Support Vector Machines

Supervised Learning

Labeled Data

quail

apple

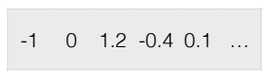
apple

corn

corn

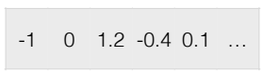
Features

quail



apple

apple



corn

x

y

Model Class

Consider classifiers of the form $y = f(x; w)$

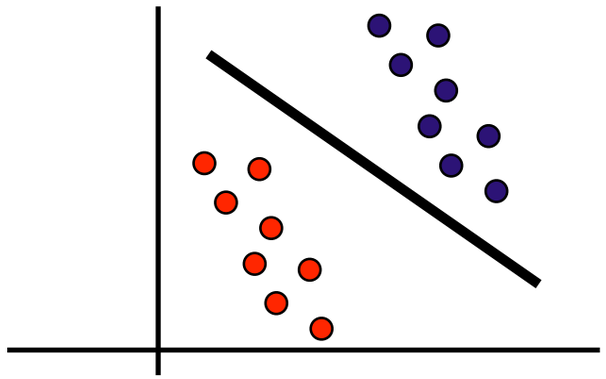
Optimization!

Learning

Find w that works well on the training data

Linear Classifiers

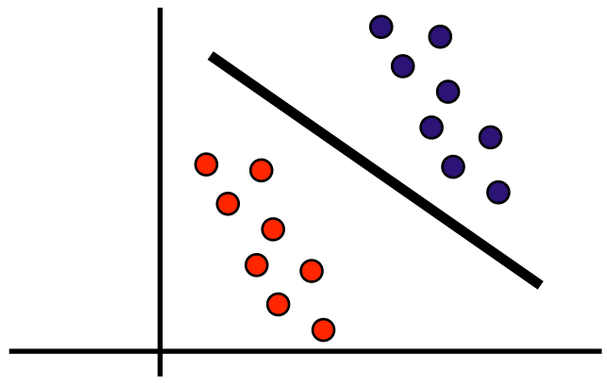
- A simple and effective family of classifiers



$$y = sign\left[\boldsymbol{w} \cdot \boldsymbol{x} + b\right]$$

- The training problem:
  - Given a set of n training points (x_i, y_i)
  - Find $\boldsymbol{w}, b$ for the “best fitting” classifier

Training Linear Classifiers



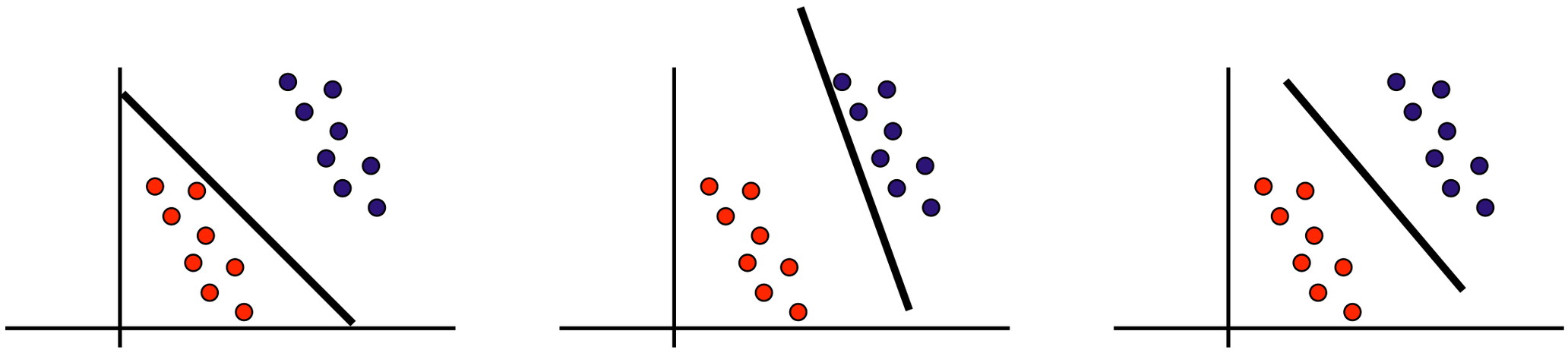
$$y = sign\left[\boldsymbol{w} \cdot \boldsymbol{x} + b\right]$$

- How do we find it?

- If there exists a classifier with zero training error we can find one with the perceptron algorithm

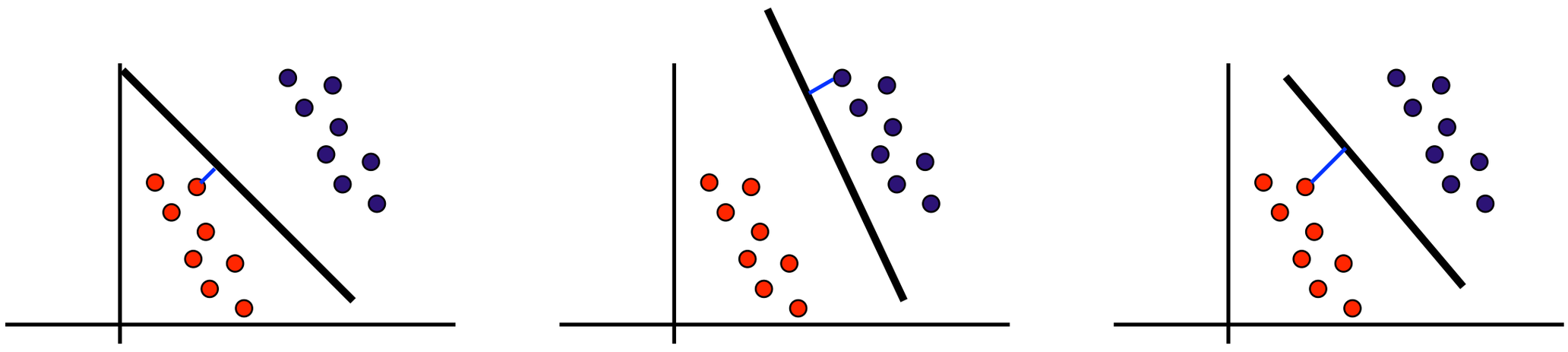
Many Possible Solutions

- If there exists one solution, there exist many.
- Which one should we choose?
- Intuitively: one that's farther away from the points.



