Machine Learning

4771

Instructor: Itsik Pe’er

Example quiz question

- The perceptron loss function
  - +3 induces a discontinuous, stair-like empirical risk function F
  - +3 is differentiable anywhere F
  - +3 is the number of misclassified datapoints F
  - +3 penalizes misclassified datapoints proportionally to $|f(x;\theta)|$ T

Example quiz question

Convergence proof of online perceptron learning

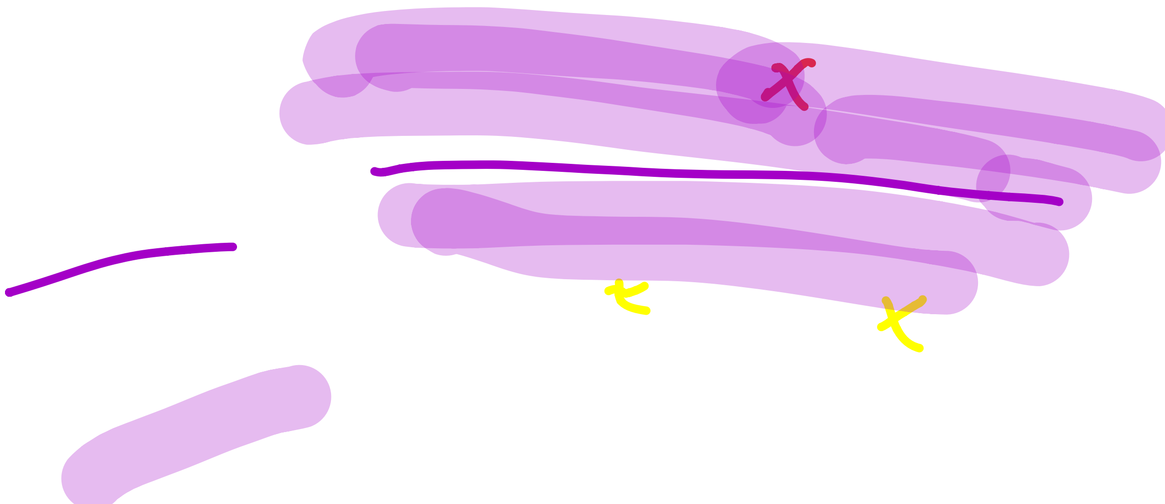
- Bounds the length of the current parameter vector from above T
- Bounds from above the projection of the current parameter vector on the optimal one F
- Relies on an optimal gap-tolerant classifier T
- Is part of the material for the quiz F

Class 8 SVMs

- Theoretical motivation
- Formulation
- Dual problem

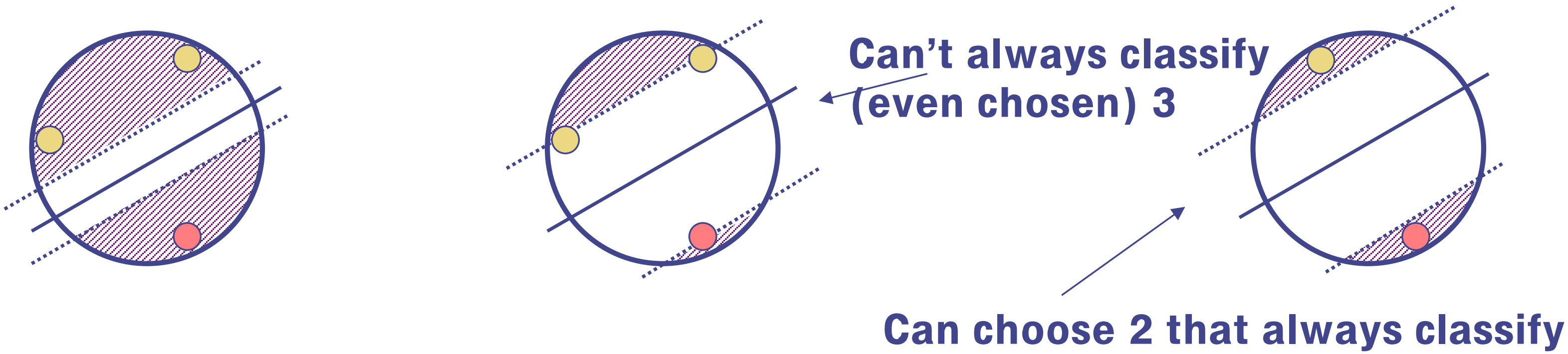
True Risk Bound & Gaps

- Wider gap (w.r.t. universe) means the function family is weaker



True Risk Bound & Gaps

- Wider gap (w.r.t. universe) means the function family is weaker



- Weakest family = widest gap

Empirical Risk Minimization

- Recall ERM: $R_{emp}(\theta) = \frac{1}{N}\sum_i^N Loss(y_i, f(\boldsymbol{x}_i;\theta)) \in [0,1]$
- Empirical $R_{emp}(\theta)$ *approximates* the true risk (expected error)

$$R(\theta) = E_P\{Loss(y, \boldsymbol{x}, \theta)\} = \int_{\boldsymbol{X}\times Y} P(\boldsymbol{x}, y) Loss(y, \boldsymbol{x}, \theta)\, dxdy \in [0,1]$$

- But, we don't know the true $P(\boldsymbol{x},y)$!
- Good news: for any θ, if infinite data, by *law of large numbers*:

