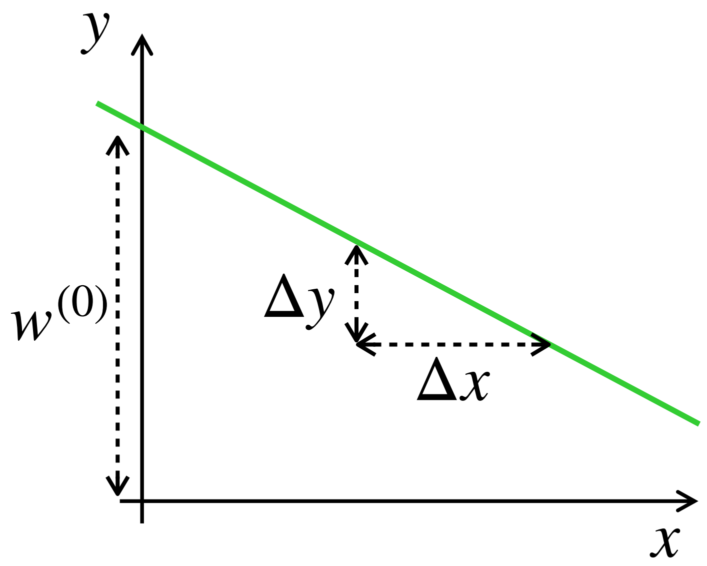
EPFL

Lecture 4: Linear Models for Classification (Part 2)

Recap: A simple parametric model: The line



- Defined by 2 parameters
  - The y-intercept $w^{(0)}$
  - The slope $w^{(1)} = \dfrac{\Delta y}{\Delta x}$
- Mathematically, a line is expressed as

$$y = w^{(1)}x + w^{(0)}$$

Recap: Hyperplane

- This can be generalized to higher dimensions
- In dimension D, we can write

$$y = w^{(0)} + w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + \ldots + w^{(D)}x^{(D)} = \mathbf{w}^T \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(D)} \\ 1 \end{bmatrix}$$

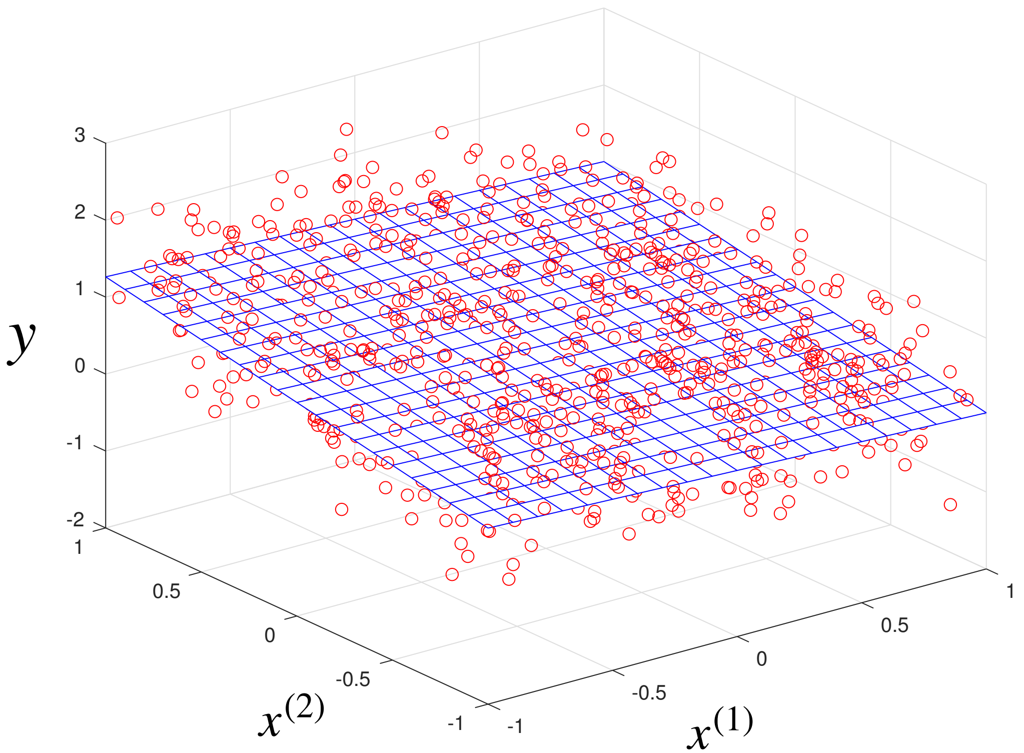
- Ultimately, whatever the dimension, we can write

$$y = \mathbf{w}^T\mathbf{x}$$

with $\mathbf{x} \in \mathbb{R}^{D+1}$, where the extra dimension contains a 1 to account for $w^{(0)}$

Recap: Linear regression

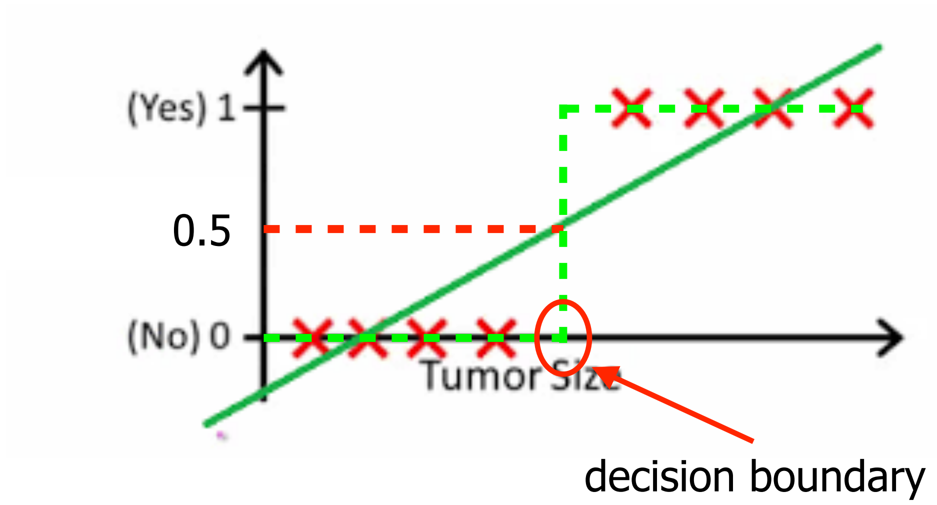
- Given N noisy pairs $\{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i \in \mathbb{R}^D$ (below, $D = 2$), find the parameters $\mathbf{w}^*$ that best fits these observations



Recap: Binary classification as regression

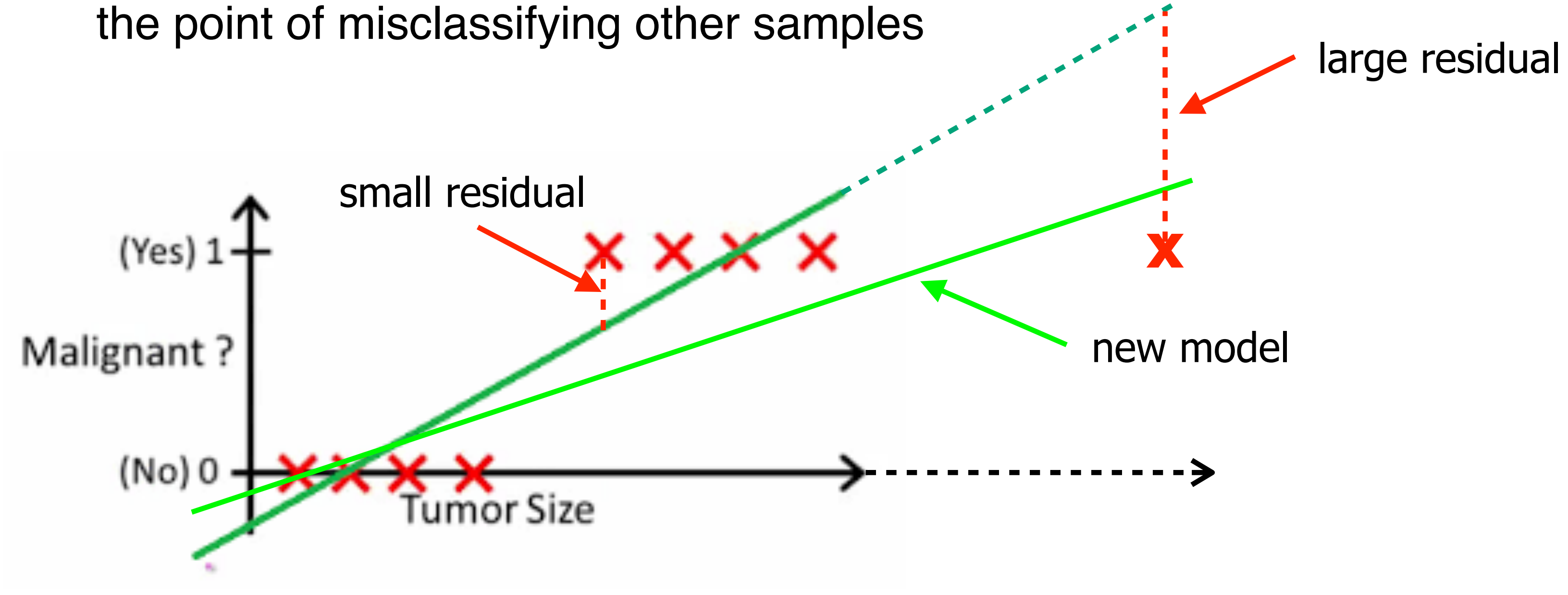
- Example (from Andrew Ng's course): Classify a tumor as benign vs malignant
  - 1D input (tumor size), 1D output (malignant vs benign)
  - Using a linear model, we can predict the label as $\hat{y} = w^{(1)}x + w^{(0)}$

$$\text{label} = \begin{cases} 1 & \text{if } \hat{y} \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$



Recap: Failures

- Back to the tumor prediction example
  - The residuals for “normal” samples are quite small
  - However, a sample with a much larger tumor size (with true label 1) would have a much larger residual
  - Training with such a sample would strongly affect the model parameters, to the point of misclassifying other samples