Maximum Margin Classifier

- For every point $\boldsymbol{x}_i$ denote $d(\boldsymbol{x}_i, \boldsymbol{w}, b)$ its distance from the hyperplane.
- Margin of a classifier: the shortest distance to the hyperplane: $\min_i d(\boldsymbol{x}_i, \boldsymbol{w}, b)$
- Goal: find classifier that maximizes $\min_i d(\boldsymbol{x}_i, \boldsymbol{w}, b)$



ML and Optimization

- We have an optimization problem
- Namely we want to find a set of parameters that will maximize some objective function (the margin) subject to some constraints (classifying correctly)
- Need a toolbox for solving such problems
- In what follows we provide an overview

Unconstrained Optimization

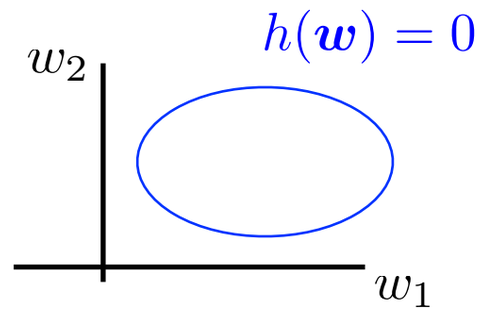
- Use **w** to denote optimization variables
- For example: $\min_{w_1, w_2} (w_1 - 2w_2)^2$
- Generally: $\min_{\boldsymbol{w}} f(\boldsymbol{w})$
- Solve by:
  - Find all **w** such that $\dfrac{\partial f(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0$
  - These stationary points are the candidates for the global minimum (and asymptotes)

Constrained Minimization

- Suppose we are only interested in variables that satisfy $h(\boldsymbol{w}) = 0$

- The optimization problem is:

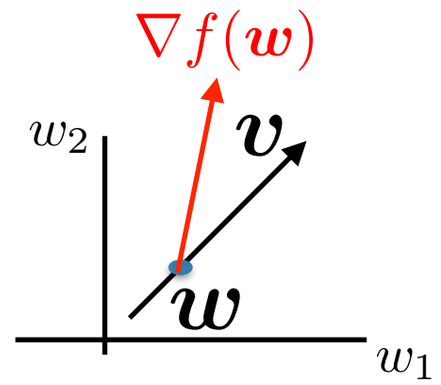
$$\begin{array}{ll} \min & f(\boldsymbol{w}) \\ \text{s.}t. & h(\boldsymbol{w}) = 0 \end{array}$$



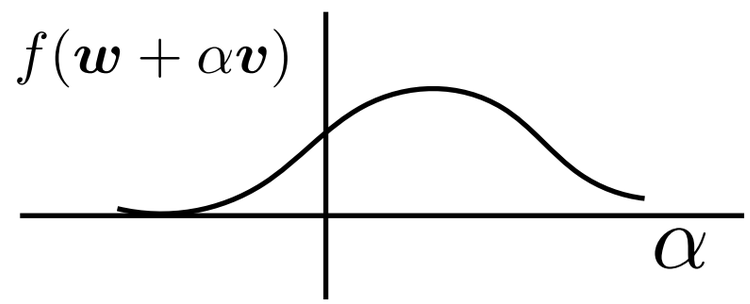
- The zero gradient point may not satisfy the constraint.

Directional Derivative

- Given function f(**w**) and direction **v**.

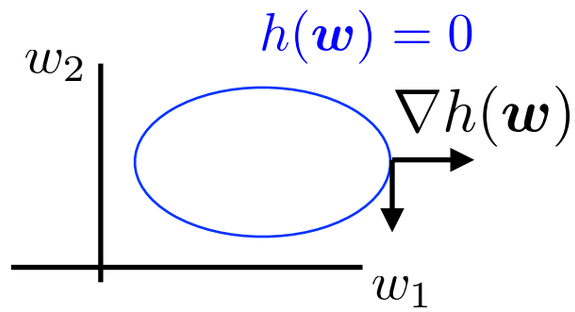


- What happens if we make a small change in direction **v** (where $|\mathbf{v}|_2=1$?)



- It is $\nabla f(\boldsymbol{w}) \cdot \boldsymbol{v}$

- The direction along the curve h(**w**)=0 has zero directional derivative.

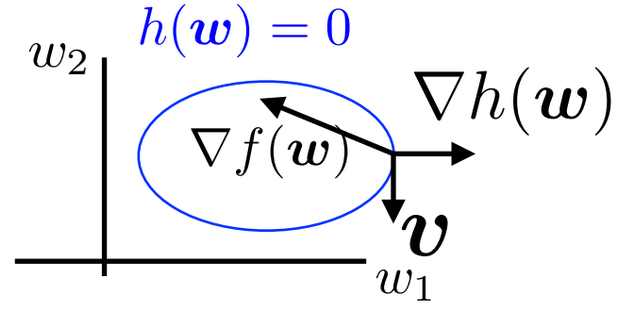


- Thus the orthogonal is the gradient

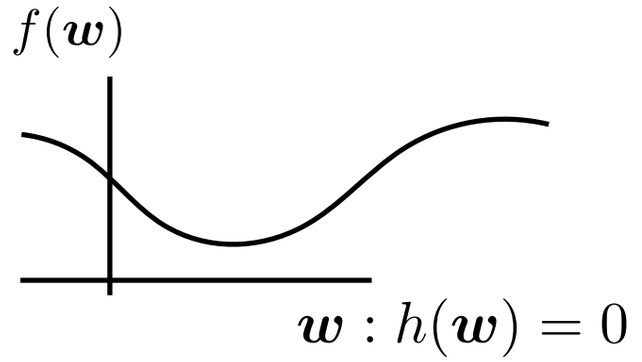
Constrained Minimization

- The optimization problem is:

$$\begin{array}{ll} \min & f(\boldsymbol{w}) \\ \text{s.}t. & h(\boldsymbol{w}) = 0 \end{array}$$



- Consider f along h
- Vector of movement along curve **v is** orthogonal to $\nabla h(\boldsymbol{w})$
- Gradient along curve $\nabla f(\boldsymbol{w}) \cdot \boldsymbol{v}$
- Is zero iff $\nabla f(\boldsymbol{w}) = \lambda \nabla h(\boldsymbol{w})$



Lagrange Multiplier

- The optimum points should satisfy:
  1. $\nabla f(\boldsymbol{w}) = \lambda \nabla h(\boldsymbol{w})$ For some λ
  2. $h(\boldsymbol{w}) = 0$ Constraint satisfied

- Alternative formulation. Define Lagrangian:

$$\mathcal{L}(\boldsymbol{w}, \lambda) = f(\boldsymbol{w}) + \lambda h(\boldsymbol{w})$$

- The optimum should satisfy:
  1. $\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \lambda) = 0$
  2. $\nabla_{\lambda} \mathcal{L}(\boldsymbol{w}, \lambda) = 0$