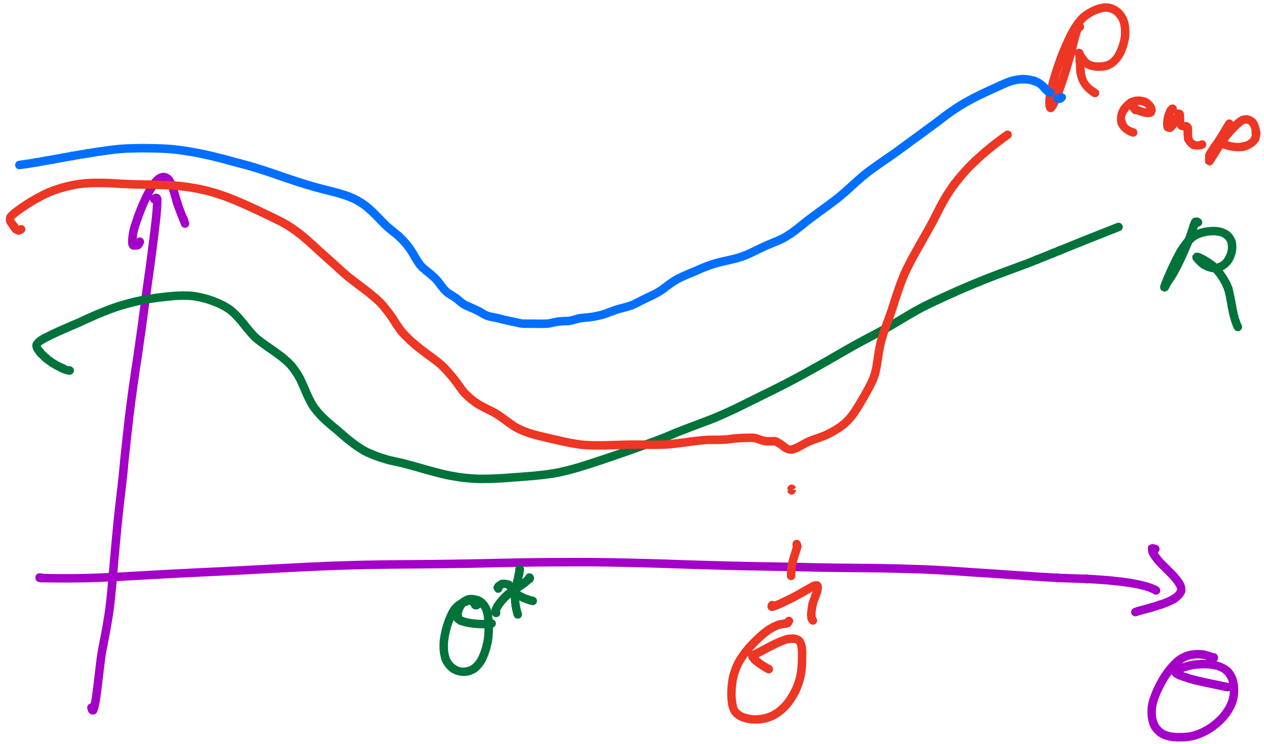
$$\lim_{n\to\infty} R_{emp}(\theta) = R(\theta)$$

- Bad news: ERM may not converge to optimum even if $N\to\infty$:

$$argmin_\theta R_{emp}(\theta) \neq argmin_\theta R(\theta)$$

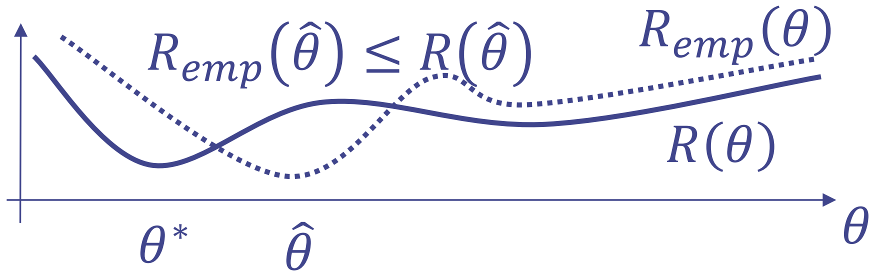
...ERM is not consistent

Bounding the True Risk



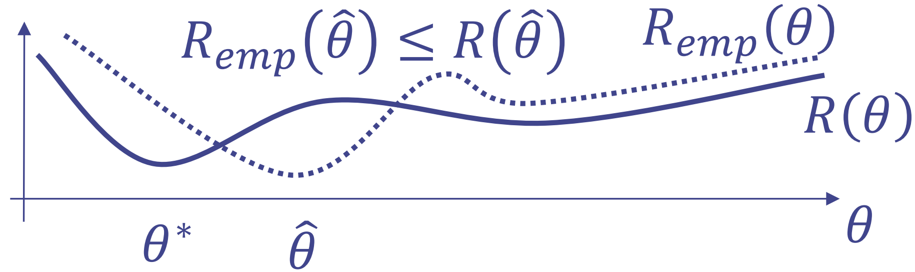
Bounding the True Risk

- ERM's risk is not guaranteed since it may do better on training than on test!

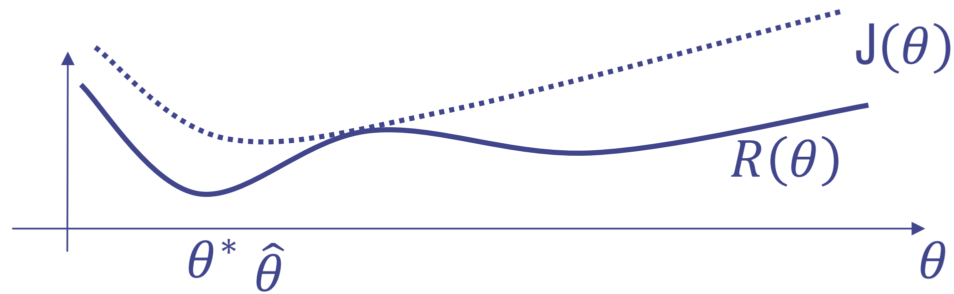


Bounding the True Risk

- ERM's risk is not guaranteed since it may do better on training than on test!



- Idea: add a prior or regularizer to $R_{emp}(\theta)$
- Define capacity or confidence $C(\theta)$ which favors simpler θ
- If $J(\theta) = R_{emp}(\theta) + C(\theta) \geq R(\theta)$, then it is guaranteed risk



- After train, can guarantee future error rate is $\leq \min_{\theta} J(\theta)$
- Structural Risk Minimization: minimize risk bound $J(\theta)$

Bound the True Risk with VCD

- Idea: Rely on the capacity of the classifier class $f(.;\theta)$
  - $h \triangleq$ # of datasets it can perfectly classify ($\neq$ # parameters!)
  - Independent of the true $P(x,y)$ so gives *worst case bound*

- Theorem (Vapnik):

With probability 1-η where $\eta \in [0,1]$, $R(\theta) \leq J(\theta)$ where:

$$J(\theta) = R_{emp}(\theta) + \frac{2h\log\left(\frac{2eN}{h}\right)+2\log\left(\frac{4}{\eta}\right)}{N}\left(1+\sqrt{1+\frac{NR_{emp}(\theta)}{h\log\left(\frac{2eN}{h}\right)+\log\left(\frac{4}{\eta}\right)}}\right)$$

N = number of data points

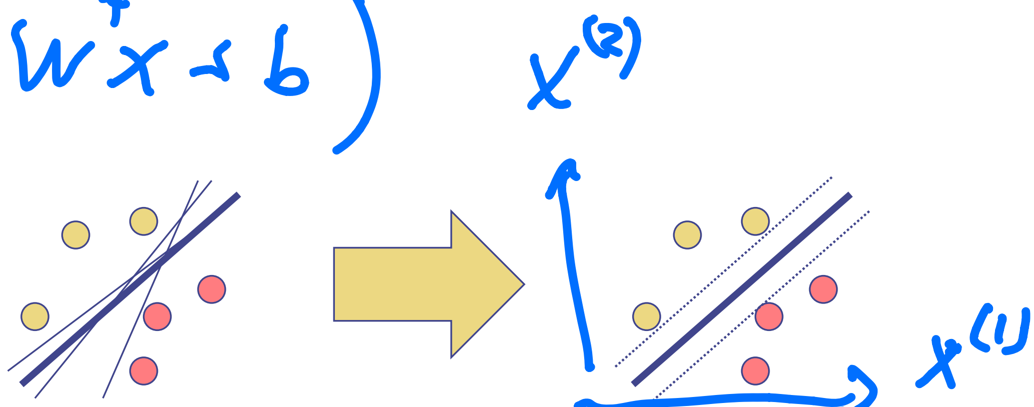
h = Vapnik-Chervonenkis (VC) dimension (1970's)
measure classifying ability of a function family