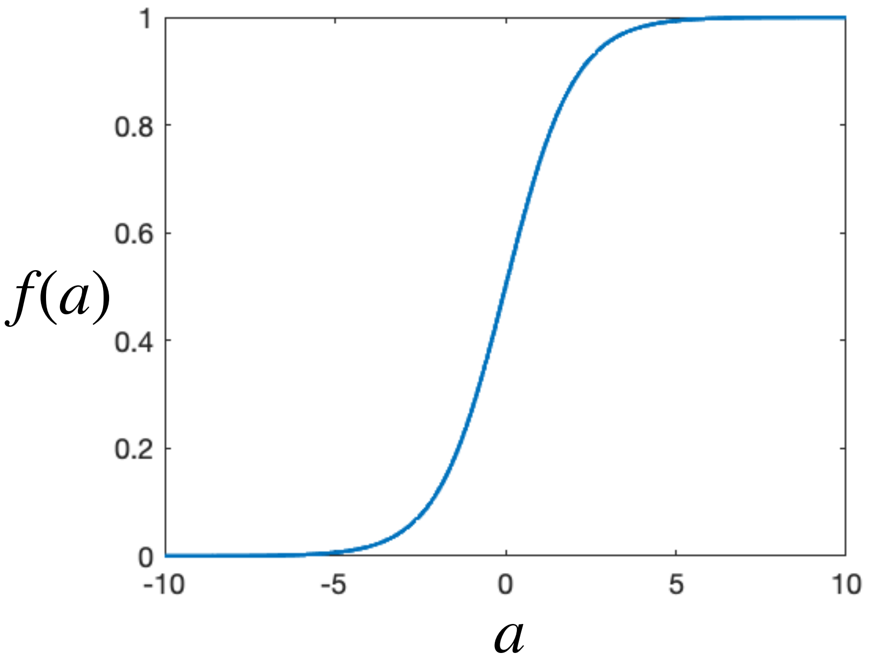


  - This is because least-square classification fits a line to non-linear data

Recap: Adding non-linearity

- To model a classification problem, one can use a smooth approximation of the step function, such as the *logistic sigmoid function*

$$f(a) = \frac{1}{1 + \exp(-a)}$$



Recap: Probabilistic interpretation

- In the binary case, with D dimensional inputs, we can write the prediction of the model as

$$\hat{y}(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

- Such a prediction can be interpreted as the probability that $\mathbf{x}$ belongs to the positive class (or as a score for the positive class)

- The probability to belong to the negative class (or score for the negative class) is then computed as $1 - \hat{y}(\mathbf{x})$

Recap: Logistic regression: Training

- Given N training samples, to learn the parameters $\mathbf{w}$ of the logistic regression model, we minimize the cross-entropy

$$R(\mathbf{w}) = -\sum_{i=1}^{N} \left(y_i \ln(\hat{y}_i) + (1 - y_i)\ln(1 - \hat{y}_i)\right)$$

- Since the true y_i is either 0 or 1, the cross-entropy can be re-written as

$$R(\mathbf{w}) = -\sum_{i \in \text{positive samples}} \ln(\hat{y}_i) - \sum_{i \in \text{negative samples}} \ln(1 - \hat{y}_i)$$

- Minimization is done via gradient descent

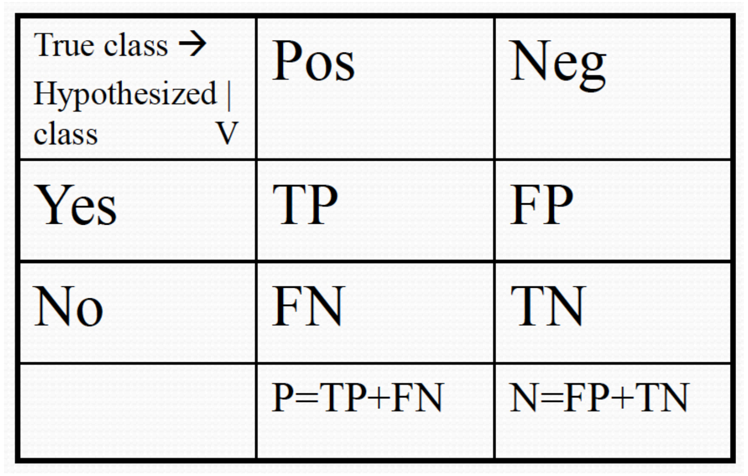
Exercises: Solution

- Describe two main differences between linear regression and logistic regression.
  1. Linear regression is used for regression tasks, i.e., to predict continuous values, whereas logistic regression is used for classification, i.e., to predict class labels.
  2. Logistic regression adds a nonlinearity to the output of the linear model.

- Describe one similarity between linear regression and logistic regression.
  - They internally rely on the same linear model and thus, for a given input dimension, have the same number of parameters.

Recap: Evaluating a classifier

- Confusion matrix: Binary case (e.g., spam vs non-spam email)



True class → Hypothesized | class V	Pos	Neg
Yes	TP	FP
No	FN	TN
	P=TP+FN	N=FP+TN

TP: True positive (number of samples correctly classified as positive)
FP: False positive (number of samples incorrectly classified as positive)

TN: True negative (number of samples correctly classified as negative)
FN: False negative (number of samples incorrectly classified as negative)

P: Total number of positive samples
N: Total number of negative samples

Figure from N. Japkowicz's ICMLA tutorial

Recap: Classification metrics

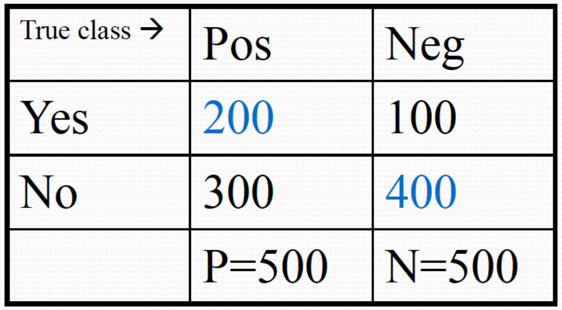
- Last time, we covered several metrics:
  - Accuracy: Percentage of correctly classified samples
  - Precision: Percentage of samples classified as positives that are truly positives
  - Recall: Percentage of positive samples that are correctly classified as positives
  - Area under the ROC curve
- Let me add two metrics that I skipped last time

Classification metrics

- False positive rate: Percentage of negative samples that are incorrectly classified as positives

$$FP\ rate = \frac{FP}{N}$$

- Example:



True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

$$FP\ rate = \frac{100}{500} = 0.2$$

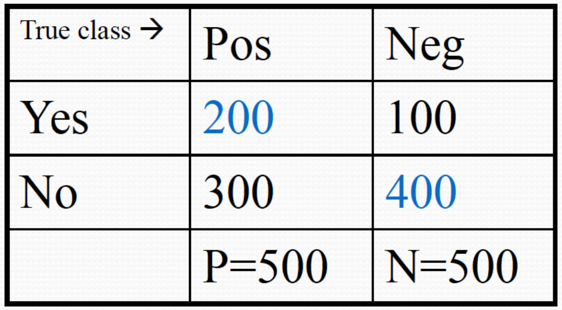
Figure from N. Japkowicz's ICMLA tutorial

Classification metrics

- F1 score: Single number that combines precision and recall

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

- Example:



True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

$$F1 = 2 \cdot \frac{0.667 \cdot 0.4}{0.667 + 0.4} = 0.5$$

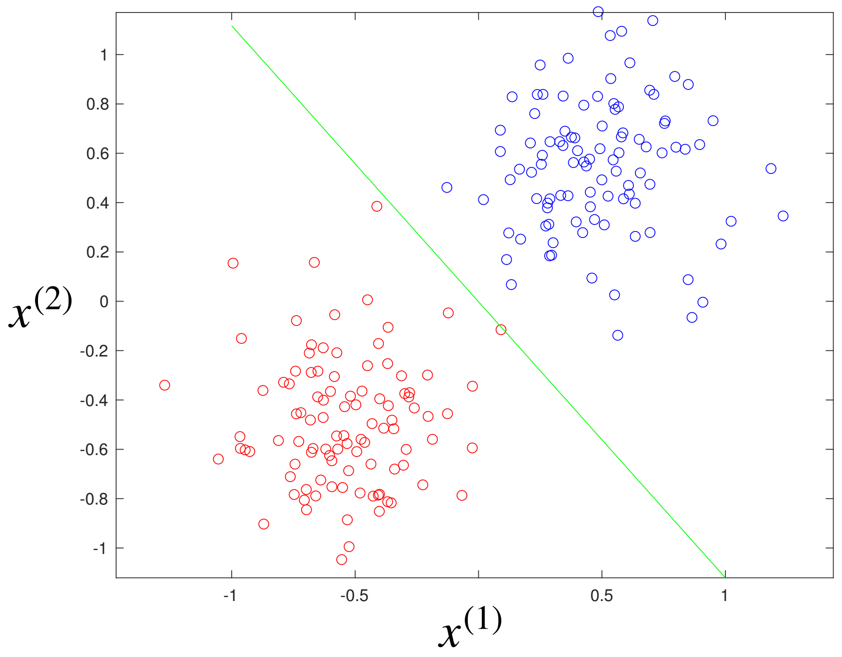
Figure from N. Japkowicz's ICMLA tutorial

Goals of today’s lecture

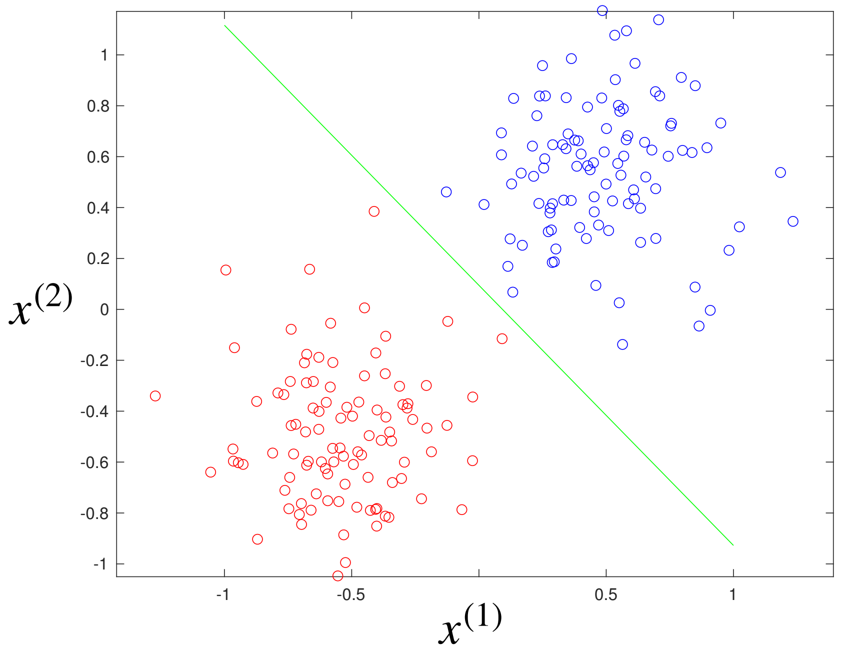
- Introduce the notion of margin between classes
- Derive the formulation for a maximum margin classifier (SVM), first for linearly separable data and then for non-linearly separable data
- Introduce basic notions of constrained optimization

Decision boundaries

- Different classifiers have different decision boundaries



Least-square classification



Logistic regression

- Can we define what a good decision boundary is?