Example

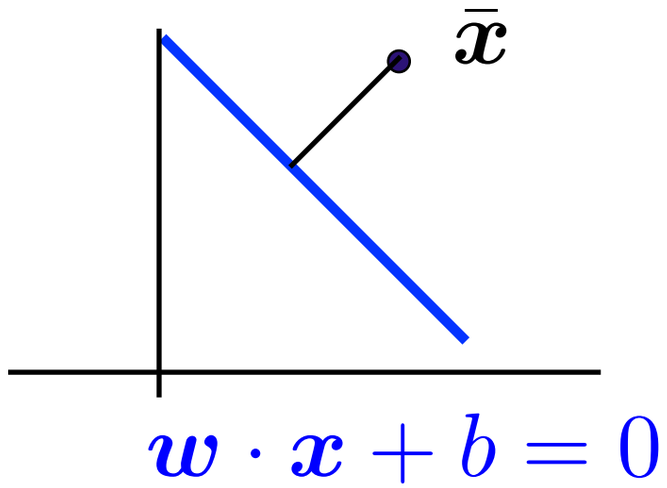
- What is the distance between hyperplane and point?

$$\min_{\boldsymbol{x}:\boldsymbol{w}\cdot\boldsymbol{x}+b=0} 0.5\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2^2$$

$$\mathcal{L}(\boldsymbol{x},\lambda) = 0.5\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2^2 + \lambda\left(\boldsymbol{w}\cdot\boldsymbol{x}+b\right)$$

$$\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x},\lambda) = (\boldsymbol{x}-\bar{\boldsymbol{x}}) + \lambda\boldsymbol{w} = 0$$

$$\boldsymbol{x} = \bar{\boldsymbol{x}} - \lambda\boldsymbol{w}$$



- Use primal feasibility to solve for λ

$$(\bar{\boldsymbol{x}}-\lambda\boldsymbol{w})\cdot\boldsymbol{w}+b=0 \longrightarrow \lambda = \frac{\bar{\boldsymbol{x}}\cdot\boldsymbol{w}+b}{\|\boldsymbol{w}\|}$$

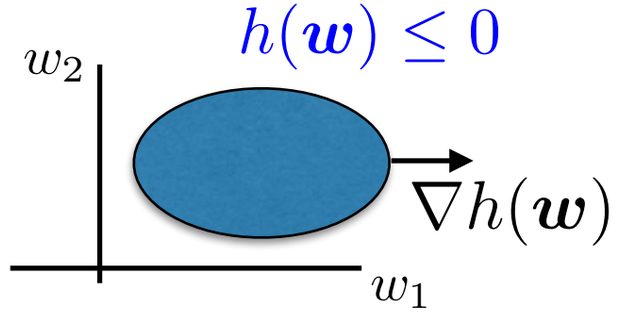
$$\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2 = \frac{|\bar{\boldsymbol{x}}\cdot\boldsymbol{w}+b|}{\|\boldsymbol{w}\|}$$

Multiple Constraints

- Solve: $$\begin{aligned} \min \quad & f(\boldsymbol{w}) \\ \text{s.}t. \quad & h_i(\boldsymbol{w}) = 0 \quad \forall i = 1, \ldots, p \end{aligned}$$
- Introduce multiplier per constraint $\lambda_1, \cdots, \lambda_p$
- Lagrangian: $\mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = f(\boldsymbol{w}) + \sum_i \lambda_i h_i(\boldsymbol{w})$
- Optimality conditions:
  1. $\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \lambda) = 0$
  2. $\nabla_{\lambda_i} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}) = 0$
- May be several such points. Need to check which one is the global optimum.

Inequality Constraints

- Solve: $\begin{aligned} \min \quad & f(\boldsymbol{w}) \\ \text{s.}t. \quad & h(\boldsymbol{w}) \leq 0 \end{aligned}$



- Optimality conditions:

1. $h(\boldsymbol{w}) \leq 0$ Constraint satisfied

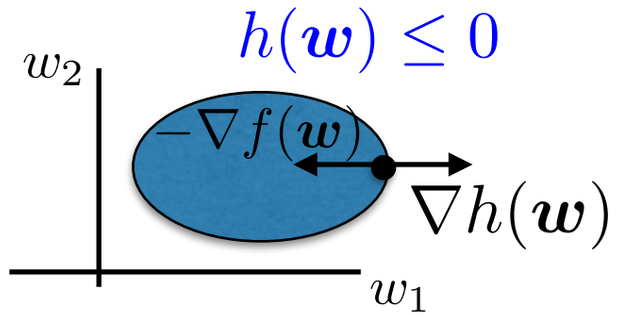
- When $h(\boldsymbol{w}) = 0$ we are "stuck" if the only directions that decrease f take us outside the constraints. Namely:

2a. $\nabla f(\boldsymbol{w}) = -\alpha \nabla h(\boldsymbol{w})$ For some $\alpha \geq 0$

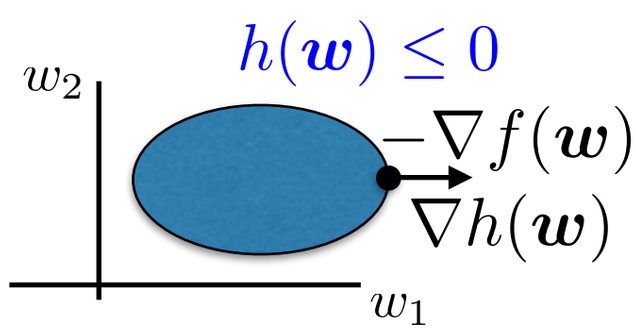
- When $h(\boldsymbol{w}) < 0$ we need:

2b. $\nabla f(\boldsymbol{w}) = 0$

Progress possible



Stuck



Complementary Slackness

$$h(\boldsymbol{w}) = 0 \longrightarrow \nabla f(\boldsymbol{w}) = -\alpha \nabla h(\boldsymbol{w})$$

$$h(\boldsymbol{w}) < 0 \longrightarrow \nabla f(\boldsymbol{w}) = 0$$

- Summarize as:

$$\nabla f(\boldsymbol{w}) = -\alpha \nabla h(\boldsymbol{w})$$

$$\alpha h(\boldsymbol{w}) = 0$$

$$\alpha \geq 0, h(\boldsymbol{w}) \leq 0$$

- Called the Karush Kuhn Tucker (KKT) conditions. Always necessary.

- Sufficient for convex optimization.