Support Vector Machines

- Support vector machines are (in the simplest case) linear classifiers that do structural risk minimization (SRM)
- Directly maximize margin to reduce guaranteed risk $J(\theta)$

$\{(x_i, y_i)\}$ $\quad x_i \in R^D \quad y_i = \{+1, -1\}$

$\mathrm{sign}(w^T x + b)$



$D = 2$

SVM Notation

- Assume first the 2-class data is linearly separable: $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_i \in \boldsymbol{R}^D$ and $y_i \in \{-1,1\}$ and $f(\boldsymbol{x}; \theta = (w, b)) = sign(w^T x + b\ )$

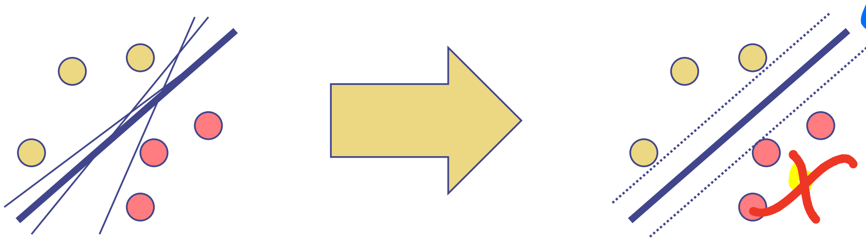
$y_i = 1: \quad w^T x_i + b \geq c$

$y_i = -1: \quad w^T x_i + b \leq -c$

$w^T x + b = c$

$w^T x + b = 0$

$w^T x + b = -c$

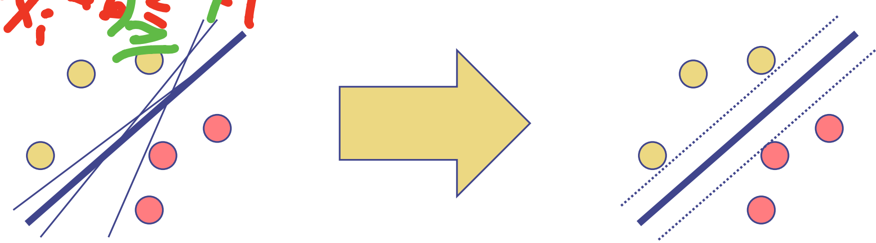


SVM Notation

- Assume first the 2-class data is linearly separable: $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_i \in \boldsymbol{R}^D$ and $y_i \in \{-1,1\}$ and $f(\boldsymbol{x}; \theta = (w, b)) = sign(w^T x + b\ )$
- Decision boundary or hyperplane given by $w^T x + b = 0$
- Many solutions exist which have empirical error $R_{emp}(\theta)=0$
- Want widest or thickest one (max margin), also it's unique!
- Define margins $w^T x + b = +\cancel{C}\ 1$ $\quad w^T x + b = -\cancel{C}\ 1$

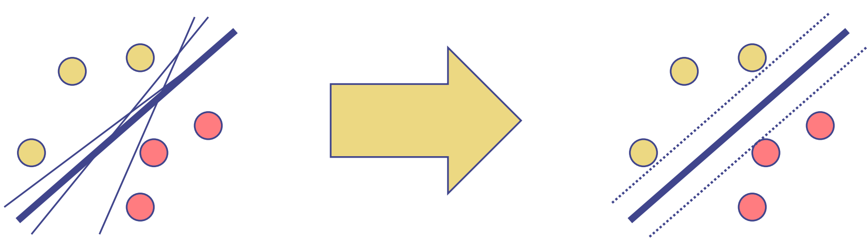
$y_i = 1: \ y_i(w^T x_i + b) \geq 1$

$y_i = -1: \ y_i(w^T x_i + b) \cancel{\leq -1} \geq 1$



SVM Notation

- Assume first the 2-class data is linearly separable: $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_i \in \boldsymbol{R}^D$ and $y_i \in \{-1,1\}$ and $f(\boldsymbol{x}; \theta = (w,b)) = sign(w^T x + b)$
- Decision boundary or hyperplane given by $w^T x + b = 0$
- Many solutions exist which have empirical error $R_{emp}(\theta)=0$
- Want widest or thickest one (max margin), also it's unique!
- Define margins $w^T x + b = +C$ $w^T x + b = -C$
- Note: can scale w,b,C WLOG so set $C = 1$

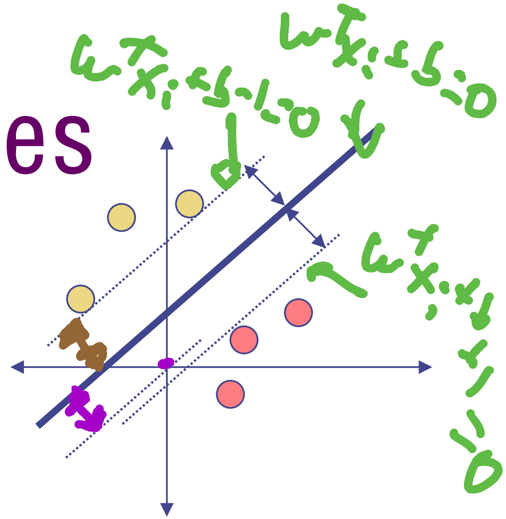


Support Vector Machines

- The constraints on the SVM for $R_{emp}(\theta)=0$ are thus:

$$w^T x_i + b \geq +1 \quad \forall y_i = +1$$
$$w^T x_i + b \leq -1 \quad \forall y_i = -1$$