Decision boundary properties

- The decision boundary is related to the parameters $\mathbf{w}$ in the following manner (N.B. today, the negative samples have label -1, not 0)
  - Let $\tilde{\mathbf{w}}$ be the vector of parameters without $w^{(0)}$

- Two points $\mathbf{x}_1$, $\mathbf{x}_2$ on the decision boundary have the same prediction ($\hat{y}_1 = \hat{y}_2 = 0$), thus

$$\tilde{\mathbf{w}}^T\mathbf{x}_1 + w^{(0)} - (\tilde{\mathbf{w}}^T\mathbf{x}_2 + w^{(0)}) = \hat{y}_1 - \hat{y}_2 = 0$$

$$\tilde{\mathbf{w}}^T(\mathbf{x}_1 - \mathbf{x}_2) = \hat{y}_1 - w^{(0)} - (\hat{y}_2 - w^{(0)}) = 0$$

and so $\tilde{\mathbf{w}}$ is orthogonal to the decision boundary

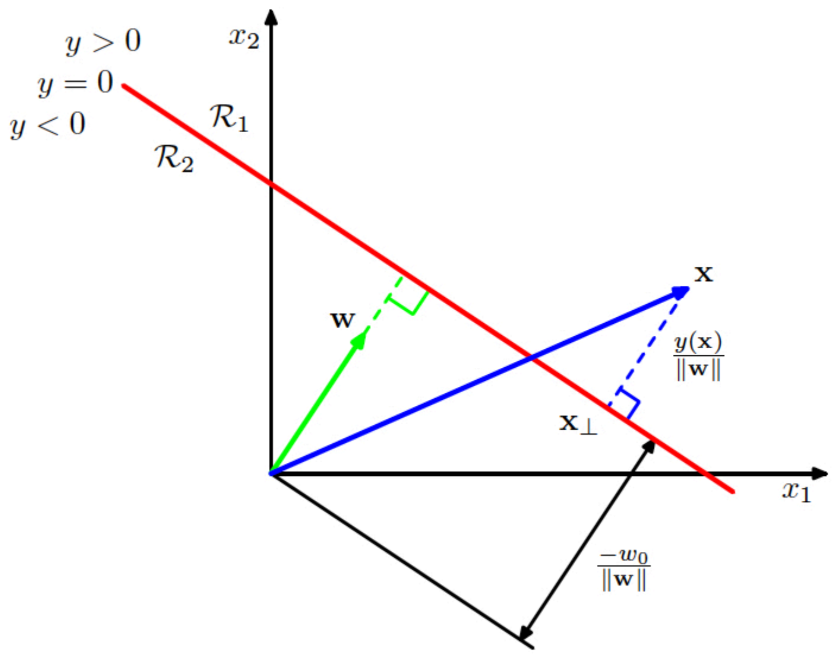


Figure from Bishop, Chapter 4.1.1

($\mathbf{w}$ in the fig. corresponds to $\tilde{\mathbf{w}}$)

Decision boundary properties

- Let $\mathbf{x}$ be an arbitrary point, and $\mathbf{x}_\perp$ its orthogonal projection on the decision boundary

- Because $\tilde{\mathbf{w}}$ is orthogonal to the decision boundary, we can write

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\tilde{\mathbf{w}}}{||\tilde{\mathbf{w}}||}$$

Multiplying on both sides by $\tilde{\mathbf{w}}^T$ yields

$$\tilde{\mathbf{w}}^T\mathbf{x} = \tilde{\mathbf{w}}^T\mathbf{x}_\perp + r||\tilde{\mathbf{w}}||$$

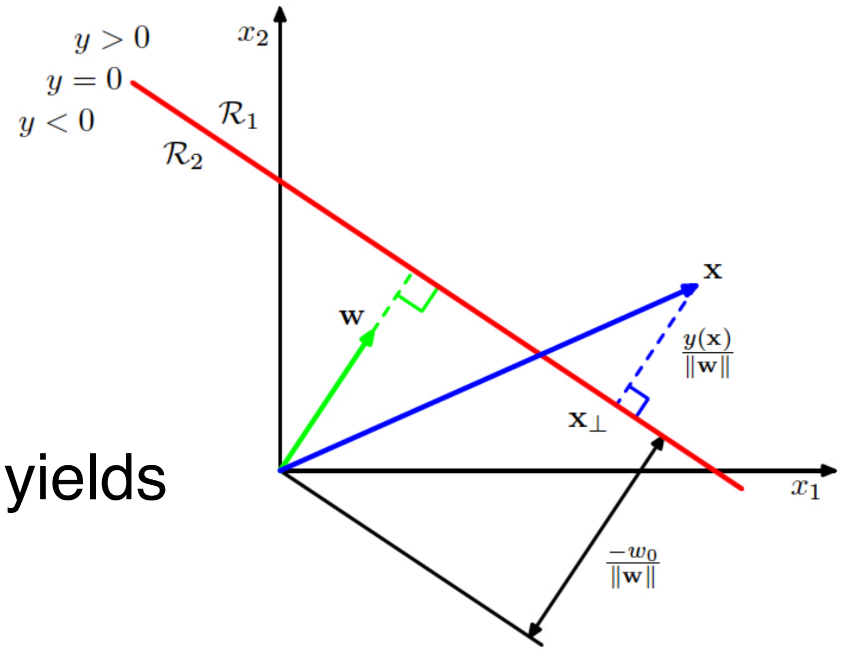


Figure from Bishop, Chapter 4.1.1

($\mathbf{w}$ in the fig. corresponds to $\tilde{\mathbf{w}}$)

Decision boundary properties

- Because $\mathbf{x}_\perp$ is on the boundary, $\hat{y}(\mathbf{x}_\perp) = 0$, and so

$$\tilde{\mathbf{w}}^T\mathbf{x} = \tilde{\mathbf{w}}^T\mathbf{x}_\perp + r\|\tilde{\mathbf{w}}\| = \hat{y}(\mathbf{x}_\perp) - w^{(0)} + r\|\tilde{\mathbf{w}}\| = -\,w^{(0)} + r\|\tilde{\mathbf{w}}\|$$

- Passing $w^{(0)}$ on the lefthand side yields

$$\tilde{\mathbf{w}}^T\mathbf{x} + w^{(0)} = \hat{y}(\mathbf{x}) = r\|\tilde{\mathbf{w}}\|$$

And so, ultimately

$$r = \frac{\hat{y}(\mathbf{x})}{\|\tilde{\mathbf{w}}\|}$$

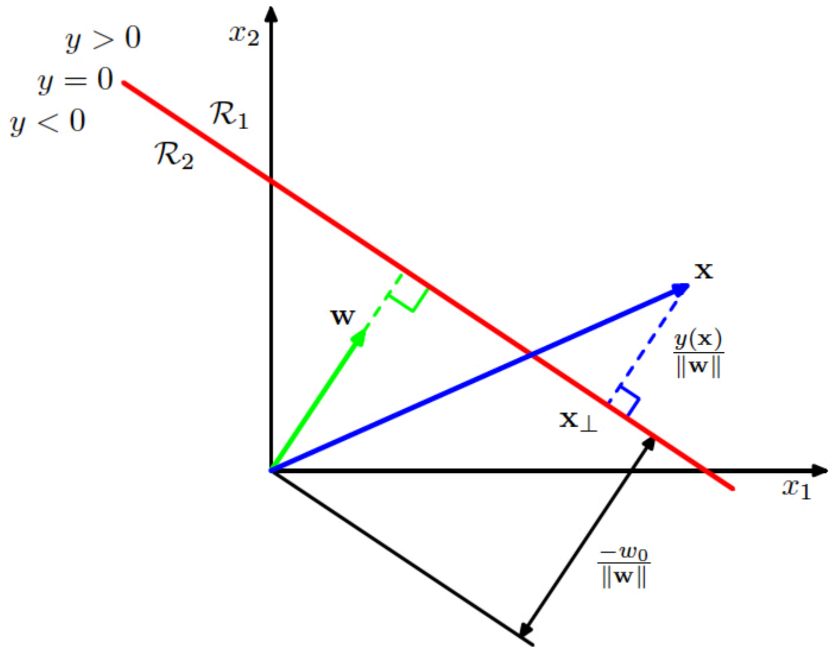


Figure from Bishop, Chapter 4.1.1

($\mathbf{w}$ in the fig. corresponds to $\tilde{\mathbf{w}}$)

Distance to the decision boundary

- In words: The signed distance from a point $\mathbf{x}$ to the boundary is given by its prediction $\hat{y}(\mathbf{x})$ and the parameters $\tilde{\mathbf{w}}$ as

$$r = \frac{\hat{y}(\mathbf{x})}{||\tilde{\mathbf{w}}||}$$

- We can use this to find a classifier whose decision boundary is as far as possible from all the points

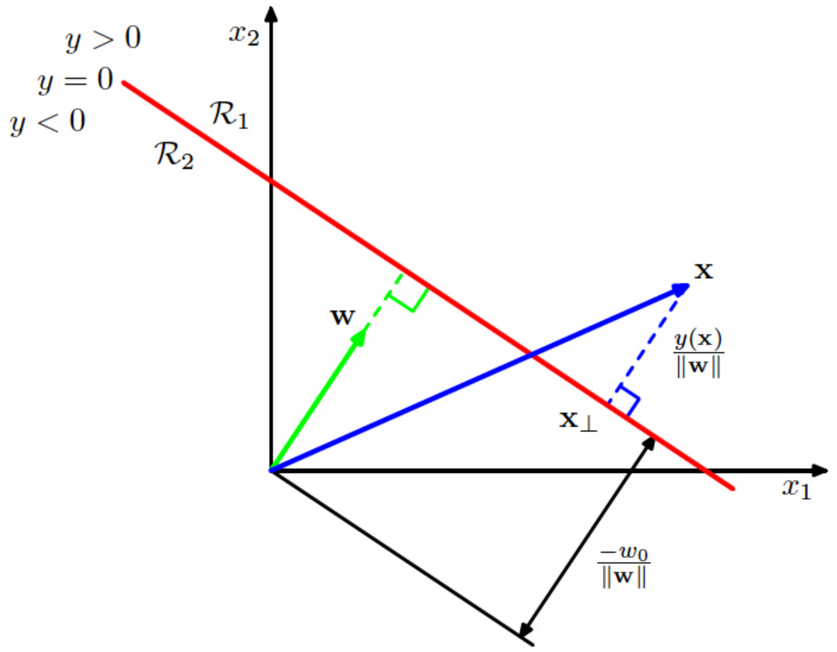
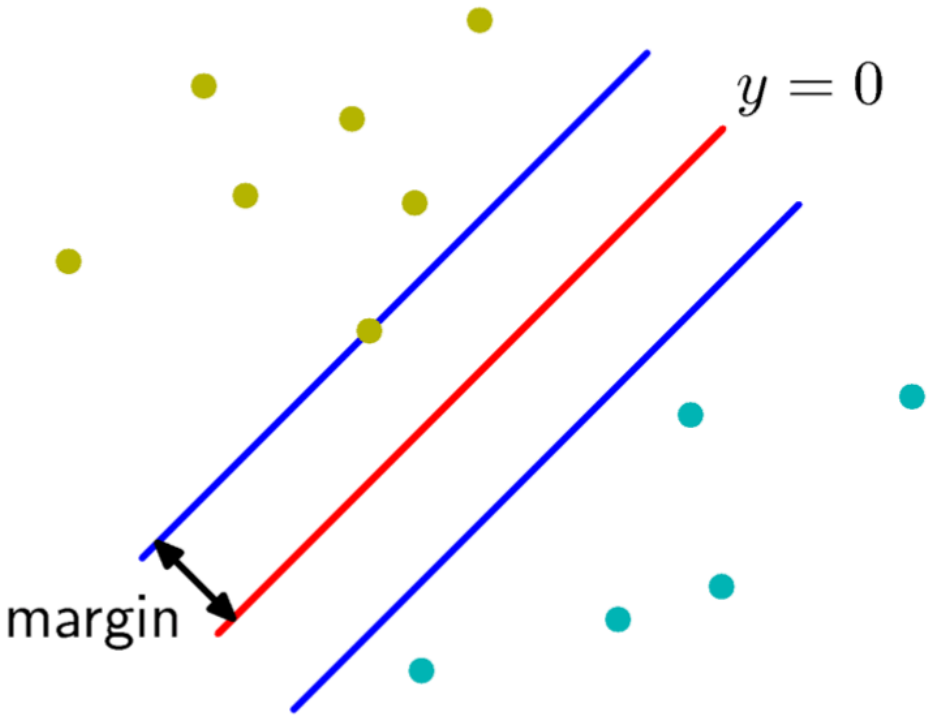