Lagrange Multipliers

- Consider the general problem:

$$\begin{aligned} \min \quad & f(\boldsymbol{w}) \\ \text{s.}t. \quad & h_i(\boldsymbol{w}) = 0 \;\; \forall i = 1, \ldots, p \\ & g_i(\boldsymbol{w}) \leq 0 \;\; \forall i = 1, \ldots, m \end{aligned}$$

- Define the Lagrangian:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = f(\boldsymbol{w}) + \sum_i \lambda_i h_i(\boldsymbol{w}) + \sum_i \alpha_i g_i(\boldsymbol{w})$$

- Optimum must satisfy:

$$\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = 0$$

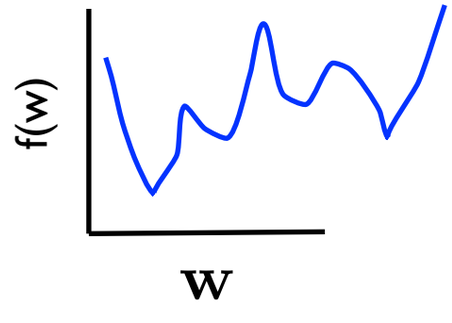
$$\alpha_i \geq 0, g_i(\boldsymbol{w}) \leq 0, h_i(\boldsymbol{w}) = 0$$

$$\alpha_i g_i(\boldsymbol{x}) = 0 \;\; \forall i$$

Typically easy if someone hands us $\boldsymbol{\alpha}, \boldsymbol{\lambda}$*!*

Convex Optimization

- General optimization problem may have many local minima/maxima and saddle points.

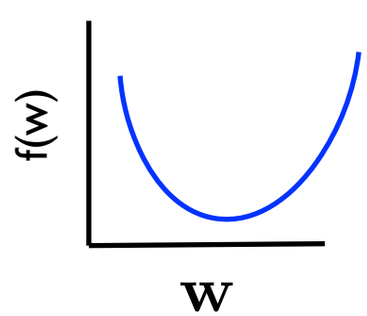


- Makes minimization hard (e.g., exponential in dimension).

- Convex optimization problems are a “nice” subclass. Require:

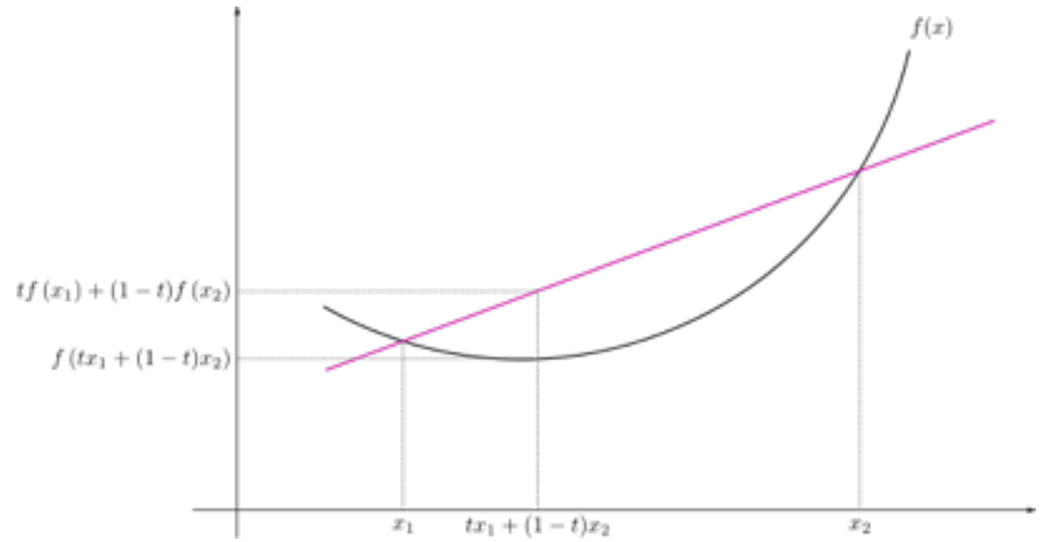
  - Convex f(w),g(x)

  - Linear h(x)



Convex Optimization

- Convex function if:
  - Value on line is less than linear function.
  - Non-negative second derivative (or Hessian)



- Examples:

$$f(\boldsymbol{w}) = \boldsymbol{w} \cdot \boldsymbol{x}$$

$$f(\boldsymbol{w}) = \max\left[\boldsymbol{w} \cdot \boldsymbol{x}, 0\right]$$

$$f(\boldsymbol{w}) = \boldsymbol{w}^T A \boldsymbol{w} \quad A \succeq 0$$

Convex Optimization

- Nice things:
  - No local optima
  - KKT conditions are sufficient for global optimality.
  - Multipliers can be solved via dual.

Convex Duality

- For every convex problem, we can define a **dual** problem that has the same value

- Optimization is over the Lagrange multipliers.

- Solution to dual implies solution to primal via KKT

- Dual might be easier to solve.

Convex Duality

- Recall the Lagrangian:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = f(\boldsymbol{w}) + \sum_i \lambda_i h_i(\boldsymbol{w}) + \sum_i \alpha_i g_i(\boldsymbol{w})$$

- Then:

$$\begin{array}{lll} \min & f(\boldsymbol{w}) & = \min_{\boldsymbol{w}} \max_{\boldsymbol{\lambda}, \boldsymbol{\alpha} \geq 0} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) \\ \text{s.}t. & h_i(\boldsymbol{w}) = 0 & \\ & g_i(\boldsymbol{w}) \leq 0 & \end{array} \qquad \text{Why?}$$

- Replacing min and max gives:

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{\lambda}, \boldsymbol{\alpha} \geq 0} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) \leq \max_{\boldsymbol{\lambda}, \boldsymbol{\alpha} \geq 0} \min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha})$$

- In the convex case it is an equality

Convex Duality

- Define: $g(\boldsymbol{\lambda}, \boldsymbol{\alpha}) = \min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha})$
- Dual problem: $\max_{\boldsymbol{\lambda}, \boldsymbol{\alpha} \geq 0} g(\boldsymbol{\lambda}, \boldsymbol{\alpha})$
- Has same value as primal problem.
- The resulting λ, α are optimal. You can recover the "primal" variables **w** via KKT. This is often easy.