- Or more simply: $y_i(w^T x_i + b) - 1 \geq 0$



$\frac{1}{||w||}$

$\frac{|b|}{||w||}$ $\frac{|b \pm 1|}{||w||}$

$\arg\max \frac{2}{||w||}$

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

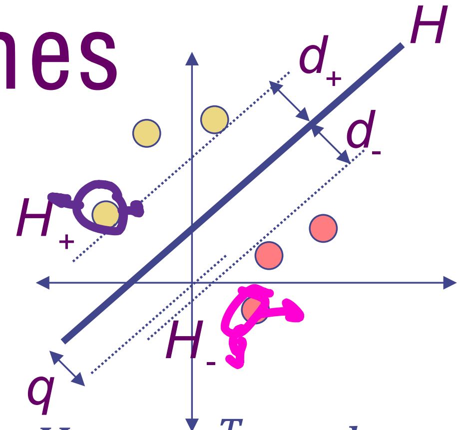
Support Vector Machines

- The constraints on the SVM for $R_{emp}(\theta)=0$ are thus:

$$w^T x_i + b \geq +1 \quad \forall y_i = +1$$

$$w^T x_i + b \leq -1 \quad \forall y_i = -1$$

- Or more simply: $y_i(w^T x_i + b) - 1 \geq 0$
- The margin of the SVM is: $m = d_+ + d_-$
- Distance to origin:

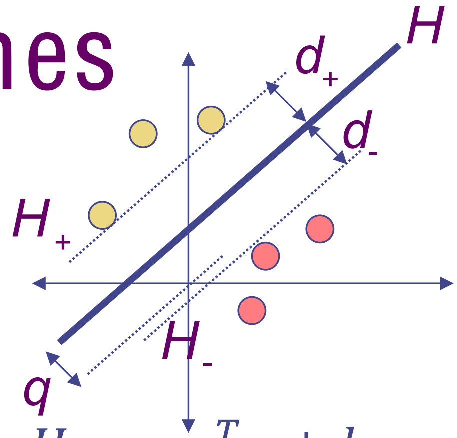


$$H_+ \rightarrow w^T x + b = +1$$

$$H \rightarrow w^T x + b = 0$$

$$H_- \rightarrow w^T x + b = -1$$

Support Vector Machines



- The constraints on the SVM for $R_{emp}(\theta)=0$ are thus:

$$w^T x_i + b \geq +1 \quad \forall y_i = +1$$
$$w^T x_i + b \leq -1 \quad \forall y_i = -1$$

- Or more simply: $y_i(w^T x_i + b) - 1 \geq 0$
- The margin of the SVM is: $m = d_+ + d_-$

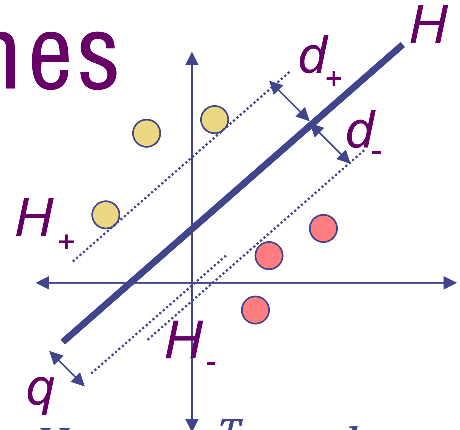
$$H_+ \rightarrow w^T x + b = +1$$
$$H \rightarrow w^T x + b = 0$$
$$H_- \rightarrow w^T x + b = -1$$

- Distance to origin:

$$H \rightarrow q = \frac{|b|}{\|w\|} \qquad H_+ \rightarrow q_+ = \frac{|b-1|}{\|w\|} \qquad H_- \rightarrow q_- = \frac{|-1-b|}{\|w\|}$$

- Therefore: $d_+ = d_- = 1/\|w\|$ and margin $m = 2/\|w\|$

Support Vector Machines



- The constraints on the SVM for $R_{emp}(\theta)=0$ are thus:

$$w^T x_i + b \geq +1 \quad \forall y_i = +1$$
$$w^T x_i + b \leq -1 \quad \forall y_i = -1$$

- Or more simply: $y_i(w^T x_i + b) - 1 \geq 0$
- The margin of the SVM is: $m = d_+ + d_-$
- Distance to origin:

$$H_+ \rightarrow w^T x + b = +1$$
$$H \rightarrow w^T x + b = 0$$
$$H_- \rightarrow w^T x + b = -1$$

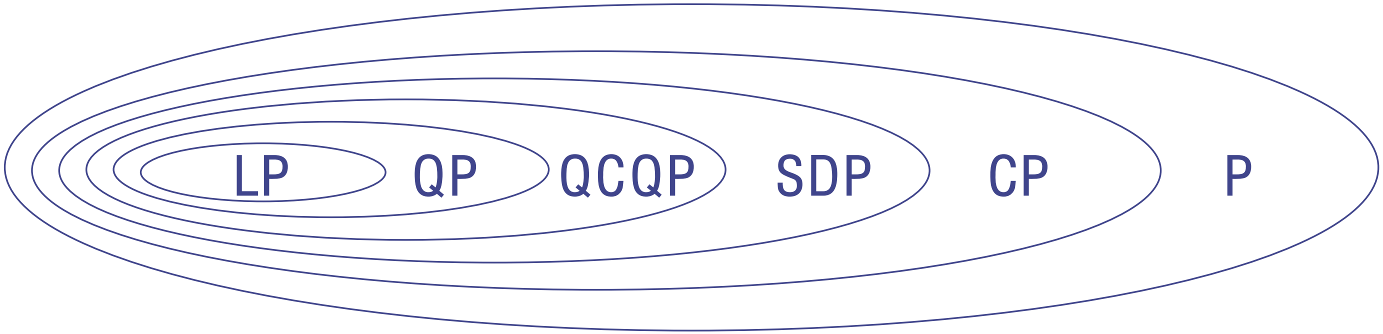
$$H \rightarrow q = \frac{|b|}{\|w\|} \qquad H_+ \rightarrow q_+ = \frac{|b-1|}{\|w\|} \qquad H_- \rightarrow q_- = \frac{|-1-b|}{\|w\|}$$

- Therefore: $d_+ = d_- = 1/\|w\|$ and margin $m = 2/\|w\|$
- Want to max margin, or equivalently minimize: $\|w\|$ or $\|w\|^2$
- SVM Problem: minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) - 1 \geq 0$
- This is a quadratic program!
- Python: cvxopt

Note: Quadratic Programming

- A hierarchy of optimization packages to use:

Linear Programming (LP)
<Quadratic Programming (QP)
<Quadratically Constrained Quadratic Programming
<Semidefinite Programming (SDP)
<Convex Programming (CP)
<Polynomial Time Algorithms (P)



Note: Quadratic Programming

- LP < QP < QCQP < SDP < Convex Programming

- LP $$\min_{\vec{x}} \vec{b}^T \vec{x} \quad \text{s.t. } \vec{c}^T \vec{x} \geq \alpha_i \forall i$$

- QP $$\min_{\vec{x}} \frac{1}{2} \vec{x}^T H \vec{x} + \vec{b}^T \vec{x} \quad \text{s.t. } \vec{c}^T \vec{x} \geq \alpha_i \forall i$$