Figure from Bishop, Chapter 4.1.1

($\mathbf{w}$ in the fig. corresponds to $\tilde{\mathbf{w}}$)

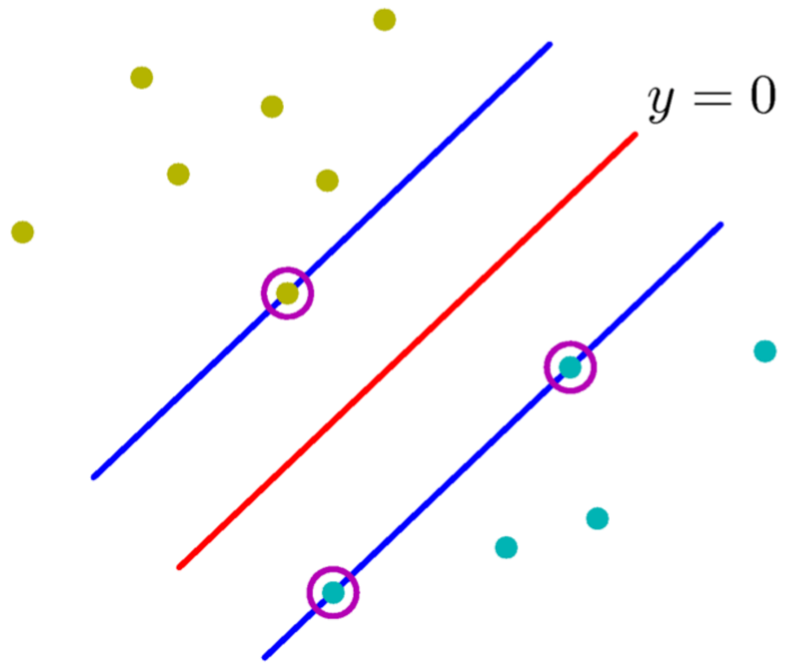
Margin (Chapter 7.1 in Bishop's book)

- The orthogonal distance between the boundary and the nearest sample is called *margin*



Support vectors

- A good decision boundary should maximize the margin
  - Such a boundary is determined by a subset of the data points, named *support vectors* (indicated with circles)



Maximum margin classifier

- For a solution where all the points are correctly classified

$$y_i \cdot \hat{y}_i = y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) > 0$$

- Making use of the definition of the signed distance to the boundary, we can write the (unsigned) distance as

$$\tilde{r}_i = \frac{y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})}{||\tilde{\mathbf{w}}||}$$

- A maximum margin classifier then aims to maximize this distance for the point closest to the boundary, i.e., maximize the minimum such distance

Maximum margin classifier

- Mathematically, this can be expressed as

$$\{\tilde{\mathbf{w}}^*, w^{(0)*}\} = \text{argmax}_{\tilde{\mathbf{w}}, w^{(0)}} \underbrace{\min_{i=1}^{N} \left(\frac{y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})}{||\tilde{\mathbf{w}}||} \right)}$$

Margin: Distance between the boundary and the nearest sample

- Unfortunately, solving this optimization problem is difficult

- We will therefore convert it to an equivalent problem that is easier to solve

Maximum margin classifier: Scale

- Note that the distance to the boundary is invariant to scaling the parameters

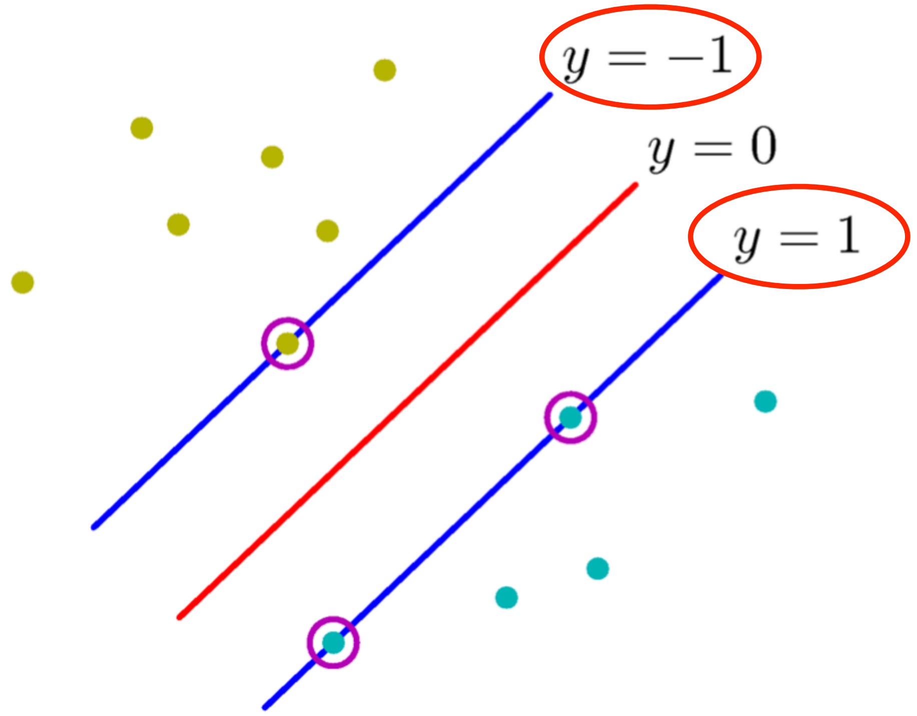
$$\frac{y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})}{||\tilde{\mathbf{w}}||} = \frac{\lambda \cdot y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})}{\lambda \cdot ||\tilde{\mathbf{w}}||} = \frac{y_i \cdot (\lambda \cdot \tilde{\mathbf{w}}^T \mathbf{x}_i + \lambda \cdot w^{(0)})}{||\lambda \cdot \tilde{\mathbf{w}}||}$$

- This means that there are an infinite number of equivalent solutions for $\tilde{\mathbf{w}}$ and $w^{(0)}$
- And we can constrain the parameters to be such that, for the point j that is closest to the boundary,

$$y_j \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_j + w^{(0)}) = 1$$

Maximum margin classifier: Scale

- In other words, with such a fixed scale, the support vectors will lie on hyperplanes at a distance $1/||\tilde{\mathbf{w}}||$ from the decision boundary



Maximum margin classifier

- This means that any point i must satisfy

$$y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1$$

- For the points for which the equality holds, the constraints are said to be *active*; for the other ones, they are *inactive*
  - There will always be at least one active constraint, because there is always one closest point to the boundary
  - Once the margin is maximized, there will always be at least two active constraints, one from each class

- Because of these constraints, we have that

$$\min_i \left(y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \right) = 1$$

Maximum margin classifier

- This means that we can, to maximize the margin, we only need to maximize $1/||\tilde{\mathbf{w}}||$

- This is equivalent to minimizing $||\tilde{\mathbf{w}}||^2$, so we can write the solution to a max-margin classifier as

$$\min_{\tilde{\mathbf{w}}, w^{(0)}} \frac{1}{2} ||\tilde{\mathbf{w}}||^2$$

$$\text{subject to} \quad y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1 \quad \forall i$$

- This formulation is called *Support Vector Machine*
  - The factor 1/2 was added for later convenience and does not affect the solution

Support Vector Machine

- The SVM formulation

$$\min_{\tilde{\mathbf{w}}, w^{(0)}} \frac{1}{2} ||\tilde{\mathbf{w}}||^2$$

$$\text{subject to} \;\; y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1 \;, \;\; \forall i$$

