Project Groups

So far, I got the following groups:

1. **Hussain**, Tanvir; **Lewis**, Cameron; **Villamar**, Sandra
2. **Dong**, Meng; **Long**, Jianzhi; **Wen**, Bo; **Zhang**, Haochen
3. **Chen**, Yuzhao; **Li**, Zonghuan; **Song**, Yuze; **Yan**, Ge
4. **Li**, Jiayuan; **Xiao**, Nan; **Yu**, Nancy; **Zhou**, Pei
5. **Li**, Zheng; **Tao**; Jianyu; **Yang**, Fengqi
6. **Bian**, Xintong; **Jiang**, Yufan; **Wu**, Qiyao
7. **Chen**, Yongxing; **Yao**, Yanzhi; **Zhang**, Canwei
8. **Nukala**, Kishore; **Pulleti**, Sai; **Vaidyula**, Srikar
9. **Baluja**, Michael; **Cao**, Fangning; **Huff**, Mikael; **Shen**, Xuyang
10. **Arun**, Aditya; **Long**, Heyang; **Peng**, Haonan
11. **Cowin**, Samuel; **Hanna**, Aaron; **Liao**, Albert; **Mandadi**, Sumega
12. **Jia**, Yichen; **Jiang**, Zhiyun; **Li**, Zhuofan
13. **Dandu**, Murali; **Daru**, Srinivas; **Pamidi**, Sri
14. **Huang**, Yen-Ting; **Wang**, Shi; **Wang**, Tzu-Kao
15. **Chen**, Luobin; **Feng**, Ruining; **Wu**, Ximei; **Xu**, Haoran
16. **Chen**, Rex; **Liang**, Youwei; **Zheng**, Xinran
17. **Aguilar**, Matthew; **Millhiser**, Jacob; **O'Boyle**, John; **Sharpless**, Will
18. **Wang**, Haoyu; **Wang**, Jiawei; **Zhang**, Yuwei
19. **Chen**, Yinbo; **Di**, Zonglin; **Mu**, Jiteng
20. **Chowdhury**, Debalina; **He**, Scott; **Ye**, Yiheng
21. **Lin**, Wei-Ru; **Ru**, Liyang; **Zhang**, Shaohua
22. **Bhavsar**, Shivad; **Blazej**, Christopher; **Bu**, Yinyan; **Liu**, Haozhe
23. **Chen**, Claire; **Hsieh**, Chia-Wei; **Lin**, Jui-Yu; **Tsai**, Ya-Chen
24. **Cheng**, Yu; **Yu**, Zhaowei; **Zaidi**, Ali
25. **Assadi**, Parsa; **Brugere**, Tristan; **Pathak**, Nikhil; **Zou**, Yuxin
26. **Candassamy**, Gokulakrishnan; **Dixit**, Rajeev; **Huang**, Joyce

If you haven't sent me the composition of your group, please send me an email: mvasconcelos@eng.ucsd.edu with the group members. If I don't hear from you by Monday, 1/24 @ 11:59pm, I will assume that **you will not be doing a project and not taking the class for credit** (either letter–grade or S/U).

ECE 271B – Winter 2022

Linear Discriminants

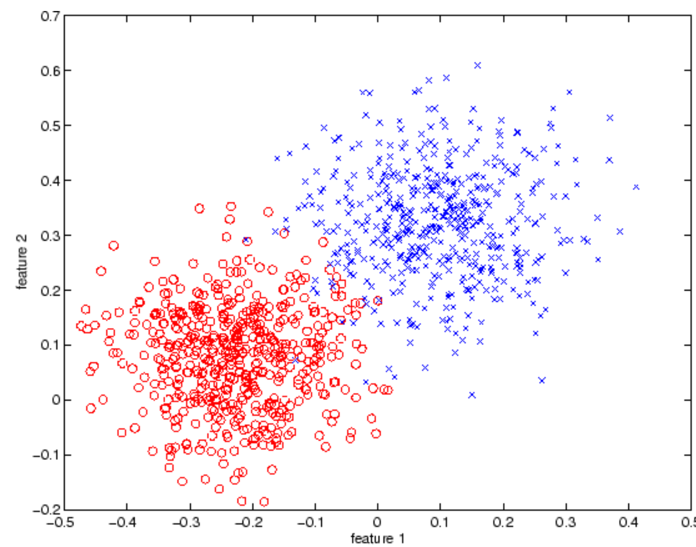
Disclaimer:
This class will be recorded
and made available to students asynchronously.

Manuela Vasconcelos

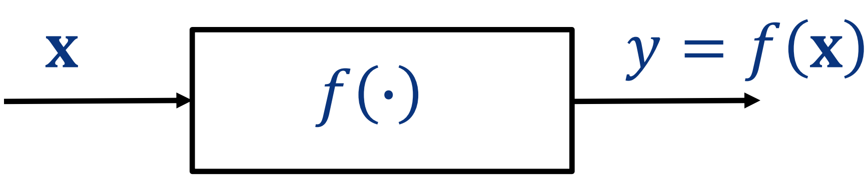
ECE Department, UCSD

Classification

- a **classification problem** has <u>two</u> types of variables
  - $\mathbf{x}$ – vector of observations (**features**) in the world
  - y – state (**class**) of the world
- e.g.
  - $\mathbf{x} \in \mathcal{X} \in \mathbb{R}^2 = (\text{fever}, \text{blood pressure})$
  - $\mathrm{y} \in \mathcal{Y} = \{\text{disease}, \text{no disease}\}$



- $\mathbf{x}$, y related by (unknown) function



- **goal**: design a **classifier** $h\colon \mathcal{X} \longrightarrow \mathcal{Y}$ such that $h(\mathbf{x}) = f(\mathbf{x}), \forall \mathbf{x}$

goal:
design classifier $h \colon \mathcal{X} \longrightarrow \mathcal{Y}$ s.t. $h(\mathbf{x}) = f(\mathbf{x}), \forall \mathbf{x}$

Loss Functions and Risk

- usually $h(\cdot)$ is parametric $h(\mathbf{x}, \boldsymbol{\alpha})$ and **cannot** approximate $f(\cdot)$ arbitrary well
- there is a loss/cost

$$\boxed{L[y, h(\mathbf{x}, \boldsymbol{\alpha})]}$$

  of making a prediction $h(\mathbf{x})$ when the true value is y

- goal: to find the **set of parameters** $\boldsymbol{\alpha}$ that minimize the expected value of the loss/cost, which is called the risk

$$R(\boldsymbol{\alpha}) = E_{\mathbf{X},Y}\{L[y, h(\mathbf{x}, \boldsymbol{\alpha})]\}$$

$$= \int P_{\mathbf{X},Y}(\mathbf{x}, y) L[y, h(\mathbf{x}, \boldsymbol{\alpha})] \, d\mathbf{x} \, dy$$

- Q: what is the **function** $h(\cdot)$ that minimizes the risk?

$$R(\boldsymbol{\alpha}) = E_{\mathbf{X},Y}\{L[y, h(\mathbf{x}, \boldsymbol{\alpha})]\}$$
$$= \int P_{\mathbf{X},Y}(\mathbf{x}, y) L[y, h(\mathbf{x}, \boldsymbol{\alpha})] \, d\mathbf{x} \, dy$$

Loss Functions and Risk

- Q: what is the **function** $h(\cdot)$ that minimizes the risk?