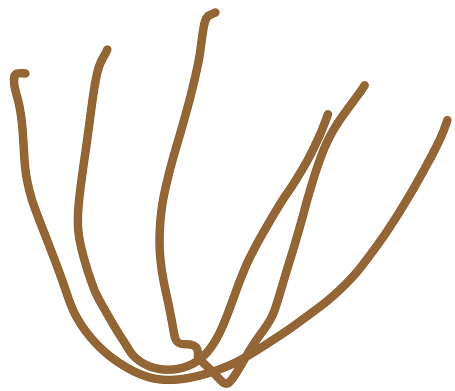
- QCQP

- SDP

- CP $$\min_{\vec{x}} f(\vec{x}) \quad \text{s.t. } g(\vec{x}) \geq \alpha$$

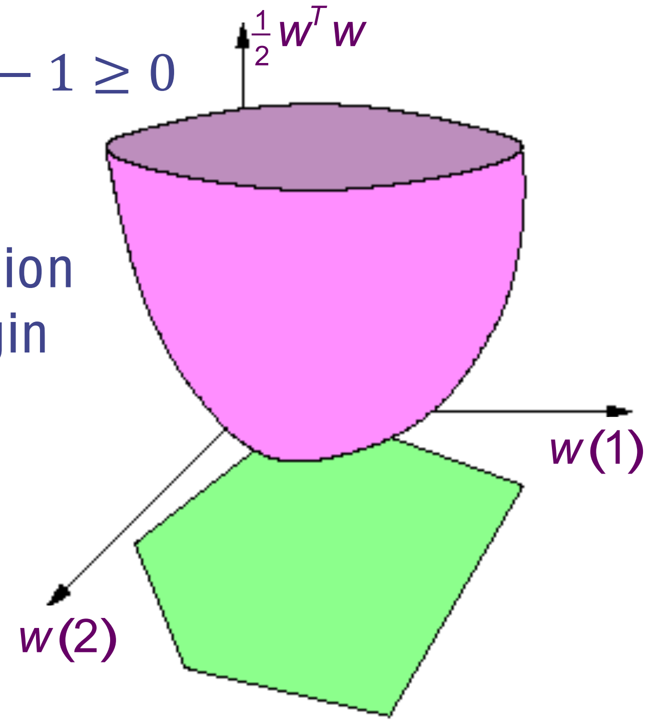
SVM Quadratic Program

- Find w
- Minimizing $\frac{1}{2}w^T w$
- Satisfying $\forall i: y_i(w^T x_i + b) - 1 \geq 0$



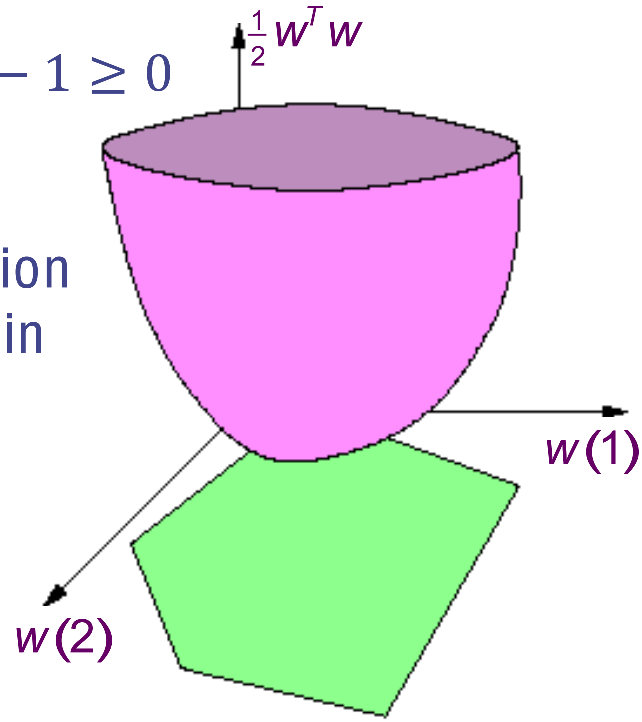
SVM Quadratic Program

- Each data point adds $y_i(w^T x_i + b) - 1 \geq 0$ linear inequality to QP
- Each point cuts a half plane of allowable SVMs, reduces green region
- The SVM is closest point to the origin that is still in the green region



SVM Quadratic Program

- Each data point adds $y_i(w^T x_i + b) - 1 \geq 0$ linear inequality to QP
- Each point cuts a half plane of allowable SVMs, reduces green region
- The SVM is closest point to the origin that is still in the green region
- The preceptron algorithm just puts us randomly in green region
- QP runs in cubic polynomial time
- There are D values in the w vector
- Needs $O(D^3)$ run time

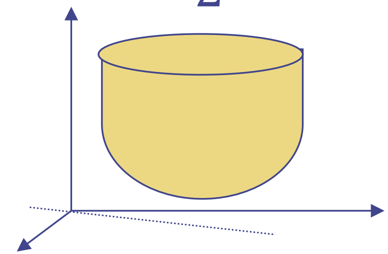


Note: Lagrange Multipliers

- How to minimize a function subject to equality constraints?

$$\min_{x_1,x_2} f(\vec{x}) = \min_{x_1,x_2} b_1 x_1 + b_2 x_2 + \frac{1}{2} H_{11}(x_1)^2 + H_{12} x_1 x_2 + \frac{1}{2} H_{22}(x_2)^2$$

$$= \min_{\vec{x}} \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T H \vec{x}$$



$$\nabla_x f = b + Hx = 0 \qquad x = H^{-1} b$$

- Only walk on $x_1 = 2x_2$

$$x_1 - 2x_2 = 0$$

$$\min_x \max_\lambda f - \lambda (x_1 - 2x_2)$$

Note: Lagrange Multipliers

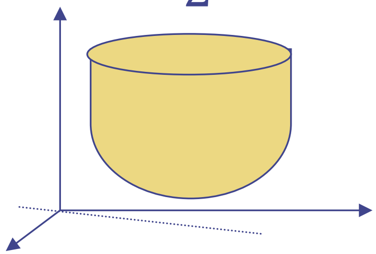
- How to minimize a function subject to equality constraints?

$$\min_{x_1,x_2} f(\vec{x}) = \min_{x_1,x_2} b_1 x_1 + b_2 x_2 + \frac{1}{2} H_{11} (x_1)^2 + H_{12} x_1 x_2 + \frac{1}{2} H_{22} (x_2)^2$$

$$= \min_{\vec{x}} \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T H \vec{x}$$

$$\Rightarrow \nabla_{\vec{x}} f = \vec{b} + H\vec{x} = 0$$

$$\Rightarrow \vec{x} = -H^{-1} \vec{b}$$



- Only walk on $x_1 = 2x_2$ or... $x_1 - 2x_2 = 0$ ...