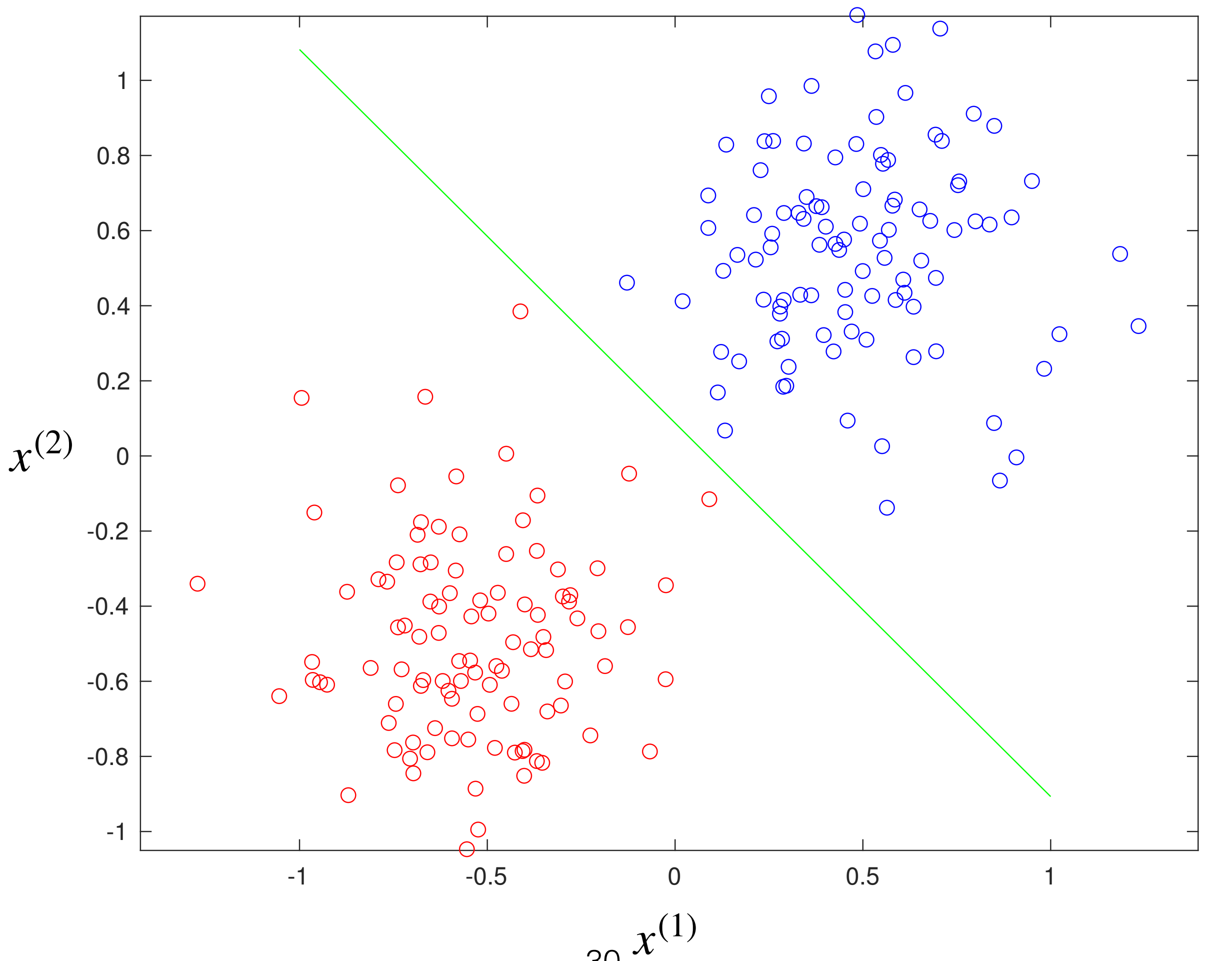
is a quadratic program

  - Quadratic objective function, with linear constraints

- Here, the problem is convex, and can be solved to optimality with standard iterative solvers

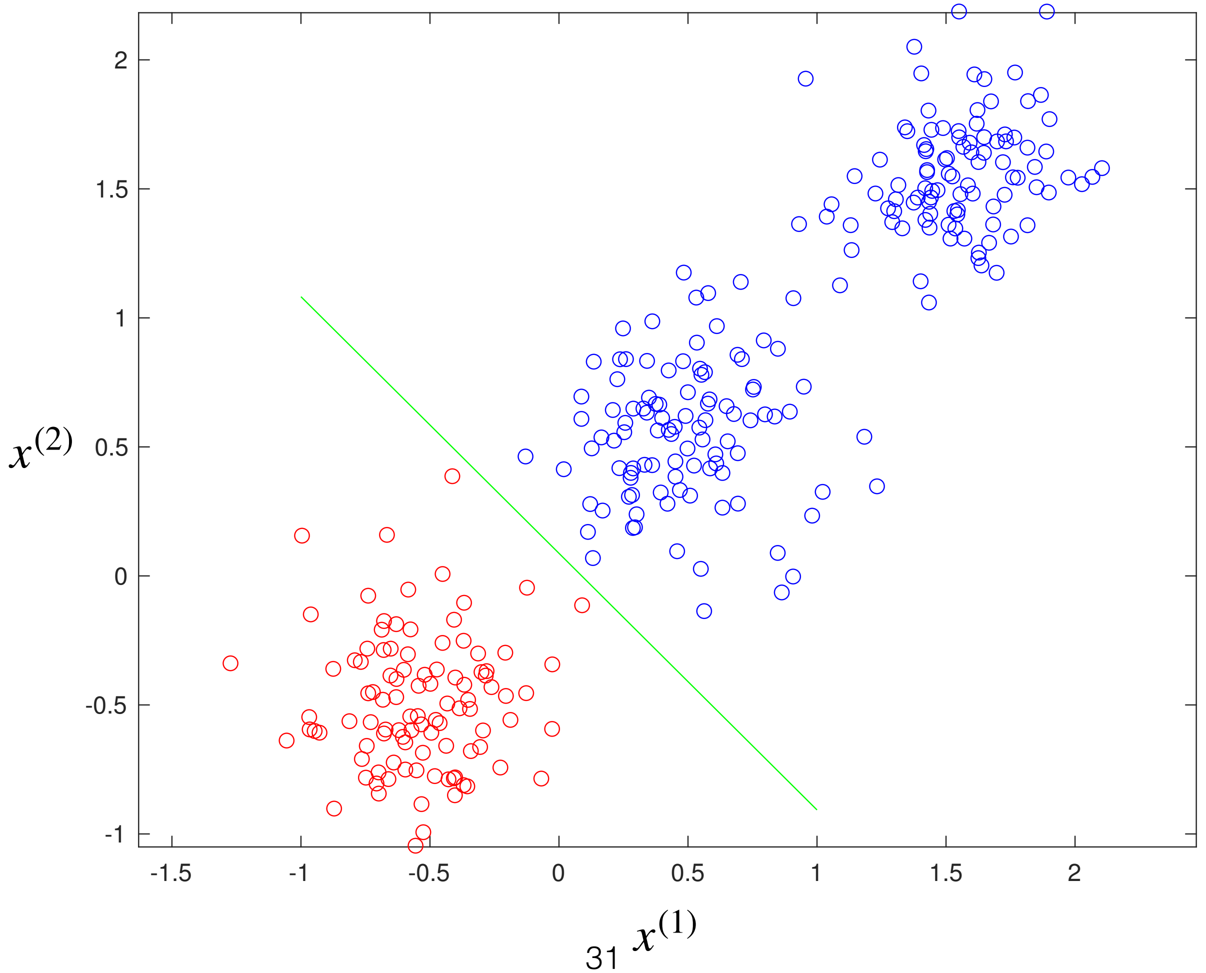
SVM Example

- Back to our toy data (2D samples from 2 classes)