- since

$$R^* = \min_h E_{\mathbf{X},Y}\{L[y, h(\mathbf{x})]\} = \min_h E_{\mathbf{X}}\left\{E_{Y|\mathbf{X}}(L[y, h(\mathbf{x})]|\mathbf{x})\right\}$$

  the optimal decision function is

$$\boxed{h^*(\mathbf{x}) = \arg\min_h E_{Y|\mathbf{X}}\{L[y, h(\mathbf{x})]|\mathbf{x}\}, \forall \mathbf{x}}$$

- **classification**: "0–1" loss

$$\boxed{L[y, h(\mathbf{x}, \boldsymbol{\alpha})] = \begin{cases} 0, & y = h(\mathbf{x}, \boldsymbol{\alpha}) \\ 1, & y \neq h(\mathbf{x}, \boldsymbol{\alpha}) \end{cases}}$$

  is common because

$$R(\boldsymbol{\alpha}) = 0 \cdot P_{\mathbf{X},Y}[y = h(\mathbf{x}, \boldsymbol{\alpha})] + 1 \cdot P_{\mathbf{X},Y}[y \neq h(\mathbf{x}, \boldsymbol{\alpha})] = P_{\mathbf{X},Y}[y \neq h(\mathbf{x}, \boldsymbol{\alpha})]$$

Bayes Classifier

- under the "0−1" loss, this becomes

$$h^*(\mathbf{x}) = \arg\min_h P_{Y|\mathbf{X}}\left[y \neq h(\mathbf{x})|\mathbf{x}\right]$$
$$= \arg\min_h \left(1 - P_{Y|\mathbf{X}}[h(\mathbf{x})|\mathbf{x}]\right)$$
$$= \arg\max_h P_{Y|\mathbf{X}}\left[h(\mathbf{x})|\mathbf{x}\right]$$

and, since y is in a discrete set,

$$\boxed{h^*(\mathbf{x}) = \arg\max_i P_{Y|\mathbf{X}}\left[i|\mathbf{x}\right]}$$

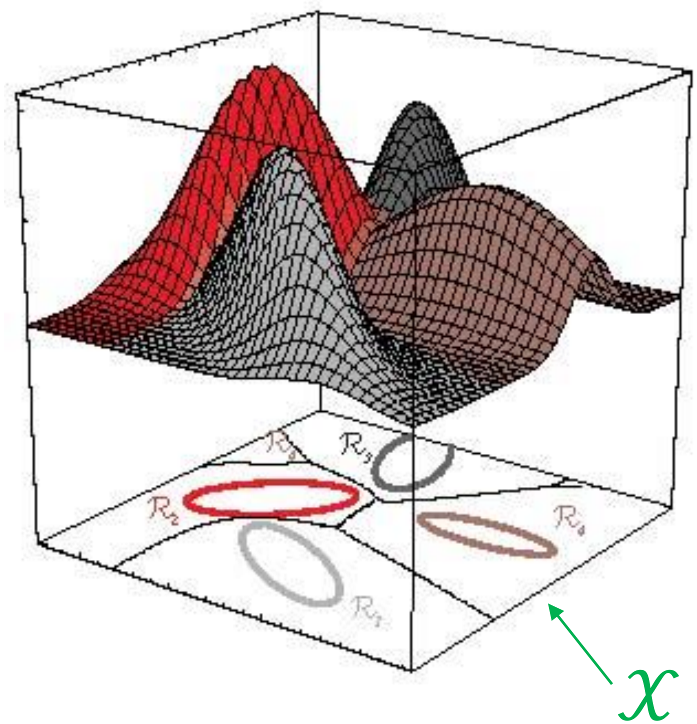
- the optimal decision is to pick the class of largest posterior probability
- this is the BDR – Bayes Decision Rule (**Bayes classifier**)

Bayes Decision Rule

- it carves up the observation space $\mathcal{X}$, assigning a label to each region

- clearly, h^* **depends** on the class densities

$$
\begin{aligned}
h^*(\mathbf{x}) &= \arg\max_i P_{Y|\mathbf{X}}[i|\mathbf{x}] \\
&= \arg\max_i \frac{P_{\mathbf{X}|Y}[\mathbf{x}|i]P_Y[i]}{P_{\mathbf{X}}[\mathbf{x}]} \\
&= \arg\max_i P_{\mathbf{X}|Y}[\mathbf{x}|i]P_Y[i] \\
&= \arg\max_i \{\log P_{\mathbf{X}|Y}[\mathbf{x}|i] + \log P_Y[i]\}
\end{aligned}
$$



Bayes Decision Rule

- this is **problematic**, since we **don't know** what these densities are
- in 271A, you have seen that density estimation is a **tricky** business
- key idea of **discriminant learning**:
  - estimating the densities to **then** derive the boundary is a **bad strategy**
    - density estimation is an "**ill–posed**" **problem** (slight change in problem conditions can lead to arbitrarily large change in the solution)
    - density estimation always has an **infinite number of solutions** (think of a Gaussian as a mixture of Gaussians)
  - Vapnik's rule:

  "when solving a problem, avoid solving a more general problem as an intermediate step!"

Discriminant Learning

- work directly with the decision function
  - postulate a (parametric) family of decision boundaries
  - pick the element in this family that produces the best classifier

- Q: what is a good family of decision boundaries?