Note: Lagrange Multipliers

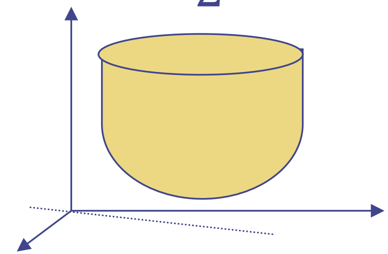
- How to minimize a function subject to equality constraints?

$$\min_{x_1,x_2} f(\vec{x}) = \min_{x_1,x_2} b_1 x_1 + b_2 x_2 + \frac{1}{2} H_{11}(x_1)^2 + H_{12} x_1 x_2 + \frac{1}{2} H_{22}(x_2)^2$$

$$= \min_{\vec{x}} \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T H \vec{x}$$

$$\Rightarrow \nabla_{\vec{x}} f = \vec{b} + H\vec{x} = 0$$

$$\Rightarrow \vec{x} = -H^{-1}\vec{b}$$



$x^T \begin{bmatrix} 1 \\ -2 \end{bmatrix} = 0$

- Only walk on x_1=2x_2 or… x_1-2x_2=0 …
- Use Lagrange Multipliers…
- λ blows up the minimization if we don't satisfy the constraint:

$$\min_{x_1,x_2} \max_{\lambda} f(\vec{x}) + \lambda(equality\ condition = 0)$$

$\lambda \cdot x^T \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

$\nabla_x = b + Hx + \lambda \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

Note: Lagrange Multipliers

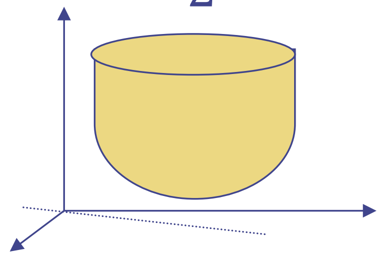
- How to minimize a function subject to equality constraints?

$$\min_{x_1,x_2} f(\vec{x}) = \min_{x_1,x_2} b_1 x_1 + b_2 x_2 + \frac{1}{2} H_{11}(x_1)^2 + H_{12} x_1 x_2 + \frac{1}{2} H_{22}(x_2)^2$$

$$= \min_{\vec{x}} \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T H \vec{x}$$

$$\Rightarrow \nabla_{\vec{x}} f = \vec{b} + H\vec{x} = 0$$

$$\Rightarrow \vec{x} = -H^{-1}\vec{b}$$

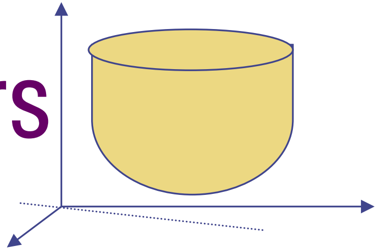


- Only walk on $x_1 = 2x_2$ or... $x_1 - 2x_2 = 0$ ...
- Use Lagrange Multipliers...
- λ blows up the minimization if we don't satisfy the constraint:

$$\min_{x_1,x_2} \max_{\lambda} f(\vec{x}) + \lambda(equality\ condition = 0)$$

$$= \min_{x_1,x_2} b_1 x_1 + b_2 x_2 + \frac{H_{11}(x_1)^2}{2} + H_{12} x_1 x_2 + \frac{H_{22}(x_2)^2}{2} + \lambda(x_1 - 2x_2)$$

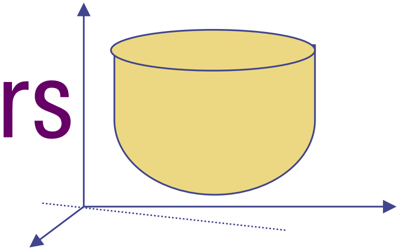
Note: Lagrange Multipliers



- Minimization with equality constraint:
  1) Add each constraint times an extra variable
     (a Lagrange multiplier λ, like an adversary variable)
  2) Take partials with respect to x and set to zero
  3) Plug in solution into constraint to find lambda

$$\min_{\vec{x}} \max_{\lambda} f(\vec{x}) + \lambda(equality\ condition = 0)$$

Note: Lagrange Multipliers



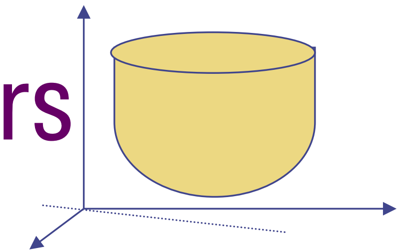
- Minimization with equality constraint:
  1) Add each constraint times an extra variable
     (a Lagrange multiplier λ, like an adversary variable)
  2) Take partials with respect to x and set to zero
  3) Plug in solution into constraint to find lambda

$$\min_{\vec{x}} \max_{\lambda} f(\vec{x}) + \lambda(equality\ condition = 0)$$

$$= \min_{\vec{x}} \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T H \vec{x} + \lambda(x_1 - 2x_2)$$

$$x_1 - 2x_2 = x^T \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

Note: Lagrange Multipliers



- Minimization with equality constraint:
  1) Add each constraint times an extra variable
     (a Lagrange multiplier λ, like an adversary variable)
  2) Take partials with respect to x and set to zero
  3) Plug in solution into constraint to find lambda

$$\min_{\vec{x}} \max_{\lambda} f(\vec{x}) + \lambda(equality\ condition = 0)$$

$$x_1 - 2x_2 = x^T \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

$$= \min_{\vec{x}} \vec{b}^T \vec{x} + \frac{1}{2} \vec{x}^T H \vec{x} + \lambda(x_1 - 2x_2)$$

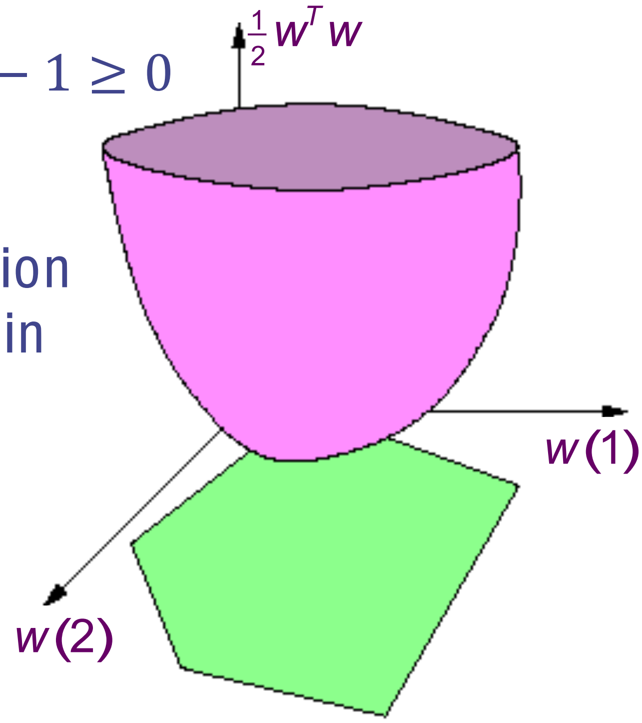
$$\Rightarrow \nabla_{\vec{x}} f = \vec{b} + H\vec{x} + \lambda \begin{bmatrix} 1 \\ -2 \end{bmatrix} = 0$$

