{(i)}\, \boldsymbol{w}.\boldsymbol{x}^{(\boldsymbol{i})} \geq 1 \;\; \forall i$$

$$\equiv \min_{\boldsymbol{\beta} \in \mathbb{R}^{\boldsymbol{N}}} \boldsymbol{\beta}^\top \boldsymbol{G}\boldsymbol{\beta} \quad s.t. \, y^{(i)}(\boldsymbol{G}\boldsymbol{\beta})_i \geq 1 \;\; \forall i$$

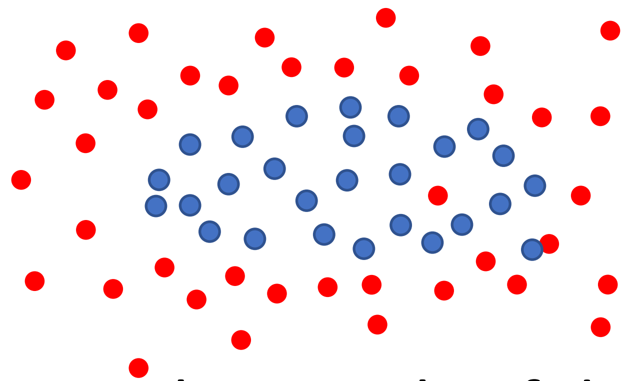
- Optimization problem depends on $\boldsymbol{x}^{(\boldsymbol{i})}$ only through the values of $G_{ij} = \boldsymbol{x}^{(\boldsymbol{i})}.\boldsymbol{x}^{(\boldsymbol{j})}$ for $i, j \in [N]$.
- What about prediction?

$$\widehat{\boldsymbol{w}}.\boldsymbol{x} = \sum_i \beta_i\, \boldsymbol{x}^{(\boldsymbol{i})}.\boldsymbol{x}$$

- Function $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}.\boldsymbol{x}'$ is called the Kernel
- Learning non-linear classifiers using feature transformations, i.e., $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}.\phi(\boldsymbol{x})$ for some $\phi(\boldsymbol{x})$
  - only thing we need to know is $K_\phi(\boldsymbol{x}, \boldsymbol{x}') = K(\phi(\boldsymbol{x}), \phi(\boldsymbol{x}'))$

Kernels As Prior Knowledge

- If we think that positive examples can (almost) be separated by some ellipse:



  then we should use polynomials of degree 2

- A Kernel encodes a measure of *similarity* between objects. A bit like NN, except that it must be a valid inner product function.

Slide credit: Nati Srebro