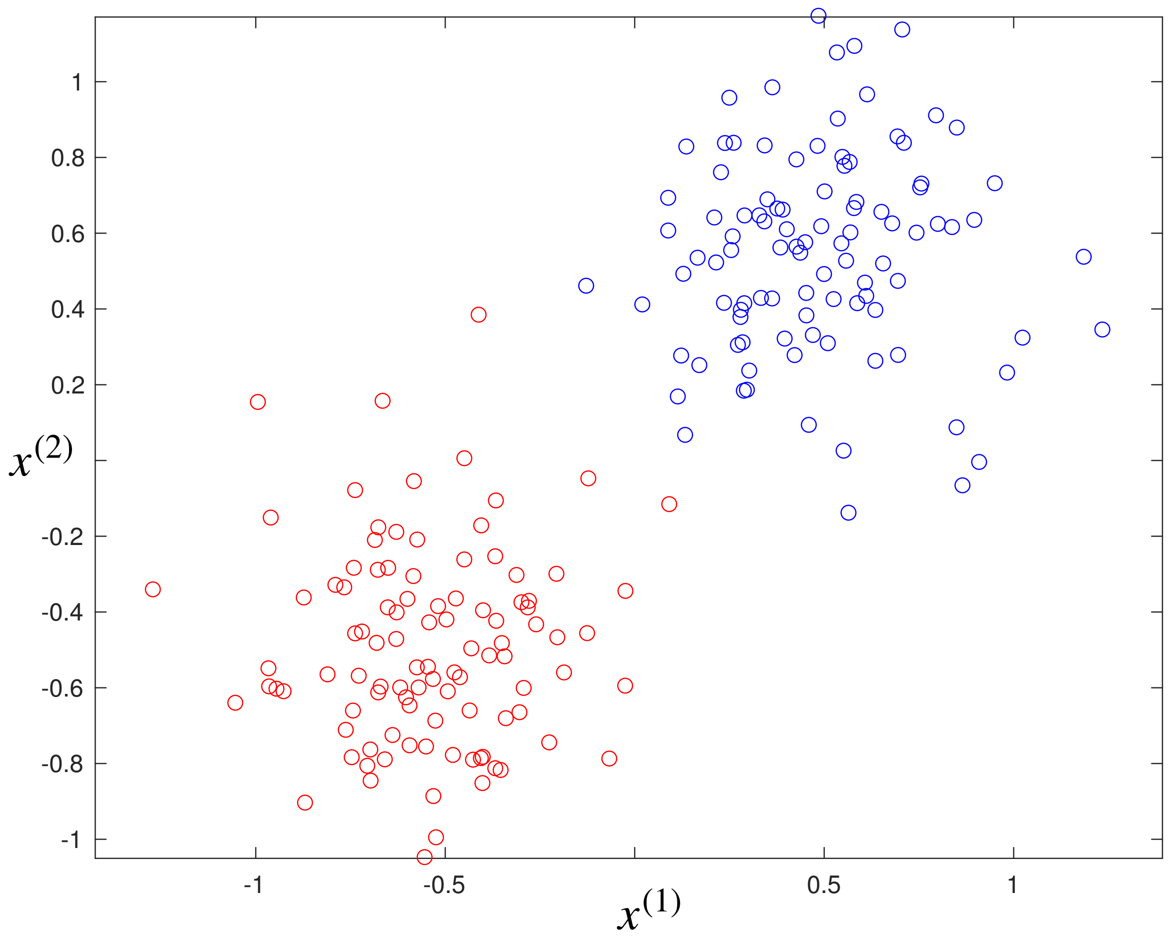
SVM Example

- In the presence of additional samples, it still works



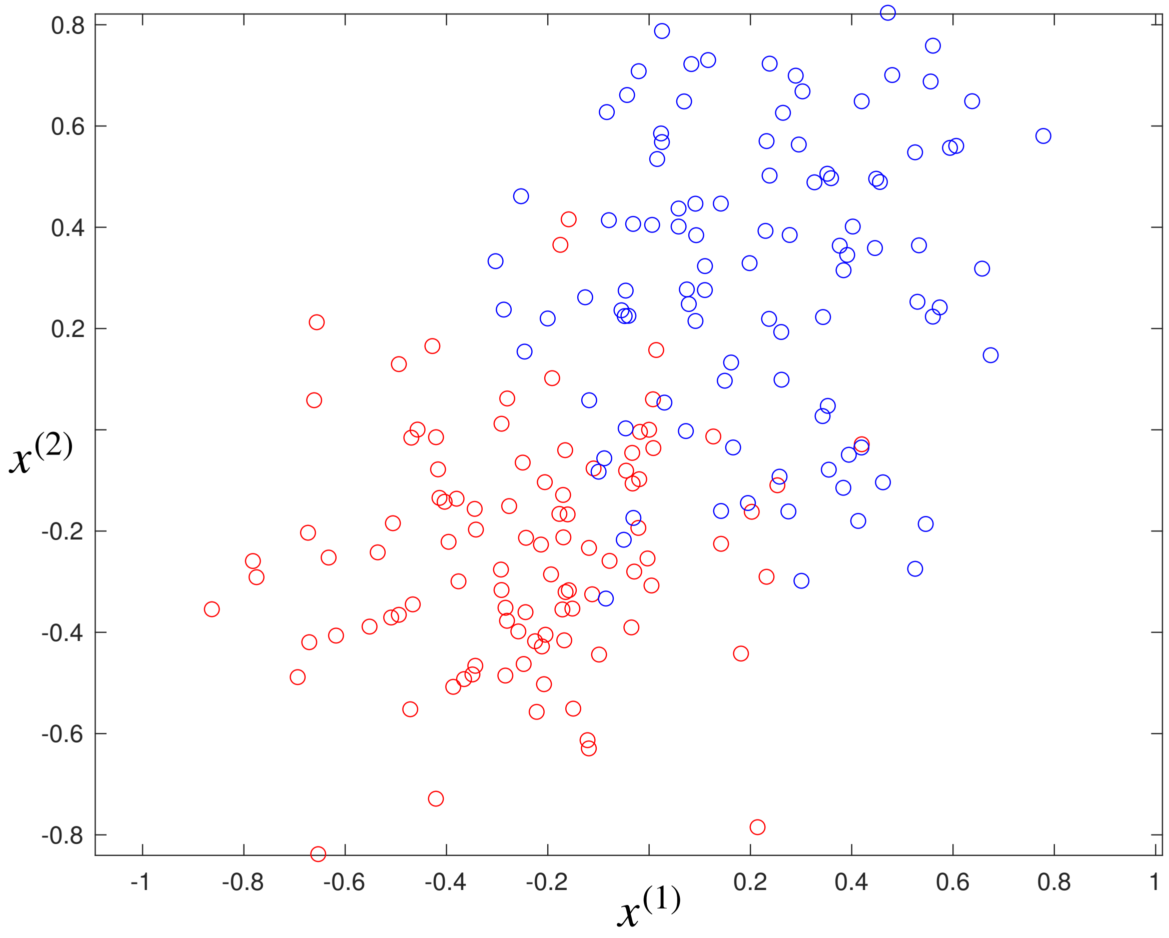
Overlapping classes

- In practice, the data essentially never looks like this



Overlapping classes

- but rather like this



Support Vector Machine

- This means that the constraints in the SVM formulation

$$\min_{\tilde{\mathbf{w}}, w^{(0)}} \frac{1}{2} ||\tilde{\mathbf{w}}||^2$$

$$\text{subject to } \; y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1 \;, \;\; \forall i$$

  cannot all be satisfied

- So the problem simply cannot be solved as such
  - For some samples, we need to allow $y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})$ to be less than 1
  - Note that for the other classifiers we have seen so far, this is not a real problem, because they do not have hard constraints; this would simply translate to a higher loss value

Slack variables

- To overcome this, we introduce an additional slack variable ξ_i for each sample

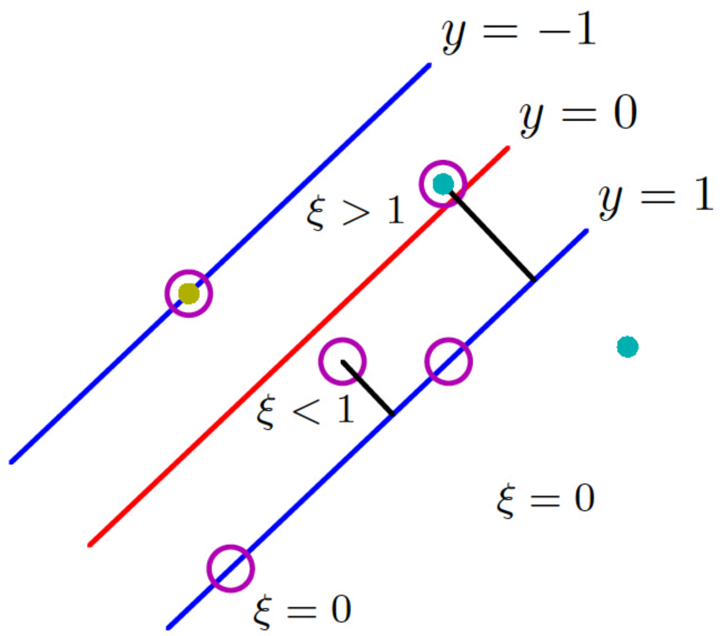
- We then re-write the constraints as

$$y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1 - \xi_i$$

- With $\xi_i \geq 0$, this allows violating the original contraints

Slack variables

- Visually, the slack variables represent how much one violates the original constraints
  - If $0 < \xi_i \leq 1$, sample i lies inside the margin, but is still correctly classified
  - If $\xi_i \geq 1$, then sample i is misclassified



Slack variables

- During training, we then optimize the slack variables jointly with the other parameters
- Naively, we could thus think of writing the problem as

$$\min_{\tilde{\mathbf{w}}, w^{(0)}, \{\xi_i\}} \frac{1}{2} ||\tilde{\mathbf{w}}||^2$$

$$\text{subject to } \; y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1 - \xi_i \; , \;\; \forall i$$

$$\xi_i \geq 0 \; , \;\; \forall i$$

- However, this would simply allow the model to violate all the original constraints at no cost, and would make a useless classifier

Slack variables

- To prevent this, we encourage the slack variables to remain close to zero
- We therefore write the SVM problem as

$$\min_{\tilde{\mathbf{w}}, w^{(0)}, \{\xi_i\}} \frac{1}{2}\|\tilde{\mathbf{w}}\|^2 + C\sum_{i=1}^{N} \xi_i$$

$$\text{subject to } \; y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1 - \xi_i \;, \;\; \forall i$$

$$\xi_i \geq 0 \;, \;\; \forall i$$

where C sets the influence of the regularizer on the slack variables

Exercise

- The slack variables allow SVM to handle the case where the classes are not truly linearly separable but still close to being so. Describe two reasons why this can happen in practice.

Support vector machine: Demo

- http://www.cristiandima.com/basics-of-support-vector-machines/

SVM: Solution

- The SVM optimization problem can be solved using standard quadratic programming frameworks

- The formulation we have seen so far is referred to as the primal problem

- Nevertheless, one can instead derive the dual SVM problem
    - We will see later today and in a future lecture why this can be useful

Interlude

Constrained optimization & Lagrange duality

Constrained optimization

- In general, an optimization problem can be written

$$\begin{aligned} \min_{\mathbf{w}} \quad & R(\mathbf{w}) \\ \text{subject to} \quad & f_i(\mathbf{w}) \leq 0, \quad i = 1, \ldots, M \qquad (M \text{ inequality constraints}) \\ & h_i(\mathbf{w}) = 0, \quad i = 1, \ldots, P \qquad (P \text{ equality constraints}) \end{aligned}$$

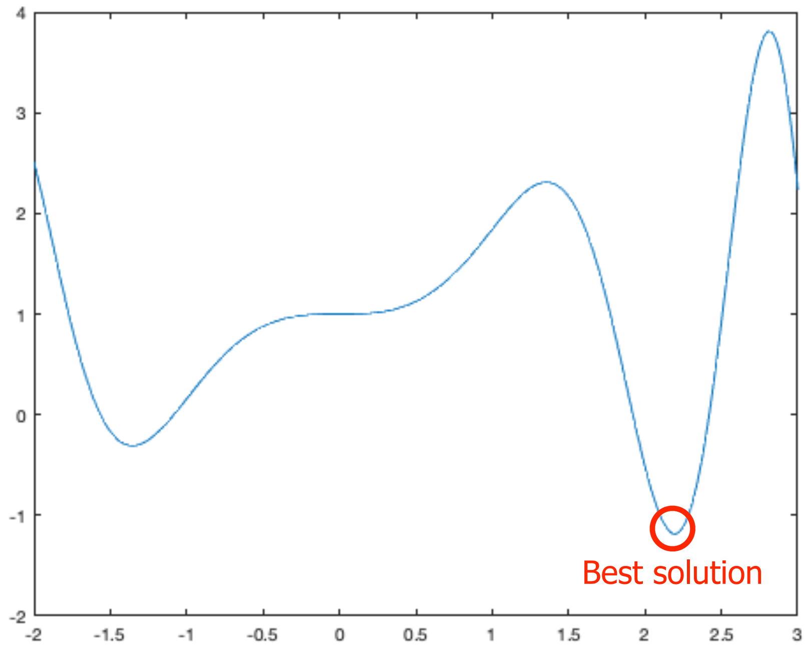
- Note that the constrained problem will typically not have the same solution as the unconstrained one