- to get some insight, let's stick with the **BDR** a bit longer
- assume we have two **Gaussian** classes, equal covariance $\mathbf{\Sigma}$, equal probability $P_Y(i) = {}^1\!/_2\,, i \in \{0,1\}$
- notation: a Gaussian of mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ is

$$\boxed{G(\mathbf{x}, \boldsymbol{\mu}, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\mathbf{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}}$$

$$G(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

Discriminant Learning

- for two **equal** probability Gaussians of **equal** covariance

$$h^*(\mathbf{x}) = \arg\max_i \{\log P_{\mathbf{X}|Y}[\mathbf{x}|i] + \log P_Y[i]\}$$
$$= \arg\max_i \{\log G(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) + \log \tfrac{1}{2}\}$$
$$= \arg\min_i \{(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\}$$

which means

$$h^*(\mathbf{x}) = \begin{cases} 0, & \text{if } (\mathbf{x} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_0) < (\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_1) \\ 1, & \text{if } (\mathbf{x} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_0) > (\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_1) \end{cases}$$

Linear Discriminants

$$h^*(\mathbf{x}) = \begin{cases} 0, & \text{if } (\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) < (\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \\ 1, & \text{if } (\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) > (\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \end{cases}$$

- the decision boundary is the set of points

$$(\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) = (\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)$$

which, after some algebra manipulation, becomes

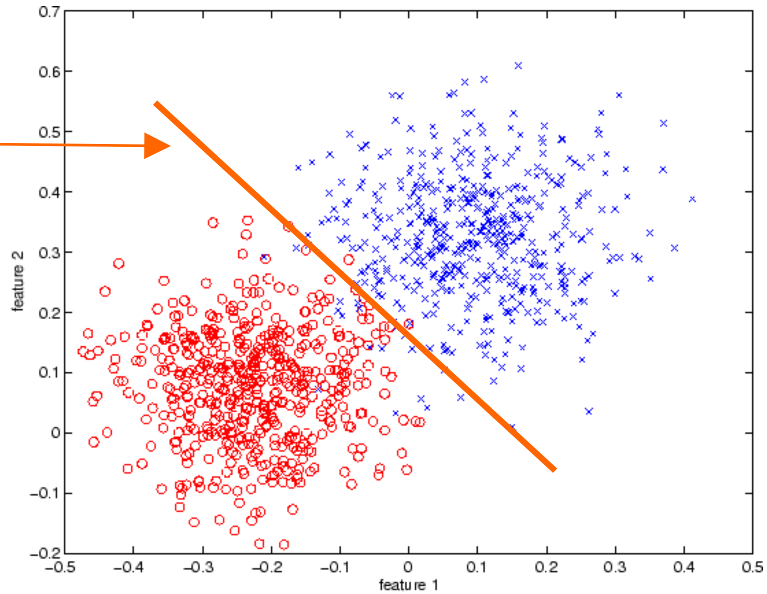
$$\boxed{2(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}}\mathbf{x} + \boxed{\boldsymbol{\mu}_0^T\,\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T\,\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1} = 0$$

- this is the equation of the hyper–plane

$$\boxed{\mathbf{w}^T\mathbf{x} + b = 0}$$

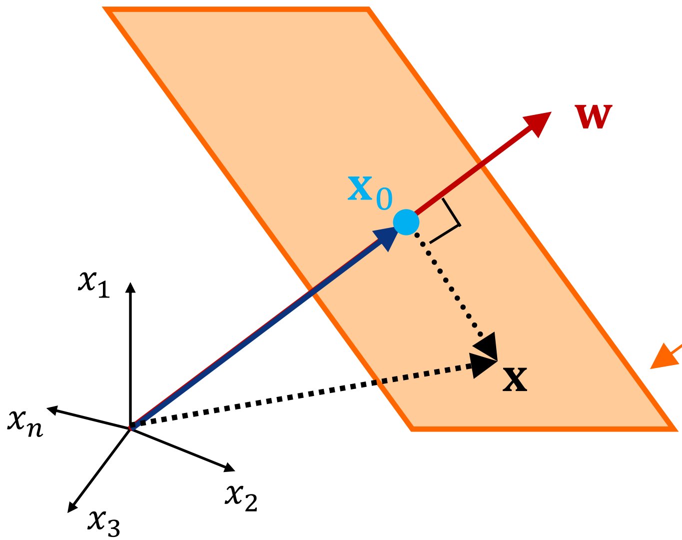
with

$$\mathbf{w} = 2\,\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)$$
$$b = \boldsymbol{\mu}_0^T\,\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T\,\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1$$



and we have a linear discriminant

Linear Discriminants



- the hyper−plane equation can also be written as

$$\boxed{\mathbf{w}^T\mathbf{x} + b = 0} \Leftrightarrow \mathbf{w}^T\left(\mathbf{x} + \frac{\mathbf{w}}{\|\mathbf{w}\|^2} b\right) = 0 \Leftrightarrow$$

$$\boxed{\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0} \quad \text{with} \quad \boxed{\mathbf{x}_0 = -b\frac{\mathbf{w}}{\|\mathbf{w}\|^2}}$$

- **geometric interpretation**
  - plane of normal $\mathbf{w}$
  - that passes through $\mathbf{x}_0$

Linear Discriminants

- under this notation, and after some algebra, the **decision function**

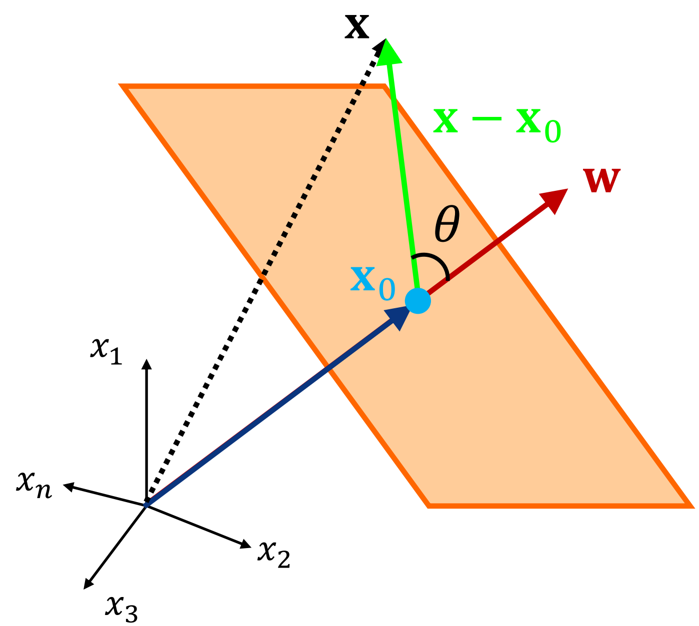
$$h^*(\mathbf{x}) = \begin{cases} 0, & \text{if } (\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) < (\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \\ 1, & \text{if } (\mathbf{x}-\boldsymbol{\mu}_0)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) > (\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \end{cases}$$

becomes

$$\boxed{h^*(\mathbf{x}) = \begin{cases} 1, & \text{if } g(\mathbf{x}) > 0 \\ 0, & \text{if } g(\mathbf{x}) < 0 \end{cases}}$$

with

$$\boxed{\begin{aligned} g(\mathbf{x}) &= \mathbf{w}^T(\mathbf{x}-\mathbf{x}_0) \\ &= \|\mathbf{w}\|\|\mathbf{x}-\mathbf{x}_0\|\cos\theta \end{aligned}}$$