$$\Rightarrow \vec{x} = -H^{-1}\lambda \begin{bmatrix} 1 \\ -2 \end{bmatrix} - H^{-1}\vec{b}$$

$$\Rightarrow \left(-H^{-1}\lambda \begin{bmatrix} 1 \\ -2 \end{bmatrix} - H^{-1}\vec{b}\right)^T \begin{bmatrix} 1 \\ -2 \end{bmatrix} = 0 \Rightarrow \lambda = \frac{\vec{b}^T H^{-1} \begin{bmatrix} 1 \\ -2 \end{bmatrix}}{\begin{bmatrix} 1 \\ -2 \end{bmatrix}^T H^{-1} \begin{bmatrix} 1 \\ -2 \end{bmatrix}}$$

SVM Quadratic Program

- Each data point adds $y_i(w^T x_i + b) - 1 \geq 0$ linear inequality to QP
- Each point cuts a half plane of allowable SVMs, reduces green region
- The SVM is closest point to the origin that is still in the green region
- The preceptron algorithm just puts us randomly in green region
- QP runs in cubic polynomial time
- There are D values in the w vector
- Needs $O(D^3)$ run time
- But, there is a DUAL SVM in $O(N^3)$!



SVM in Dual Form

- We can also solve the problem via convex duality
- Primal SVM problem L_P:

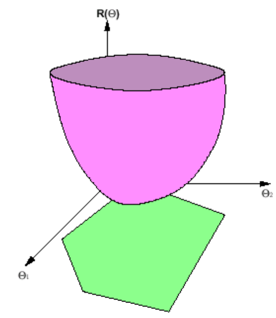
minimize $\frac{1}{2}||w||^2$ subject to $y_i(w^T x_i + b) - 1 \geq 0$

- This is a convex program
- With Lagrange α:

$L_P =$

$$\frac{1}{2}||w||^2 - \sum_i \alpha_i \left[y_i(w^T x_i + b) - 1 \right]$$

$\alpha_i \geq 0$



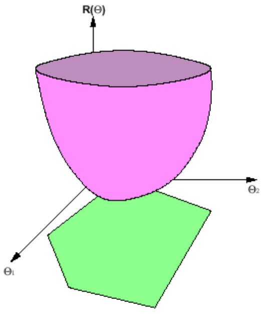
SVM in Dual Form

- We can also solve the problem via convex duality
- Primal SVM problem L_P:

minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) - 1 \geq 0$

- This is a convex program
- With Lagrange α:

$$L_P = \min_{w,b} \max_{\alpha \geq 0} \underline{\frac{1}{2}\|w\|^2} - \sum_i \alpha_i \left(y_i (w^T x_i + b) - 1 \right)$$



- Take derivatives with:

$$\frac{\partial}{\partial w} L_P = w - \sum_i \alpha_i y_i x_i = 0$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0$$

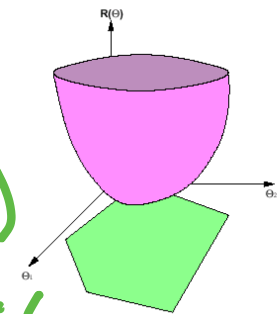
SVM in Dual Form

- We can also solve the problem via convex duality
- Primal SVM problem L_P:

minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) - 1 \geq 0$

- This is a convex program
- With Lagrange α:

$$L_P = \min_{w,b} \max_{\alpha \geq 0} \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left(y_i (w^T x_i + b) - 1\right)$$



$\sum \alpha_i y_i x_i \quad \sum \alpha_j y_j x_j$

- Take derivatives with:

$$\frac{\partial}{\partial w} L_P = w - \sum_i \alpha_i y_i x_i = 0 \rightarrow w = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0$$

0

Plug back in:

$-\frac{1}{2}\left(\sum_i \alpha_i y_i x_i\right)^2 \qquad -b \sum_i \alpha_i y_i + \sum \alpha_i$

$\left(-1 + \frac{1}{2}\right)$

SVM in Dual Form

- We can also solve the problem via convex duality
- Primal SVM problem L_P:

minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) - 1 \geq 0$

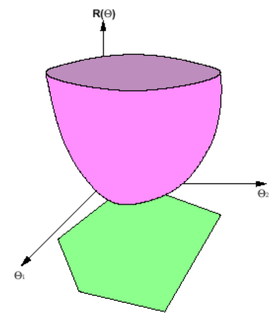
- This is a convex program
- Try taking derivatives with Lagrange α:

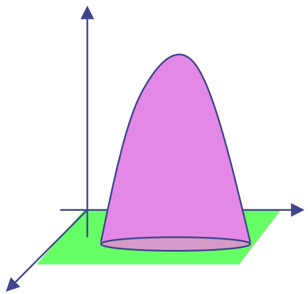
$$L_P = \min_{w,b} \max_{\alpha \geq 0} \frac{1}{2}\|w\|^2 - \sum_i \alpha_i (y_i(w^T x_i + b) - 1)$$

$$\frac{\partial}{\partial w} L_P = w - \sum_i \alpha_i y_i x_i = 0 \rightarrow w = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0$$

- Plug back in, dual:



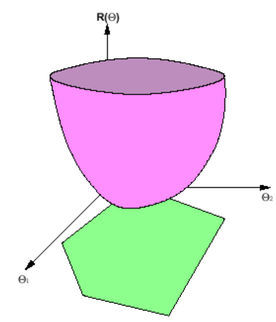


SVM in Dual Form

- We can also solve the problem via convex duality
- Primal SVM problem L_P:

minimize $\frac{1}{2}\|w\|^2$ subject to $y_i(w^T x_i + b) - 1 \geq 0$

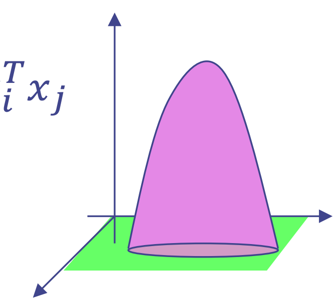
- This is a convex program
- Try taking derivatives with Lagrange α:



$$L_P = \min_{w,b} \max_{\alpha \geq 0} \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left(y_i\left(w^T x_i + b\right) - 1\right)$$

$$\frac{\partial}{\partial w} L_P = w - \sum_i \alpha_i y_i x_i = 0 \rightarrow w = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0$$



- Plug back in, dual: $L_D = max \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j$

Subject to constraints: $\sum_i \alpha_i y_i = 0$, $\alpha_i \geq 0$

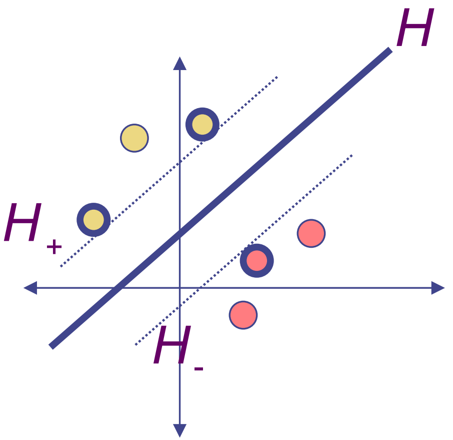
Also just QP, but in N variables !