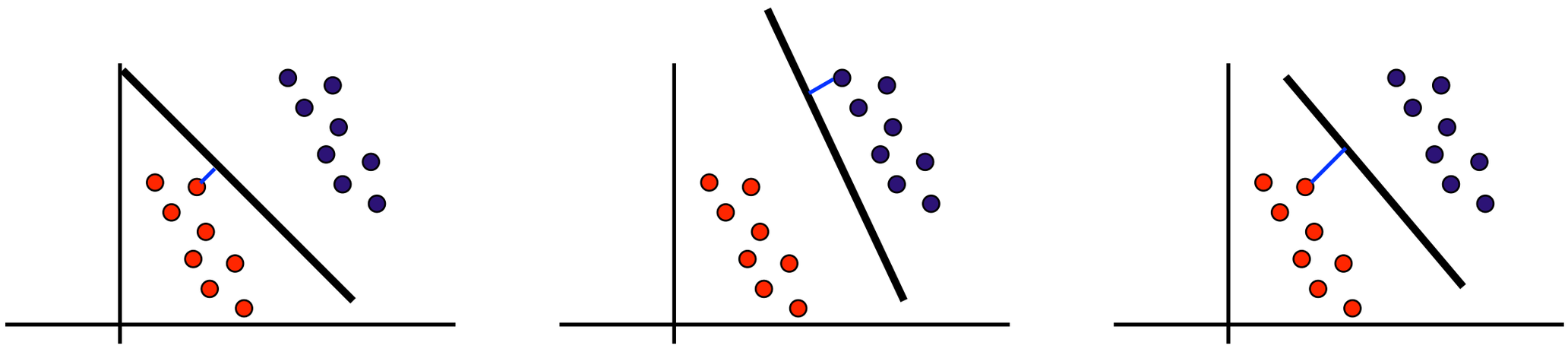
$$\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = 0$$

$$\alpha_i \geq 0, g_i(\boldsymbol{w}) \leq 0, h_i(\boldsymbol{w}) = 0$$

$$\alpha_i g_i(\boldsymbol{x}) = 0 \quad \forall i$$

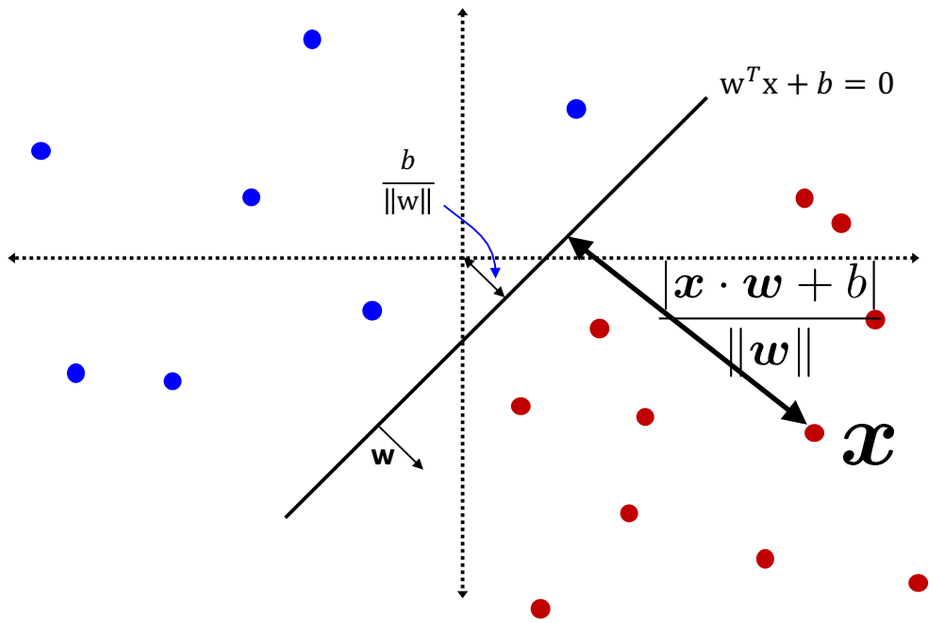
Maximum Margin Classifier

- For every point $\boldsymbol{x}_i$ denote $d(\boldsymbol{x}_i, \boldsymbol{w}, b)$ its distance from the hyperplane.
- Margin of a classifier: the shortest distance to the hyperplane: $\min_i d(\boldsymbol{x}_i, \boldsymbol{w}, b)$
- Goal: find classifier that maximizes $\min_i d(\boldsymbol{x}_i, \boldsymbol{w}, b)$



Geometry of Linear Classifiers

- **w** is the orthogonal direction to the hyperplane.
- Proof: if $\boldsymbol{x}_1, \boldsymbol{x}_2$ on hyperplane then $\boldsymbol{w} \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0$
- Distance from origin to hyperplane is $\dfrac{|b|}{\|\boldsymbol{w}\|}$

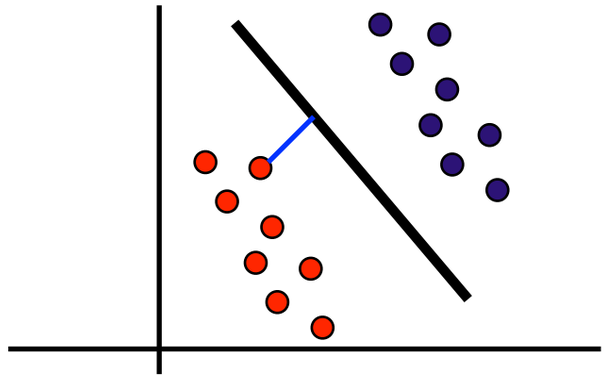


$$y = sign\left[\boldsymbol{w} \cdot \boldsymbol{x} + b\right]$$

Max Margin Hyperplane

- Find a hyperplane that maximizes the minimum distance
- Solve:

$$\begin{aligned} &\max_{\boldsymbol{w}} && \frac{1}{\|\boldsymbol{w}\|} \min_i |\boldsymbol{w} \cdot \boldsymbol{x}_i + b| \\ &\text{s.}t. && y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right) \geq 0 \end{aligned}$$



- Any solution $(\boldsymbol{w}, b)$ can be rescaled to $(c\boldsymbol{w}, cb)$ and not affect the objective or constraints.
- We can rescale such that $\min_i |\boldsymbol{w} \cdot \boldsymbol{x}_i + b| = 1$

$$\begin{aligned} &\max_{\boldsymbol{w}} && \|\boldsymbol{w}\|^{-1} \\ &\text{s.}t. && y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right) \geq 0 \\ & && \min_i |\boldsymbol{w} \cdot \boldsymbol{x}_i + b| = 1 \end{aligned}$$

Max Margin Hyperplane

$$\begin{array}{ll} \max_{\boldsymbol{w}} & \|\boldsymbol{w}\|^{-1} \\ \text{s.}t. & y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right) \geq 0 \\ & \min_i \left|\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right| = 1 \end{array}$$

- Equivalently:

$$\begin{array}{ll} \max_{\boldsymbol{w}} & \|\boldsymbol{w}\|^{-1} \\ \text{s.}t. & \min_i y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right) = 1 \end{array}$$

- We can relax to an inequality (why?): $\min_i y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right) \geq 1$

$$\begin{array}{ll} \max_{\boldsymbol{w}} & \|\boldsymbol{w}\|^{-1} \\ \text{s.}t. & y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right) \geq 1 \end{array} \longrightarrow \boxed{\begin{array}{ll} \min_{\boldsymbol{w}} & \|\boldsymbol{w}\|^{2} \\ \text{s.}t. & y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right) \geq 1 \end{array}}$$