Adding constraints: Example

- Inequality: 1D sine-based function seen before

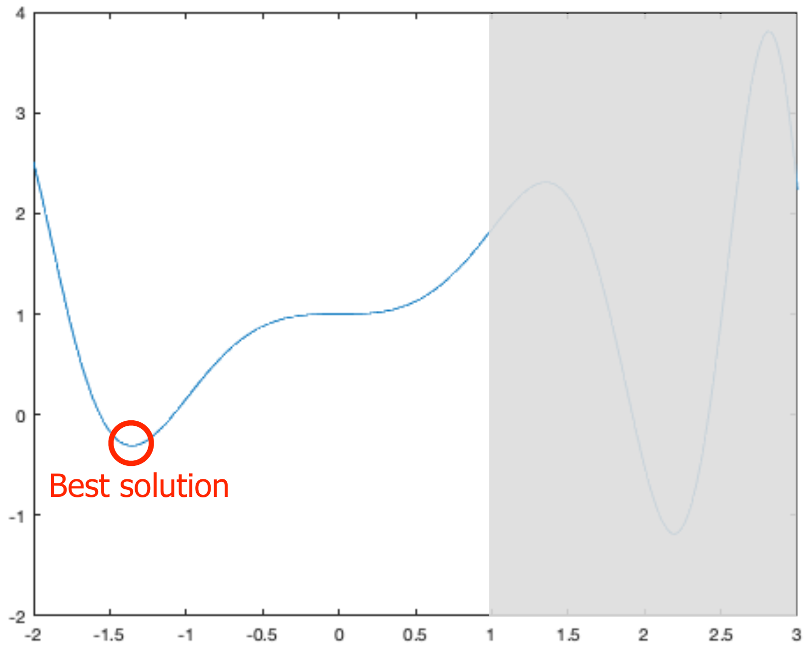
$$R(w) = w\sin(w^2) + 1$$

Unconstrained



Constrained

$f_1(w) = w - 1 \leq 0$

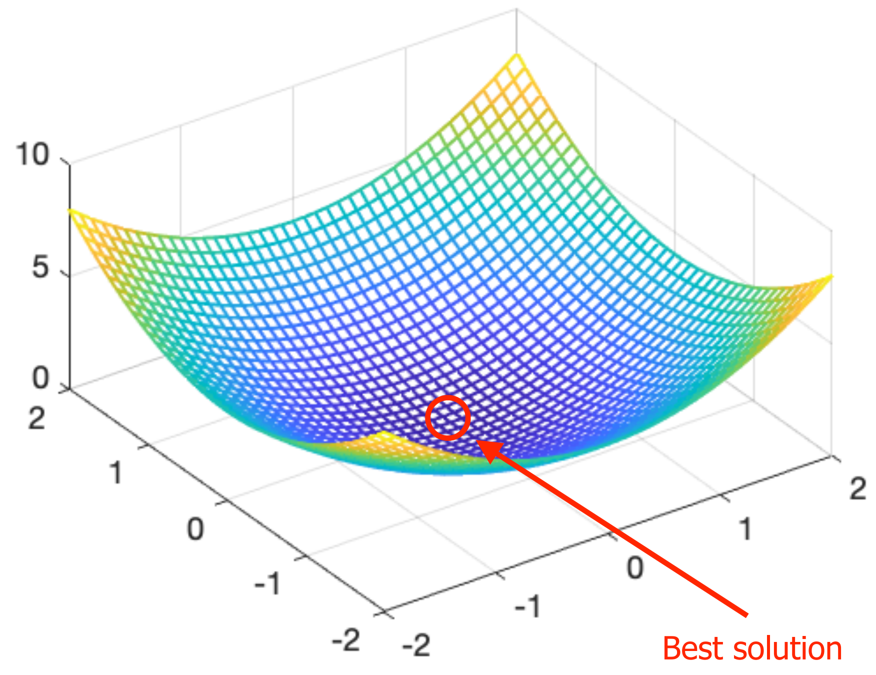


Adding constraints: Example

- Equality: 2D quadratic function

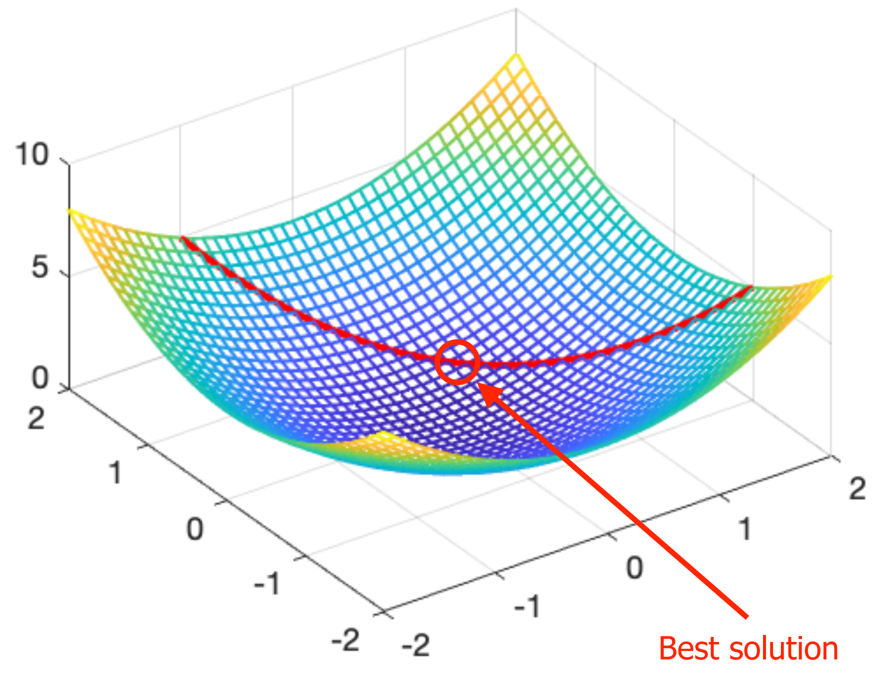
$$R(\mathbf{w}) = \left(w^{(1)}\right)^2 + \left(w^{(2)}\right)^2$$

Unconstrained



Constrained

$h_1(\mathbf{w}) = w^{(1)} + w^{(2)} - 1 = 0$



Lagrangian

- The Lagrangian of an optimization problem

$$\begin{aligned} &\min_{\mathbf{w}} \quad R(\mathbf{w}) \\ &\text{subject to} \quad f_i(\mathbf{w}) \le 0, \quad i = 1,\ldots,M \\ &\qquad\qquad\quad h_i(\mathbf{w}) = 0, \quad i = 1,\ldots,P \end{aligned}$$

is expressed as

$$L(\mathbf{w}, \lambda, \nu) = R(\mathbf{w}) + \sum_{i=1}^{M} \lambda_i f_i(\mathbf{w}) + \sum_{i=1}^{P} \nu_i h_i(\mathbf{w})$$

- λ_i is the Lagrange multiplier associated with $f_i(\mathbf{w}) \le 0$
- ν_i is the Lagrange multiplier associated with $h_i(\mathbf{w}) = 0$

Lagrange dual function

- The Lagrange dual function is expressed as

$$
\begin{aligned}
g(\lambda, \nu) &= \inf_{\mathbf{w}} \; L(\mathbf{w}, \lambda, \nu) \\
&= \inf_{\mathbf{w}} \left(R(\mathbf{w}) + \sum_{i=1}^{M} \lambda_i f_i(\mathbf{w}) + \sum_{i=1}^{P} \nu_i h_i(\mathbf{w}) \right)
\end{aligned}
$$

  - For simplicity, think of $\inf\limits_{\mathbf{w}}$ as $\min\limits_{\mathbf{w}}$
    - In other words, to compute the Lagrange dual function for given values (λ, ν), we need to minimize the Lagrangian w.r.t. $\mathbf{w}$ while keep (λ, ν) fixed

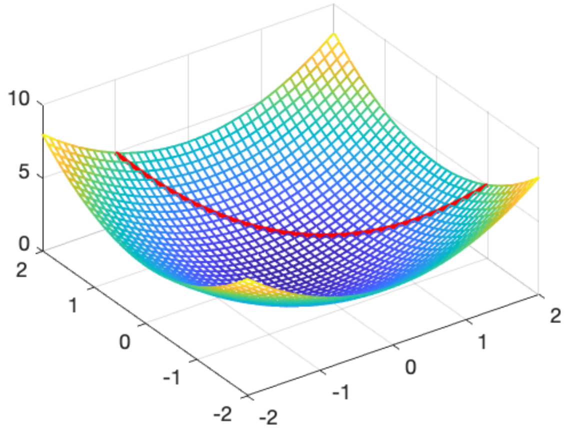
- $g(\lambda, \nu)$ is a concave function (opposite of convex, e.g., single maximum)
- It forms a lower bound to the optimal value R^* of the original problem
  - if $\lambda \geq 0$, then $g(\lambda, \nu) \leq R^*$ for any (λ, ν)

Lagrange dual function: Example

- Quadratic function with equality constraint

$$\min_{\mathbf{w}} \left(w^{(1)}\right)^2 + \left(w^{(2)}\right)^2$$

$$\text{subject to } w^{(1)} + w^{(2)} - 1 = 0$$



$$L(\mathbf{w}, \nu_1) = \left(w^{(1)}\right)^2 + \left(w^{(2)}\right)^2 + \nu_1 \left(w^{(1)} + w^{(2)} - 1\right)$$

- To minimize $L(\mathbf{w}, \nu_1)$ w.r.t. $\mathbf{w}$, we can set the gradient to 0

$$\nabla_{\mathbf{w}} L(\mathbf{w}, \nu_1) = \begin{bmatrix} 2w^{(1)} + \nu_1 \\ 2w^{(2)} + \nu_1 \end{bmatrix} = \mathbf{0} \Leftrightarrow \mathbf{w}^* = \begin{bmatrix} -\nu_1/2 \\ -\nu_1/2 \end{bmatrix}$$

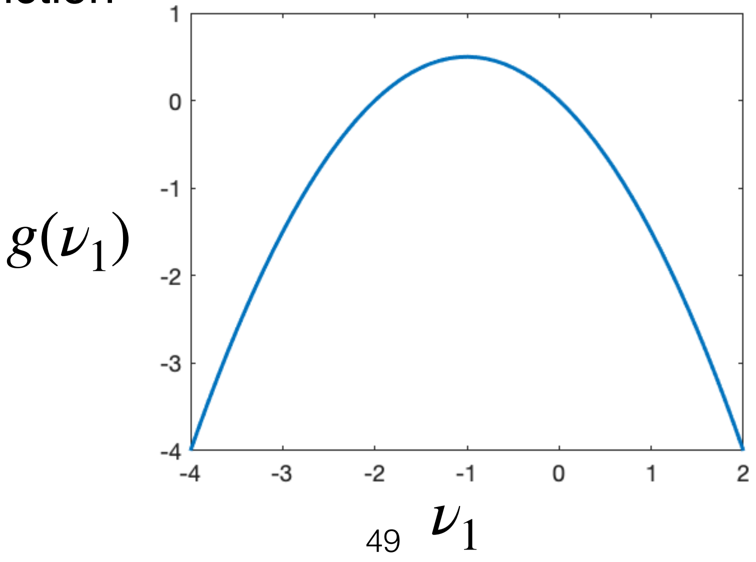
Lagrange dual function: Example

- We can then plug this result in L to obtain

$$g(\nu_1) = L\left(\begin{bmatrix}-\nu_1/2\\-\nu_1/2\end{bmatrix}, \nu_1\right) = \frac{1}{4}\nu_1^2 + \frac{1}{4}\nu_1^2 + \nu_1\left(-\frac{1}{2}\nu_1 - \frac{1}{2}\nu_1 - 1\right)$$

$$= -\frac{1}{2}\nu_1^2 - \nu_1$$

  - This is a concave function



Lagrange dual problem

- Since the Lagrange dual function is a lower bound to the optimal function value, one can aim to maximize it

$$\max_{\lambda,\nu} \; g(\lambda, \nu)$$

$$\text{subject to} \;\; \lambda \geq 0$$

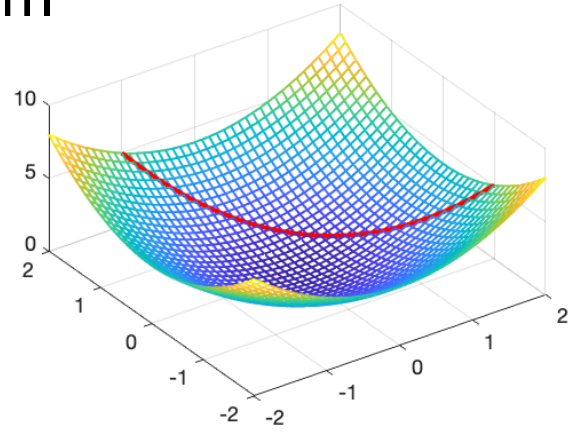
- However, this bound is not always tight
  - The maximum value of the dual problem does not always reach the minimum value of the primal (original) one
  - Nevertheless, under some conditions, it does