- $g(\mathbf{x}) > 0$ if $\mathbf{x}$ is on the side $\mathbf{w}$ points to ("$\mathbf{w}$ points to the positive side")

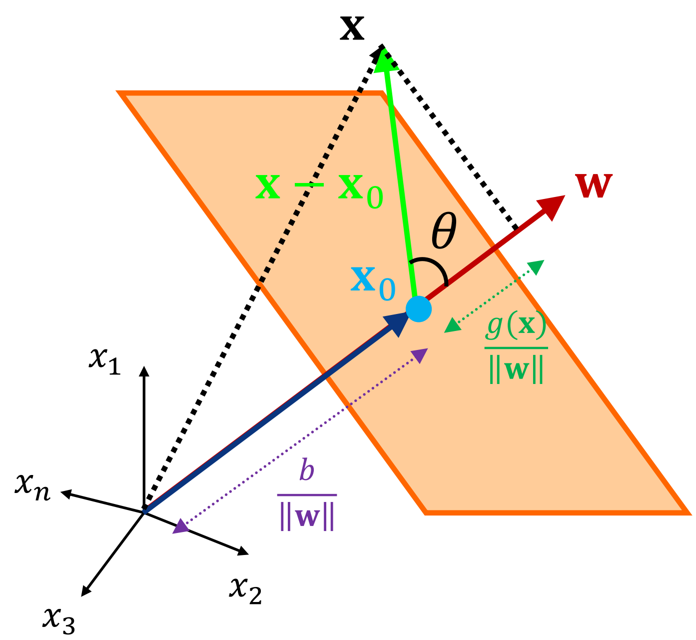
Linear Discriminants

- finally, note that

$$\boxed{\frac{g(\mathbf{x})}{\|\mathbf{w}\|} = \frac{\mathbf{w}^T}{\|\mathbf{w}\|}(\mathbf{x} - \mathbf{x}_0)}$$

  is

  - the projection of $\mathbf{x} - \mathbf{x}_0$ onto the unit vector in the direction of $\mathbf{w}$
  - length of the component of $\mathbf{x} - \mathbf{x}_0$ orthogonal to the plane



  i.e. $g(\mathbf{x})/\|\mathbf{w}\|$ is the perpendicular distance from $\mathbf{x}$ to the plane

- similarly, $b/\|\mathbf{w}\|$ is the distance from the plane to the origin since

$$\mathbf{x}_0 = -b\frac{\mathbf{w}}{\|\mathbf{w}\|^2}$$

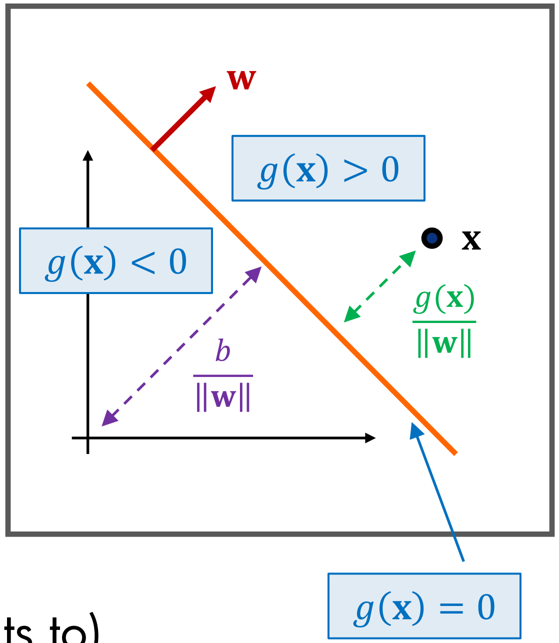
Geometric Interpretation

- in **summary**, the decision rule

$$\boxed{h^*(\mathbf{x}) = \begin{cases} 1, & \text{if } g(\mathbf{x}) > 0 \\ 0, & \text{if } g(\mathbf{x}) < 0 \end{cases}} \qquad \boxed{g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b}$$

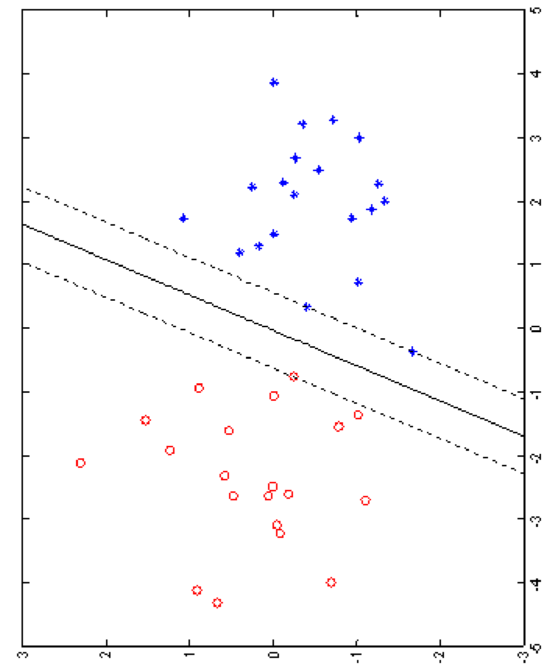
has the **properties**

- it divides $\mathcal{X}$ into two "half–planes"
- boundary is the plane with
  - normal $\mathbf{w}$
  - distance to the origin $b/||\mathbf{w}||$
- $g(\mathbf{x})/||\mathbf{w}||$ is the distance from point $\mathbf{x}$ to the boundary
  - $g(\mathbf{x}) = 0$ for points on the plane
  - $g(\mathbf{x}) > 0$ on the "positive side" (side $\mathbf{w}$ points to)
  - $g(\mathbf{x}) < 0$ on the "negative side"



Linear Discriminants

- is this a **good** decision function?
- **just seen** – it is **optimal** for
  - Gaussian classes
  - equal class probability and covariance
  - sounds **too much** as a "toy problem"
- also, **optimal** if data is **linearly separable**
  - there is a plane which has
    - all 0's on one side
    - all 1's on the other
- what **if none of these hold?**