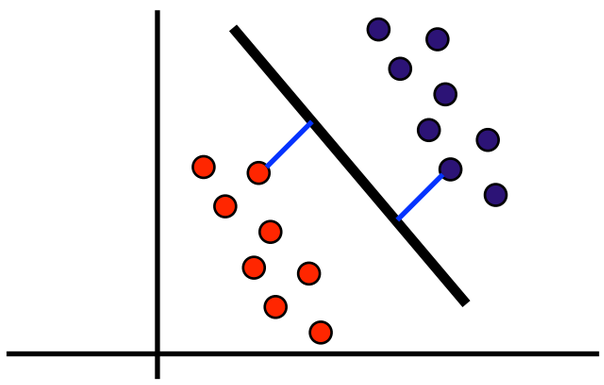
Support Vector Machines (SVM)

- The SVM classifier is the solution to:

$$\begin{array}{ll} \min_{\boldsymbol{w}} & 0.5\|\boldsymbol{w}\|^2 \\ \text{s.}t. & y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i+b\right)\geq 1 \end{array}$$

Factor 0.5 doesn't affect the optimum

- The $\boldsymbol{x}_i$ where this is an equality are "support vectors"



- It is a convex optimization problem. Called a convex quadratic program (quad. objective and linear constraints)

SVM History

- Initial version by Vapnik and Chervonenkis (63)
- Non linear version by Boser, Guyon, Vapnik (92)
- Much work on generalization theory since (by Bartlett, Shawe Taylor, Mendelson, Schoelkopf, Smola, and others).
- Many variants for regression, unsupervised learning etc.

Solving SVM

- The SVM classifier is the solution to:

$$\begin{array}{ll} \min_{\boldsymbol{w}} & 0.5\|\boldsymbol{w}\|^2 \\ \text{s.}t. & y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i+b\right)\geq 1 \end{array}$$

- You can plug this into a solver and get **w**, b

- Lets use Lagrangian to understand solution.

$$\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\alpha})=0.5\|\boldsymbol{w}\|^2+\sum_i \alpha_i\left[1-y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i+b\right)\right]$$

$$\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\alpha})=\boldsymbol{w}-\sum_i \alpha_i y_i \boldsymbol{x}_i=0 \longrightarrow \boldsymbol{w}=\sum_i \alpha_i y_i \boldsymbol{x}_i$$

$$\nabla_b\mathcal{L}(\boldsymbol{w},b,\boldsymbol{\alpha})=-\sum_i \alpha_i y_i=0 \longrightarrow \sum_i \alpha_i y_i=0$$

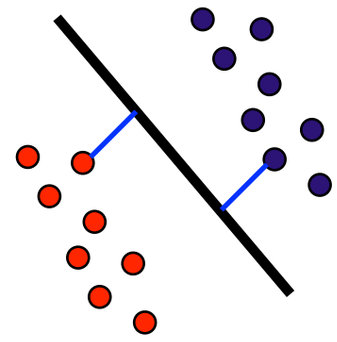
The Representer Theorem

- The optimal weight is a weight combination of the data points

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$$

- This will be very important!

- When is $\alpha_i = 0$? (recall KKT)
- Whenever $\boldsymbol{w} \cdot \boldsymbol{x}_i + b < 1$
- $\alpha_i > 0$ only when $\boldsymbol{w} \cdot \boldsymbol{x}_i + b = 1$



- Optimal weight is a combination only of support vectors!

Deriving via Dual

- How do we find α_i and then b?

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$$

- Use the dual! $g(\boldsymbol{\alpha}) = \min_{\boldsymbol{w},b} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})$

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = 0.5\|\boldsymbol{w}\|^2 + \sum_i \alpha_i \left[1 - y_i \left(\boldsymbol{w} \cdot \boldsymbol{x}_i + b\right)\right]$$

- Constrain $\sum_i \alpha_i y_i = 0$ because otherwise $g(\boldsymbol{\alpha}) = -\infty$
- We know the minimizing **w**. Plug into Lagrangian.

$$g(\boldsymbol{\alpha}) = 0.5\|\sum_i \alpha_i y_i \boldsymbol{x}_i\|_2^2 - \sum_i \alpha_i \left[y_i \left(\left(\sum_j \alpha_j y_j \boldsymbol{x}_j\right) \cdot \boldsymbol{x}_i + b\right) - 1\right]$$

$$= \sum_i \alpha_i - 0.5 \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j$$

The SVM Dual

- The dual problem is:

$$\begin{array}{ll} \max & \sum_i \alpha_i - 0.5 \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \\ \text{s.}t. & \alpha_i \geq 0, \sum_i \alpha_i y_i = 0 \end{array}$$

- Number of variables and constraints is number of training points.
- Also a convex quadratic program (why?)
- Obtaining the primal **w**:

$$\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i$$

Finding b

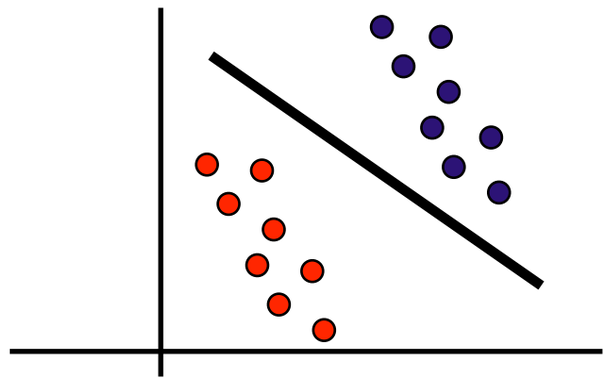
- Recall from KKT that support vectors ($\alpha_i > 0$) satisfy:

$$\boldsymbol{w} \cdot \boldsymbol{x}_i + b = 1$$

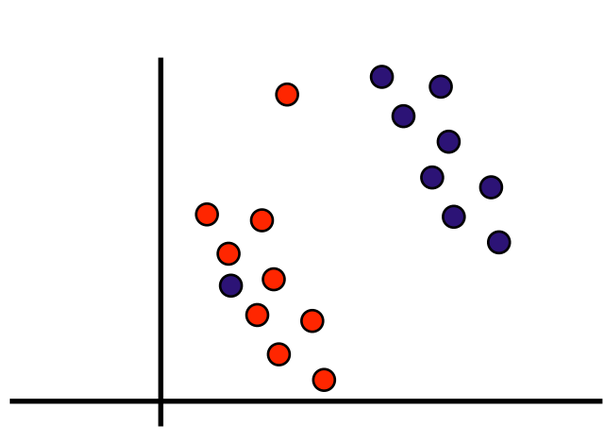
- Since we know **w** we can solve for b.
- Should give same value for all support vectors.