SVM Dual Solution Properties

- We have dual convex program:

$L_D = max \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j$ subject to $\sum_i \alpha_i y_i = 0,\ \alpha_i \geq 0$

- Solve for N alphas (one per data point) instead of D w's
- Still convex (qp) so unique solution, gives alphas
- Alphas can be used to get w: $w = \sum_i \alpha_i y_i x_i$

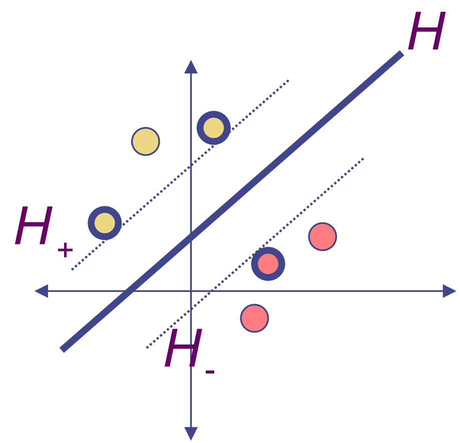


SVM Dual Solution Properties

- We have dual convex program:

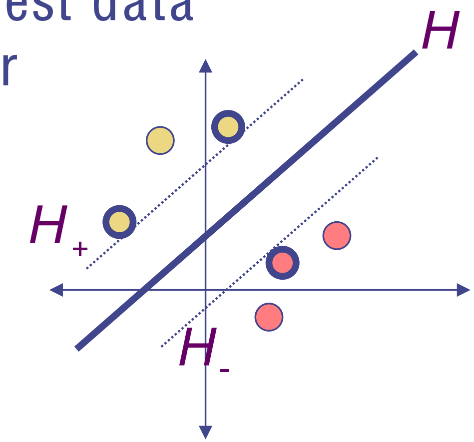
$L_D = max \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j$ subject to $\sum_i \alpha_i y_i = 0,\ \alpha_i \geq 0$

- Solve for N alphas (one per data point) instead of D w's
- Still convex (qp) so unique solution, gives alphas
- Alphas can be used to get w: $w = \sum_i \alpha_i y_i x_i$
- Support Vectors: have non-zero alphas shown with thicker circles, all live on the margin: $w^T x_i + b = \pm 1$
- Solution is sparse, most alphas=0 these are *non-support vectors* SVM ignores them if they move (without crossing margin) or if they are deleted, SVM doesn't change (stays robust)



SVM Dual Solution Sparsity

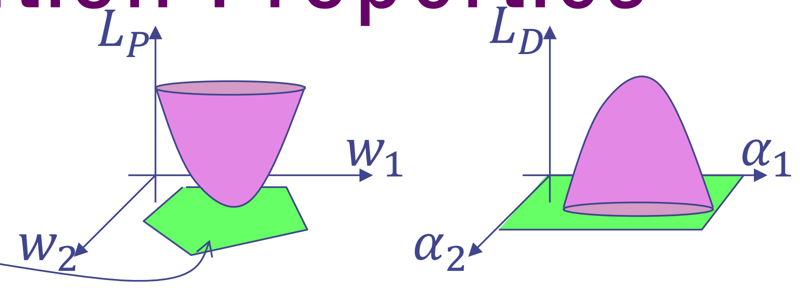
- Support Vectors: have non-zero alphas shown with thicker circles, all live on the margin: $w^T x_i + b = \pm 1$
- Solution is sparse, most alphas=0 these are *non-support vectors;* SVM ignores them if they move (without crossing margin) or are deleted, SVM doesn't change (stays robust)
- Importance:
  - Means SVM only uses some of training data to learn
  - Helps improve ability to generalize to test data
  - Classifier can be computationally faster



SVM Dual Solution Properties

- Primal & Dual Illustration:

$$w^T \boldsymbol{x}_i + b \geq \pm 1$$

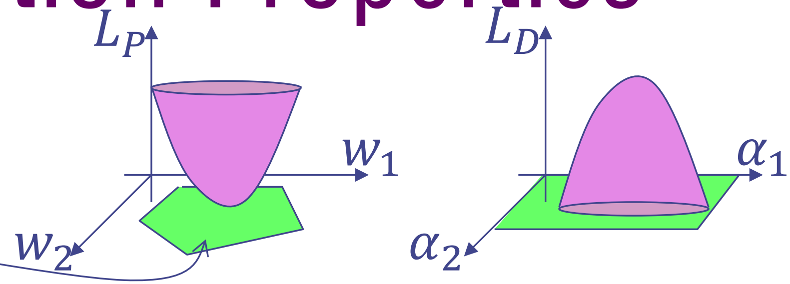


- Recall we could get w from alphas: $w = \sum_i \alpha_i y_i \boldsymbol{x}_i$
- Or use as is: $f(\boldsymbol{x}) = sign\,(\boldsymbol{x}^T w + b) = sign(\sum_i \alpha_i y_i \boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{x}_i + b)$

SVM Dual Solution Properties

- Primal & Dual Illustration:

  $w^T \boldsymbol{x}_i + b \geq \pm 1$



- Recall we could get w from alphas: $w = \sum_i \alpha_i y_i \boldsymbol{x}_i$
- Or use as is: $f(\boldsymbol{x}) = sign\,(\boldsymbol{x}^T w + b) = sign(\sum_i \alpha_i y_i \boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{x}_i + b)$
- Karush-Kuhn-Tucker Conditions (KKT): solve value of b
  on margin (for nonzero alphas) have: $w^T \boldsymbol{x}_i + b = y_i$
  using known w, compute b for each support vector
  $\widetilde{b_i} = y_i - w^T \boldsymbol{x}_i$ then $b = average(\widetilde{b_i})$

Summary

- SVM maximizes gap
- Solved by QP in #dimensions
- Dual: QP in #datapoints
- Leans on a few support vectors