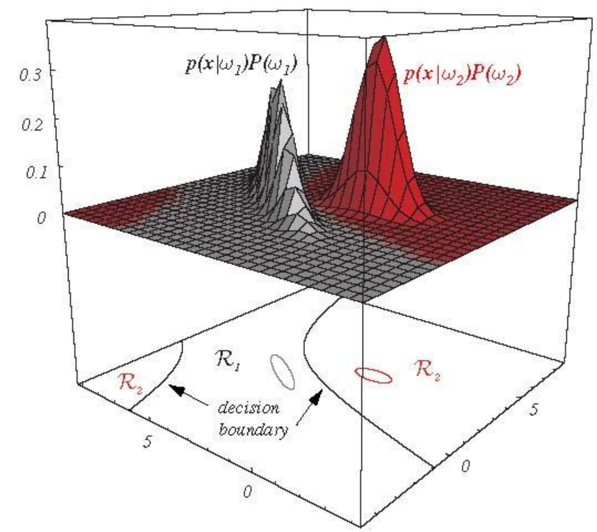


Alternatives

- 1) use a higher–order decision function
  - e.g. a quadratic boundary

    $$\mathbf{x}^T \mathbf{W}\,\mathbf{x} + \mathbf{w}^T \mathbf{x} + \mathbf{w}_0 = 0$$

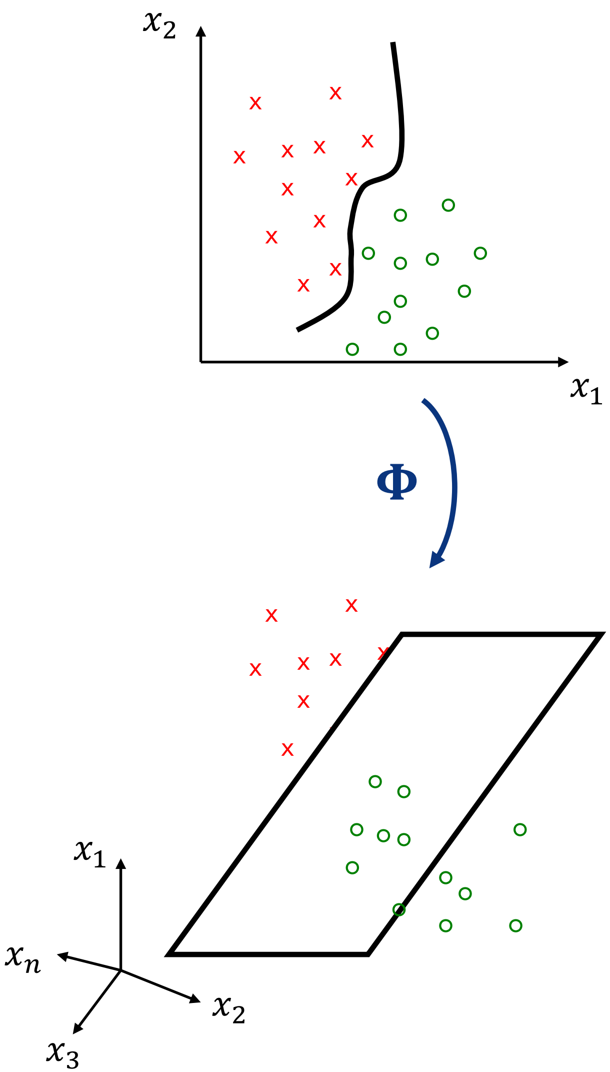
    is the optimal solution for **any** Gaussian problem (2 Gaussian classes, no constraints)



- looks like we are going to need a <u>very</u> high–order polynomial in general!
- **lots** of parameters
- **too** much complexity
- where to stop?
- can we do something else to keep the <u>simplicity</u> of the **linear boundary**?

Alternatives

- 2) transform the space
  - introduce a mapping
    $$\mathbf{\Phi}\colon \mathcal{X} \to \mathcal{Z}$$
    such that $\dim(\mathcal{Z}) > \dim(\mathcal{X})$
  - learning a **linear** boundary in $\mathcal{Z}$ is equivalent to learning a **non−linear** boundary in $\mathcal{X}$
  - **basic idea**
    - if transformed space is high−dimensional enough
    - any finite set of points can be separated linearly



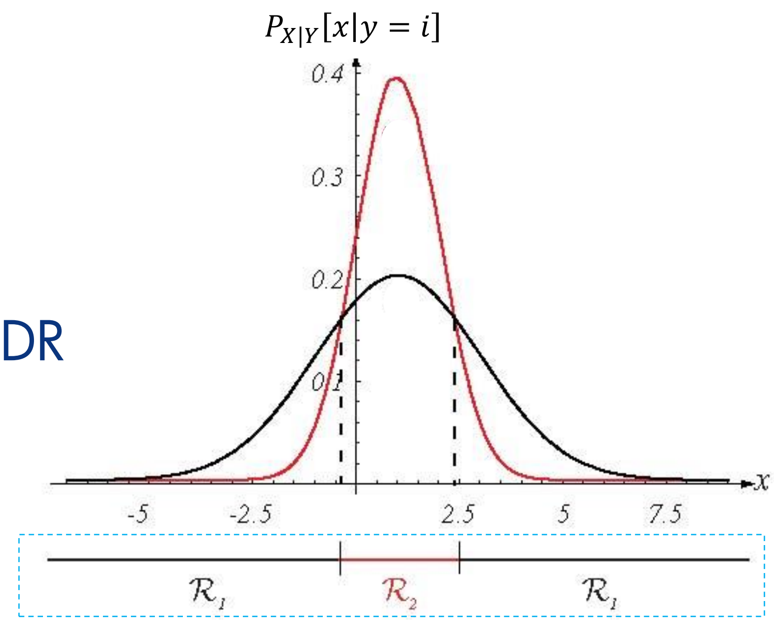
$\mathbf{\Phi}: \mathcal{X} \to \mathcal{Z}$ such that $\dim(\mathcal{Z}) > \dim(\mathcal{X})$