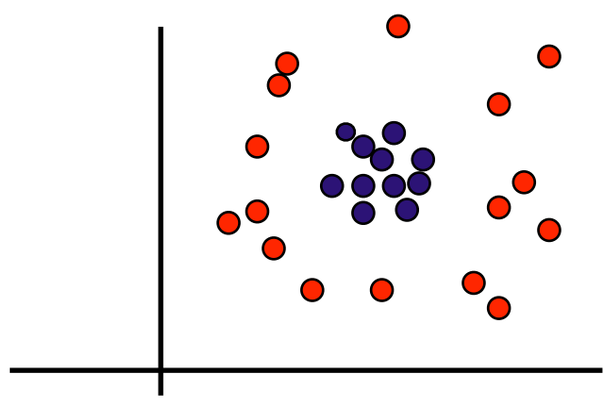
Non Separable Case

- So far we assumed a separating hyperplane exists.
- If it doesn't, our optimization problem is infeasible.



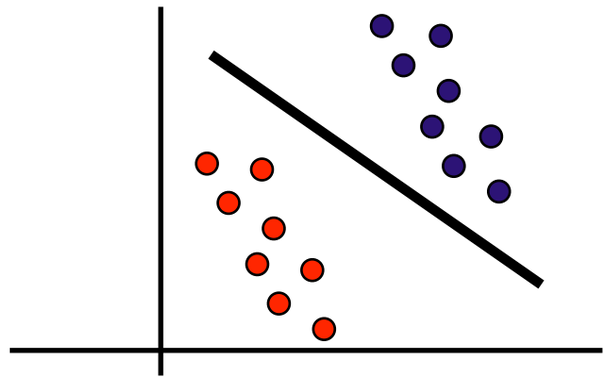
- For real data, we don't want to make this assumption. Because:
  - Data may be noisy. Linear classifier may still do ok.
  - May come from a non linear rule. Next class!

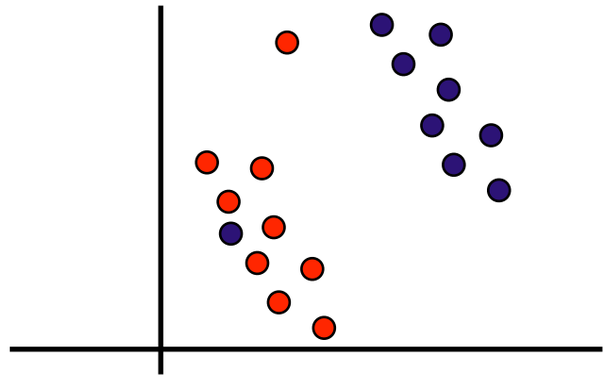


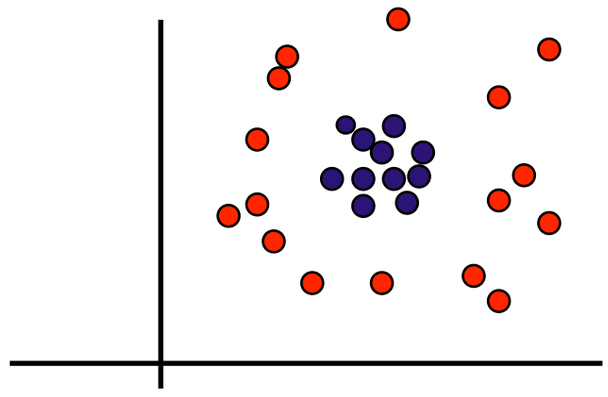


Non Separable Case

- Ideally, we would like to find the classifier that minimizes training error.
- But:
  - Turns out this is NP hard.
  - How do we incorporate margin?
- Let's start from the separable case.







Non Separable Case

- Separable case:

$$\begin{array}{ll} \min_{\boldsymbol{w}} & 0.5\|\boldsymbol{w}\|^2 \\ \text{s.}t. & y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i+b\right)\geq 1 \end{array}$$

- Need to “relax” the constraints.

- Allow violation by $\xi_i \geq 0$, but “pay” for violation.

$$\begin{array}{ll} \min_{\boldsymbol{w}} & 0.5\|\boldsymbol{w}\|^2 + C\sum_i \xi_i \\ \text{s.}t. & y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i+b\right)\geq 1-\xi_i \ , \ \xi_i \geq 0 \end{array}$$

- C is a constant that determines how much we care about classification errors as opposed to margin.

Dual for non separable

- Dual is:

$$\begin{array}{ll} \max & \sum_i \alpha_i - 0.5 \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \\ \text{s.}t. & 0 \leq \alpha_i \leq C, \sum_i \alpha_i y_i = 0 \end{array}$$

- Mapping to primal is as before.

Alternative Interpretation

- Primal is: $\begin{array}{ll} \min_{\boldsymbol{w}} & 0.5\|\boldsymbol{w}\|^2 + C\sum_i \xi_i \\ \text{s.}t. & y_i\left(\boldsymbol{w}\cdot\boldsymbol{x}_i + b\right) \geq 1 - \xi_i \ , \ \xi_i \geq 0 \end{array}$

- Can solve for ξ_i to get: $\xi_i = \max\left[0, 1 - y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b)\right]$

- Problem becomes:

$$\min_{\boldsymbol{w}} C\sum_i \max\left[0, 1 - y_i(\boldsymbol{w}\cdot\boldsymbol{x}_i + b)\right] + 0.5\|\boldsymbol{w}\|_2^2$$

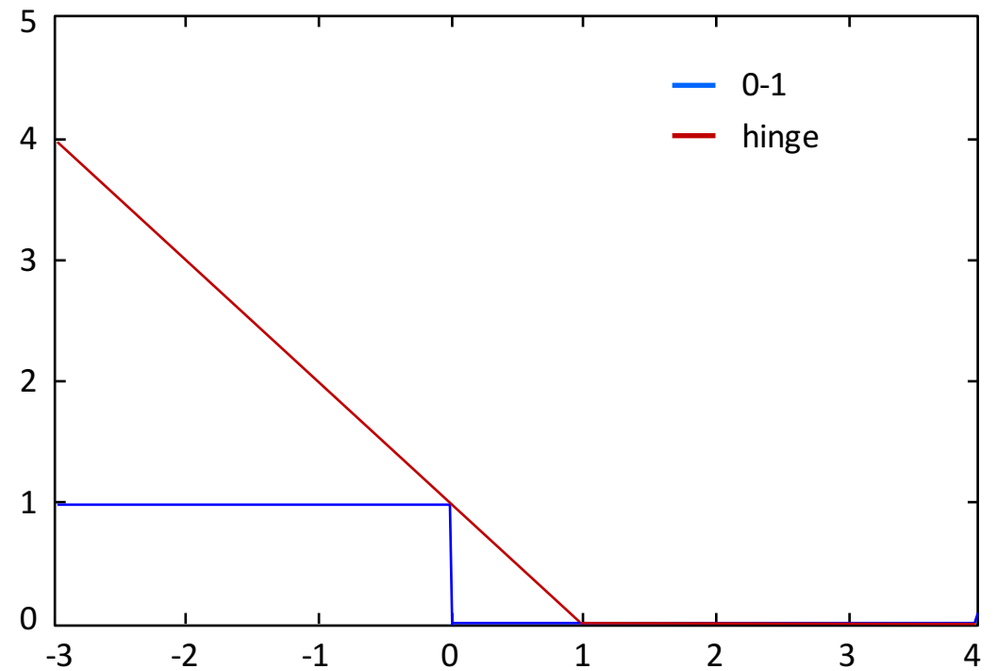
Alternative Interpretation

- Primal is: $\min_{\boldsymbol{w}} C \sum_i \max\left[0, 1 - y_i \boldsymbol{w} \cdot \boldsymbol{x}_i\right] + 0.5 \|\boldsymbol{w}\|_2^2$

- The function: $max\left[0, 1 - y_i \boldsymbol{w} \cdot \boldsymbol{x}_i\right]$ is called the hinge loss.

- Upper bounds the true classification error

- A convex upper bound!



Alternative Interpretation

- Primal is: $\min_{\boldsymbol{w}} C \sum_i \max\left[0, 1 - y_i \boldsymbol{w} \cdot \boldsymbol{x}_i\right] + 0.5 \|\boldsymbol{w}\|_2^2$

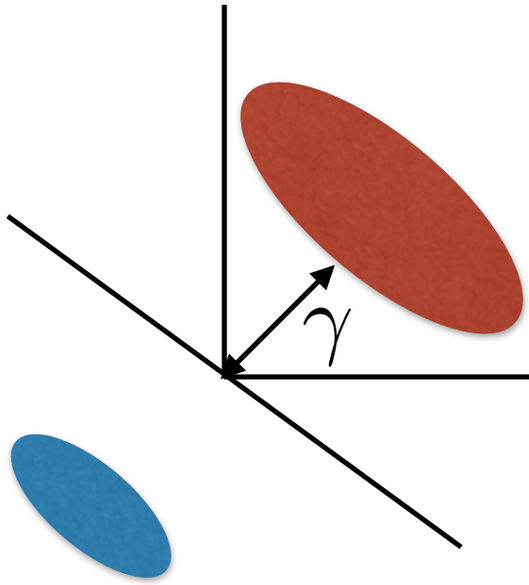
Bound on loss

Regularization

- Very common design pattern.
- Other losses and regularizers can be considered.
- Logistic loss: $\frac{1}{\ln 2} \ln(1 + e^{-y_i \boldsymbol{w} \cdot \boldsymbol{x}_i})$
- L1 regularization: $\|\boldsymbol{w}\|_1 = \sum_i |w_i|$. Sparsity inducing.

SVM and Generalization

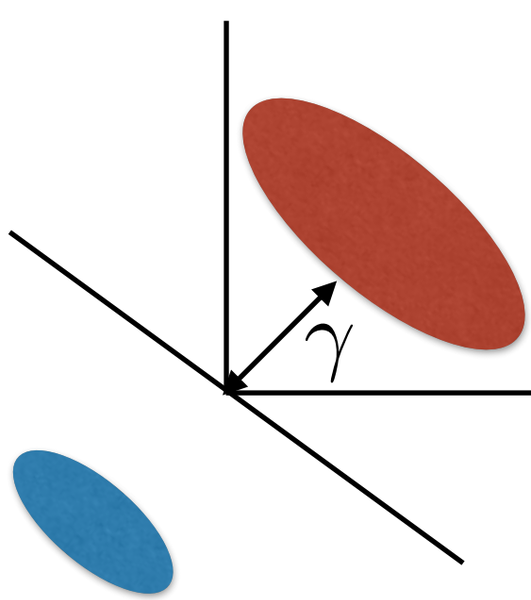
- Intuitively choosing a large margin should improve generalization
- Assume true distribution and classifier are such that the margin is γ
- Expect generalization to behave like γ^{-1}
- But can always increase γ by rescaling
- Denote R the largest norm of **x**
- Generalization scales with $R\gamma^{-1}$



SVM and Generalization

- Assume training error is zero.
- Can be shown that generalization satisfies (up to some logarithmic factors):

$$error(\boldsymbol{w}) \leq \frac{c_1}{m}\frac{R^2}{\gamma^2} + \frac{c_2}{m}\log\frac{m}{\delta}$$



- The VC dimension is replaced by $\frac{R^2}{\gamma^2}$
- Appeared in *"Structural Risk Minimization over Data-Dependent Hierarchies" (98)*

Leave one out bounds

- Another intuition: using few support vectors should lead to good generalization.
- We will show this via leave one out error.

- Denote S^{-i} training sample without $(\boldsymbol{x}_i, y_i)$
- Denote h_S the hypothesis from training on S

$$\hat{R}_{LOO}(S) = \frac{1}{m} \sum_{i=1}^{m} I\left[h_{S^{-i}}(\boldsymbol{x}_i) \neq y_i\right]$$

Leave one out bounds

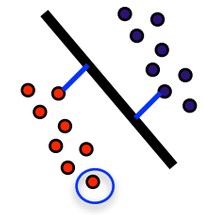
- LOO error is similar in spirit to generalization error. But we only train on m-1 points.
- Denote R(h) the generalization error of h

$$R(h) = E_{(x,y)\sim D} I\left[h(\boldsymbol{x}) \neq y\right]$$

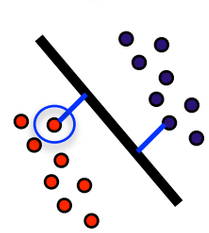
- Can show: $E_{S_m}\left[\hat{R}_{LOO}(S_m)\right] = E_{S_{m-1}}\left[R(h_{S_{m-1}})\right]$
- LOO error and generalization error have same expected value

Leave one out bounds for SVM

- What is the expected LOO error of SVM (separable case).



- If a non-support vector is left out, the solution will not change, and error will be zero.



- Otherwise there might be an error:

$$\hat{R}_{LOO}(S_m) \leq \frac{N_{SV}(S_m)}{m}$$

- Therefore: $E_{S_{m-1}}\left[R(h_{S_{m-1}})\right] \leq \frac{1}{m} E_{S_m}\left[N_{SV}(S_m)\right]$
- Generalization related to number of SVs.