Feature Transformation

- e.g.
  - two scalar Gaussians
  - zero mean, different variances
- since $P_{X|Y}(x|i) = G(x, 0, \sigma_i)$, using the BDR

$$h^*(x) = \arg\max_i P_{X|Y}[x|i]\, P_Y[i]$$

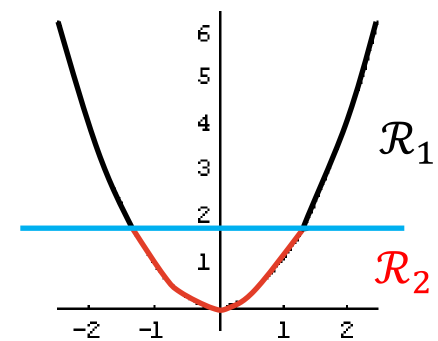
leads to this



$\mathbf{\Phi}$

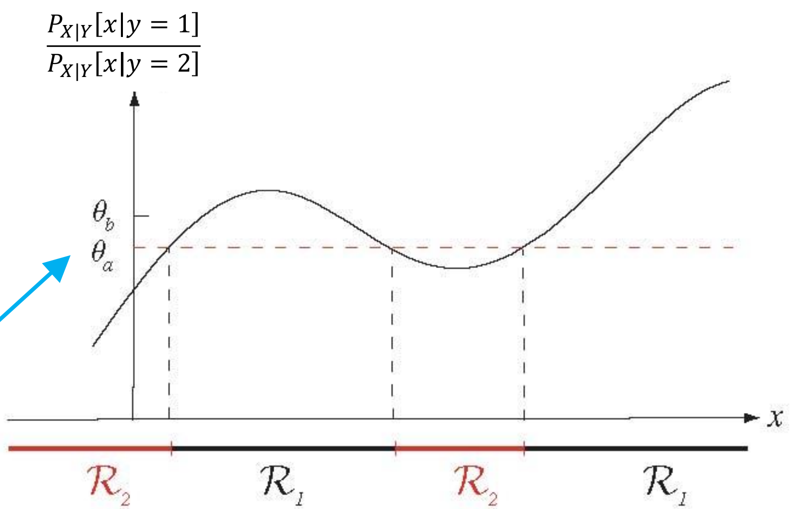
- which cannot be implemented with a linear discriminant
- but becomes feasible by mapping to 2D

$$\mathbf{\Phi}: \mathbb{R} \to \mathbb{R}^2$$
$$x \to (x, x^2)$$



Feature Transformation

- note that the **problem has <u>not</u> really changed**
  - we still have a 1D set
  - but now <u>**embedded**</u> in a 2D space
  - a lot **more** space: we can always arrange things so that the boundary is linear
  - the BDR itself tells us how to do this
  - but, once again, <u>**requires the densities**</u>
  - easier as the $\dim(\mathcal{Z})$ grows
  - usually feasible, as $\dim(\mathcal{Z}) \rightarrow \infty$
  - the problem is that evaluating $\mathbf{\Phi}(\mathbf{x})$ becomes **harder and harder**



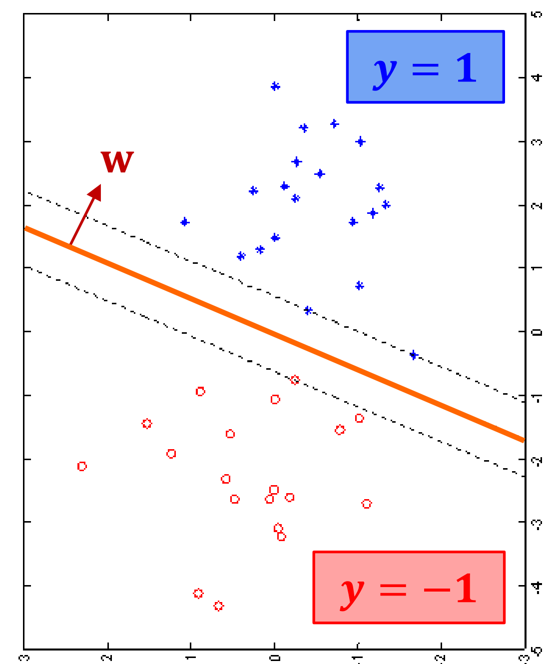
- we will see <u>how</u> to do this (<u>NNs</u>, <u>boosting</u>, <u>kernels</u>)

Back to Linear Discriminants

- for now, the goal is to explore the <u>**simplicity**</u> of the **linear discriminant**

- let's assume <u>linear separability</u>

- one **handy trick** is to use $y \in \{-1,1\}$ instead of $y \in \{0,1\}$, where
  - $y = 1$ for points on the positive side
  - $y = -1$ for points on the negative side



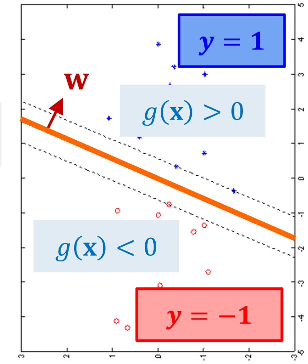
- the decision function becomes

$$\boxed{h^*(\mathbf{x}) = \begin{cases} 1, & \text{if } g(\mathbf{x}) > 0 \\ -1, & \text{if } g(\mathbf{x}) < 0 \end{cases}} \iff \boxed{h^*(\mathbf{x}) = \operatorname{sgn}[g(\mathbf{x})]}$$

Back to Linear Discriminants

$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$

$g(\mathbf{x}) > 0$ on the side $\mathbf{w}$ points to ("positive side")
$g(\mathbf{x}) < 0$ on the "negative side"



- we have a classification <u>error</u> if
  - $y = 1$ and $g(\mathbf{x}) < 0$ or $y = -1$ and $g(\mathbf{x}) > 0$

    i.e. $\boxed{y\, g(\mathbf{x}) < 0}$

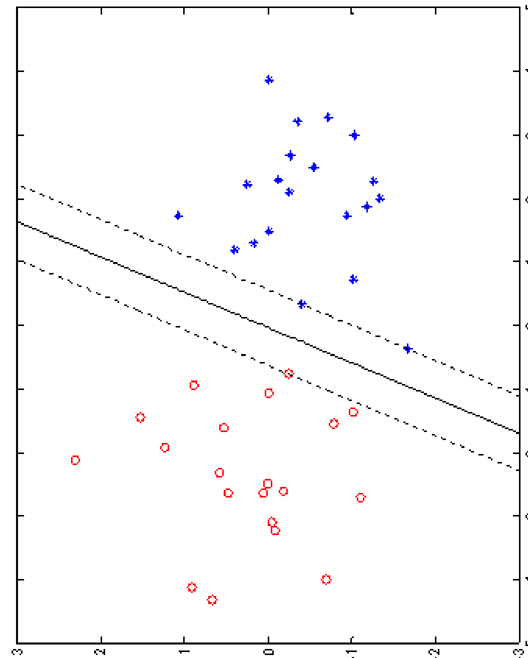
- and a <u>correct</u> classification if
  - $y = 1$ and $g(\mathbf{x}) > 0$ or $y = -1$ and $g(\mathbf{x}) < 0$

    i.e. $\boxed{y\, g(\mathbf{x}) > 0}$

- note that, since the data is <u>linearly separable</u>, given a training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, we can have <u>zero empirical risk</u>
- the necessary and sufficient condition is that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0, \forall i$$

Linear Discriminants



- in **summary**, a linear classifier can be a good decision function if data is <u>**linearly separable**</u>

- given a training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, we can have <u>**zero empirical risk**</u> if

$$\boxed{y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0, \forall i}$$

- note, however,
  - this holding on the training set only guarantees **optimality** on the **ERM** (Empirical Risk Minimization) **sense**

    Recall: training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ – we estimate the risk by the **empirical risk** (ER) in the training set

    $$R_{emp}(\boldsymbol{\alpha}) = \frac{1}{n}\sum_{i=1}^{n} L[y_i, h(\mathbf{x}_i, \alpha)]$$

  - **not** in the sense of **minimizing the <u>true</u> risk**

The Four Fundamental Questions

- Q: does Empirical Risk Minimization (ERM) assure the **minimization** of the risk?

$$R_{emp}(\boldsymbol{\alpha}) = \frac{1}{n}\sum_{i=1}^{n} L[y_i, g(\mathbf{x}_i, \boldsymbol{\alpha})]$$

- **Vapnik** and **Chervonenkis** studied this question extensively and identified four fundamental questions

  1. What are the necessary and sufficient conditions for **consistency** of ERM, i.e. **convergence**?
  2. How **fast** is the rate of convergence? If n needs to be very large, ERM is useless in practice since we only have a **finite training set**.
  3. Is there a way to **control** the rate of convergence?
  4. How can we design **algorithms** to control this rate?

- the formal answer to these questions requires a mathematical sophistication beyond what we require here

The Four Fundamental Questions

$$R_{emp}(\boldsymbol{\alpha}) = \frac{1}{n}\sum_{i=1}^{n} L[y_i, g(\mathbf{x}_i, \boldsymbol{\alpha})]$$

1. What are the necessary and sufficient conditions for **consistency** of ERM, i.e. **convergence**?
2. How **fast** is the rate of convergence? If n needs to be very large, ERM is useless in practice since we only have a **finite training set**.
3. Is there a way to **control** the rate of convergence?
4. How can we design **algorithms** to control this rate?

- I will try to convey the main ideas as we go along
- the **nutshell answers** are:
  1. Yes, ERM is **consistent**.
  2. The convergence rate is **quite slow**, only asymptotic guarantees are available.
  3. Yes, there is a **way to control** the rate of convergence, but it requires a different principle which Vapnik and Chervonenkis called Structural Risk Minimization (SRM).
  4. We will talk about this.
- it turns out that SRM is an **extension** of ERM

SRM vs ERM

- ERM minimizes **only** training loss – the problem is that more complicated functions always produce smaller training loss
- to **guarantee good generalization**, we need to penalize complexity
- Vapnik and Chervonenkis formalized this idea by showing that

$$\boxed{R(\boldsymbol{\alpha}) \leq R_{emp}(\boldsymbol{\alpha}) + \Phi(n, g)}$$

- $\Phi(n, g)$ is a confidence interval that depends on
  - number of training points n
  - VC dimension of the family of functions $g(x, \boldsymbol{\alpha})$
- **VC dimension**:
  - a measure of complexity, usually a function of the number of parameters
  - we will talk more about this

SRM vs ERM

- note that minimizing the bound provides guarantees on the risk even when the training set is finite!
- significance:
  - this is much **more relevant** in practice than the classical results which only give asymptotic guarantees
  - the bound inspires a practical way to control the **generalization ability**
- **controlling generalization**:

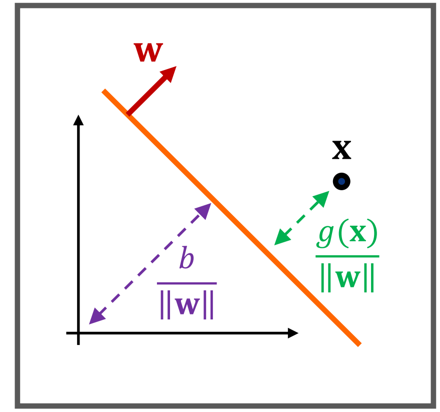
$$R(\boldsymbol{\alpha}) \leq R_{emp}(\boldsymbol{\alpha}) + \Phi(n, g)$$

  - given the function family,
    - the first term only depends on parameters
    - the second term depends on the family of functions
- **in practice**, this is achieved by introducing a $\boxed{\text{margin}}$

The Margin

- the **margin** is the distance from the boundary to the closest point

$$\gamma = \min_i \frac{|g(\mathbf{x}_i)|}{||\mathbf{w}||} = \min_i \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{||\mathbf{w}||}$$

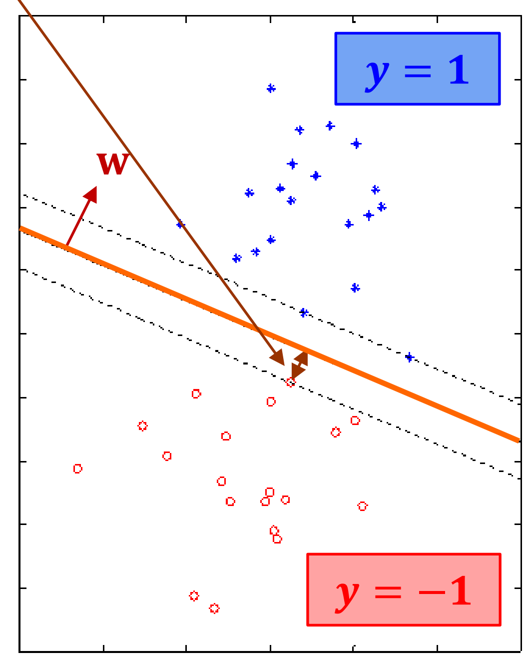


- among all planes such that

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 0, \forall i$$

there will be no error if it is strictly greater than zero

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) > 0, \forall i \quad \Leftrightarrow \quad \boxed{\gamma > 0}$$

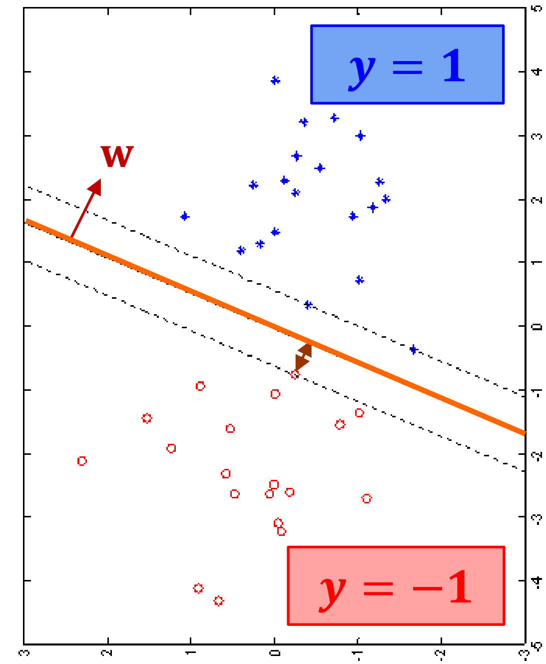


- note that this is **ill−defined** in the sense that γ does not change if both $\mathbf{w}$ and b are scaled by λ → we need **normalization**

The Margin

$$\gamma = \min_i \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

- this is **similar** to what we have seen for Fisher discriminants
  - a **natural** normalization is $\|\mathbf{w}\| = 1$
  - however, it introduces a **quadratic constraint** and complicates optimization

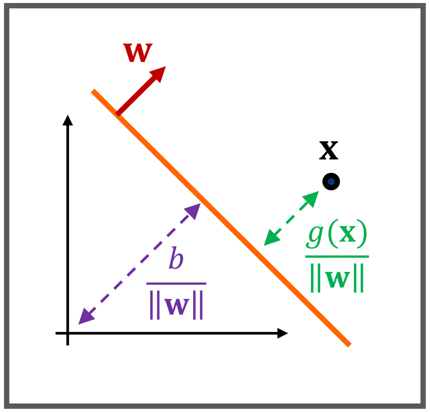


- a more <u>convenient</u> normalization is to make $|g(\mathbf{x})| = 1$ **for the closest point**, i.e.

$$\boxed{\min_i |\mathbf{w}^T \mathbf{x}_i + b| = 1}$$

under which

$$\boxed{\gamma = \frac{1}{\|\mathbf{w}\|}}$$



Support Vector Machines

- under this normalization,

$$|\mathbf{w}^T\mathbf{x}_i + b| \geq 1, \forall i$$

$$\Leftrightarrow \ [\mathrm{sgn}(\mathbf{w}^T\mathbf{x}_i + b)](\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i$$

$$\Leftrightarrow \ \boxed{y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i}$$

$$\boxed{\gamma = \frac{1}{\|\mathbf{w}\|}}$$

- the SVM is the classifier that maximizes the margin under this set of constraints, i.e.

$$\boxed{\min_{\mathbf{w},b} \ \|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i}$$

Relationship to SRM

- the SRM (Structural Risk Minimization) **principle**:
  - start from a nested collection of **families of functions**

    $$S_1 \subset \cdots \subset S_k$$

    where $S_i = \{h_i(\mathbf{x}, \boldsymbol{\alpha}), \forall \boldsymbol{\alpha}\}$

  - for each S_i, find the function (set of parameters) that **minimizes the empirical risk**

    $$R^i_{emp} = \min_{\boldsymbol{\alpha}} \frac{1}{n} \sum_{k=1}^{n} L[y_k, h_i(\mathbf{x}_k, \boldsymbol{\alpha})]$$

  - select the **function class** such that

    $$R^* = \min_i \left\{R^i_{emp} + \Phi(h_i)\right\}$$

    where $\Phi(h)$ is a function of the VC dimension (complexity) of the family S_i

Relationship to SRM

$$\min_{\mathbf{w},b} \|\mathbf{w}\|^2 \quad \text{subject to} \quad \boxed{y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i}$$

- here:
  - S_i is the family of hyperplanes such that $\|\mathbf{w}\| < \lambda_i$
  - the constraints guarantee that $R_{emp}^i = 0$
  - and the VC dimension $\Phi(h)$ (complexity) is upper−bounded by λ_i (more on this later)
- i.e. the SVM **minimizes an upper−bound of** $\Phi(h)$, while **maintaining** R_{emp}^i **zero**
- since

$$R \leq R_{emp} + \Phi(h)$$

this provides **guarantees** on the risk (more later)

Intuitively

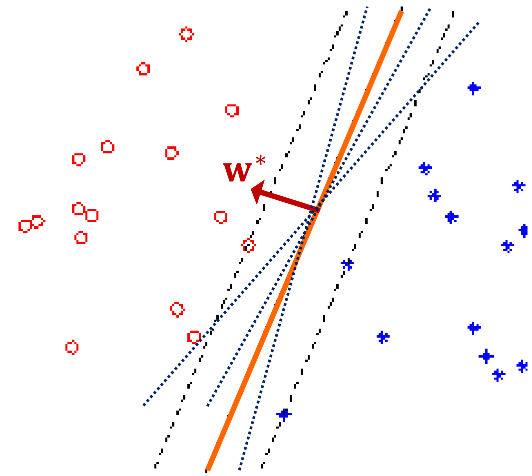
- this is penalizing complexity

$$\min_{\mathbf{w},b} ||\mathbf{w}||^2 \quad \text{subject to} \quad \boxed{y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \forall i}$$

- searching for the **more stable** hyperplane
  - among the ones that have zero training error
  - is the one that has **most room** for **discrepancies** between training and testing
  - the **margin** as a "**security gap**"

$$\gamma = \frac{1}{||\mathbf{w}||}$$

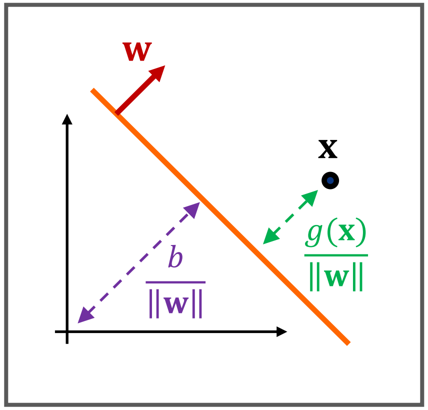
all these planes satisfy the constraints



- there are **many details which we have not filled** (more later)

Homework

- next class, we will go over the Perceptron, which is a good classifier to <u>gain insight</u> on
  - the role of the margin
  - duality
  - optimization



- like almost everything we will do in this course, it will require a very good understanding of this picture

- we will also use expressions like

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) > 0, \forall i$$

  all the time

- you should make yourself familiar with these!!!