Lagrange dual problem: Example

- As shown before, for the convex problem

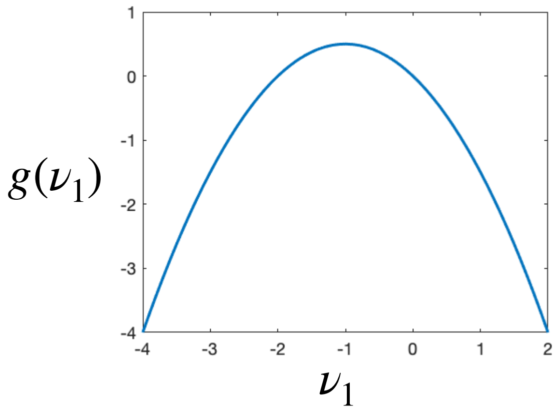
$$\min_{\mathbf{w}} \left(w^{(1)}\right)^2 + \left(w^{(2)}\right)^2$$

subject to $w^{(1)} + w^{(2)} - 1 = 0$



- the dual function is

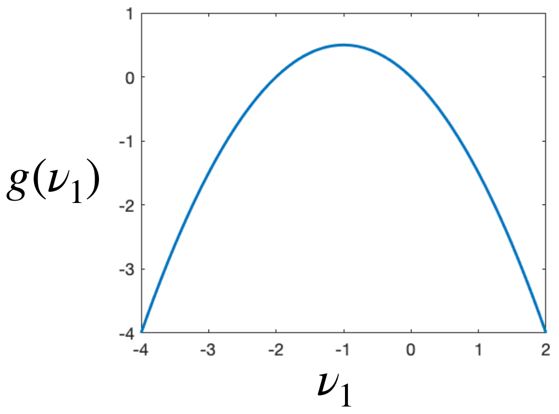
$$g(\nu_1) = -\frac{1}{2}\nu_1^2 - \nu_1$$



Lagrange dual problem: Example

- We can maximize the dual function by zeroing out its derivative

$$-\nu_1 - 1 = 0 \Leftrightarrow \nu_1 = -1$$



- Then, we have that

$$\mathbf{w}^* = \begin{bmatrix} -\nu_1/2 \\ -\nu_1/2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

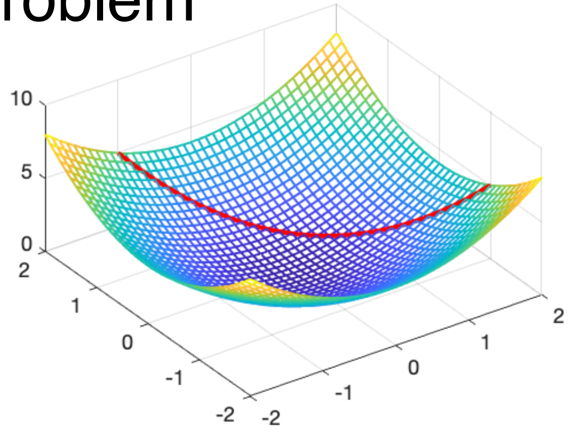
- And the maximum value is $g(-1) = 0.5$

Lagrange dual problem: Example

- Alternatively, we can rewrite the convex problem

$$\min_{\mathbf{w}} \left(w^{(1)}\right)^2 + \left(w^{(2)}\right)^2$$

subject to $w^{(1)} + w^{(2)} - 1 = 0$



- as

$$\min_{w^{(1)}} \left(w^{(1)}\right)^2 + \left(1 - w^{(1)}\right)^2 = 2\left(w^{(1)}\right)^2 - 2w^{(1)} + 1$$

- By zeroing out its derivative, we have

$$4w^{(1)} - 2 = 0 \Leftrightarrow w^{(1)} = 0.5$$

- And so $w^{(2)} = 0.5$, and the optimal function value is 0.5
  - So the bound given by the dual function in this case is tight

End of the interlude

Back to SVM

SVM: Lagrangian

- Let us first consider the case without slack variables
- As seen before, the SVM primal formulation is given by

$$\min_{\tilde{\mathbf{w}}, w^{(0)}} \frac{1}{2} \|\tilde{\mathbf{w}}\|^2$$

$$\text{subject to } \; y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) \geq 1 \; , \;\; \forall i$$

- The Lagrangian of the SVM problem is

$$L(\tilde{\mathbf{w}}, w^{(0)}, \{\alpha_i\}) = \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) - 1 \right)$$

where α_i is the Lagrange multiplier for contraint i, subject to the constraint $\alpha_i \geq 0$

Lagrange dual function

- We can then express the Lagrange dual function as

$$g(\{\alpha_i\}) = \min_{\tilde{\mathbf{w}}, w^{(0)}} L(\tilde{\mathbf{w}}, w^{(0)}, \{\alpha_i\})$$

which, as mentioned before, is a concave function that provides a lower bound to the original problem

- Because the SVM problem is convex, this lower bound is tight, so we can obtain its optimal solution by solving

$$\max_{\{\alpha_i\}} g(\{\alpha_i\})$$

$$\text{subject to} \;\; \alpha_i \geq 0 \;, \;\; \forall i$$

Lagrange dual problem

- To solve this Lagrange dual problem, we first need to get the solution to the inner minimization

$$\min_{\tilde{\mathbf{w}}, w^{(0)}} L(\tilde{\mathbf{w}}, w^{(0)}, \{\alpha_i\}) = \min_{\tilde{\mathbf{w}}, w^{(0)}} \left(\frac{1}{2}\|\tilde{\mathbf{w}}\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)}) - 1 \right) \right)$$

- This can be achieved by zeroing out the derivatives of $L(\,\cdot\,)$ w.r.t. $\tilde{\mathbf{w}}$ and $w^{(0)}$

Derivative w.r.t. $\tilde{\mathbf{w}}$

- We have

$$\nabla_{\tilde{\mathbf{w}}} L = \tilde{\mathbf{w}} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

- Setting the derivative to 0 gives

$$\tilde{\mathbf{w}} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

Derivative w.r.t. w_0

- We have

$$\frac{\partial L}{\partial w^{(0)}} = -\sum_{i=1}^{N} \alpha_i y_i$$

- Setting the derivative to 0 gives the condition

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Lagrange dual problem

- By re-inserting the solution for $\tilde{\mathbf{w}}$ in the Lagrangian, we eventually get the dual problem

$$\max_{\{\alpha_i\}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \; , \;\; \forall i$$

Condition from the derivative w.r.t. $w^{(0)}$

Standard condition on the Lagrange multipliers for inequality constraints

Lagrange dual problem

- The dual problem is also a quadratic program
  - Note that it has N variables instead of $D+1$ in the primal formulation
- Because it is a concave maximization problem, it can be solved to optimality
- At the solution, it can be shown that

$$\alpha_i^* \left(y_i \cdot ((\tilde{\mathbf{w}}^*)^T \mathbf{x}_i + w^{(0)*}) - 1\right) = 0$$

- This means that, for every sample, either $\alpha_i^* = 0$, or $y_i \cdot ((\tilde{\mathbf{w}}^*)^T \mathbf{x}_i + w^{(0)*}) = 1$
  - By definition, a point that satisfies the latter constraint is at a margin 1 from the decision boundary, and is thus a *support vector*

Prediction

- Once the optimal solution $\{\alpha_i^*\}$ is obtained, one can use them to predict the label for a new sample $\mathbf{x}$ as

$$\hat{y}(\mathbf{x}) = (\tilde{\mathbf{w}}^*)^T\mathbf{x} + w^{(0)*} = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i^T \mathbf{x} + w^{(0)*}$$

- Because for all samples that are not support vectors, we have $\alpha_i^* = 0$, the sum can be computed on the support vectors only, i.e., with $\mathcal{S}$ the set of support vector indices,

$$\hat{y}(\mathbf{x}) = \sum_{i \in \mathcal{S}} \alpha_i^* y_i \mathbf{x}_i^T \mathbf{x} + w^{(0)*}$$

- N.B. An attentive observer may have noticed that $w^{(0)}$ disappeared when computing the derivatives of the Lagrangian. To understand how to compute $w^{(0)*}$, I therefore invite the interested reader to look at Section 7.1 in Bishop's book

SVM with slack variables: Lagrangian

- When using slack variables, the Lagrangian becomes

$$L(\mathbf{w}, \{\xi_i\}, \{\alpha_i\}, \underbrace{\{\mu_i\}}_{\substack{\text{New Lagrange}\\ \text{multipliers for}\\ \text{the slack}\\ \text{variables}}}) = \underbrace{\frac{1}{2}\|\tilde{\mathbf{w}}\|^2}_{\substack{\text{Same as}\\ \text{before}}} + \underbrace{C\sum_{i=1}^{N}\xi_i}_{\substack{\text{Regularizer on}\\ \text{the slack}\\ \text{variables}}} - \underbrace{\sum_{i=1}^{N}\alpha_i\left(y_i\cdot(\tilde{\mathbf{w}}^T\mathbf{x}_i + w^{(0)}) - 1 + \xi_i\right)}_{\substack{\text{Similar to}\\ \text{before, with}\\ \text{the additional}\\ \text{slack variable}}} - \underbrace{\sum_{i=1}^{N}\mu_i\xi_i}_{\substack{\text{Constraints on}\\ \text{the slack}\\ \text{variables}}}$$

- Deriving the dual problem then follows a similar strategy to that in the case without slack variables

Lagrange dual problem with slack variables

- Ultimately, the $\{\mu_i\}$ can be suppressed, and we obtain the dual problem

$$\max_{\{\alpha_i\}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to} \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C \,, \quad \forall i$$

It only differs from the previous one in these constraints

Support vectors & prediction with slack variables

- With slack variables, at the solution, we have

$$\alpha_i^* \left(y_i \cdot ((\tilde{\mathbf{w}}^*)^T \mathbf{x}_i + w^{(0)*}) - 1 + \xi_i^* \right) = 0$$

- This means that, for every point, either $\alpha_i^* = 0$, or $y_i \cdot ((\tilde{\mathbf{w}}^*)^T \mathbf{x}_i + w^{(0)*}) = 1 - \xi_i^*$

- The prediction has the same form as before, and the sum can again be computed only on the samples for which $\alpha_i^* \neq 0$, i.e.,

$$\hat{y}(\mathbf{x}) = \sum_{i \in \mathcal{S}} \alpha_i^* y_i \mathbf{x}_i^T \mathbf{x} + w^{(0)*}$$

Slack variables: Optimal values

- By re-writing the contraints, we have

$$\xi_i \geq 1 - y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})$$

- After optimization, because we minimize $C\sum_{i=1}^{N} \xi_i$, we have
    - For the samples i that satisfy the original contraints $y_i \cdot (\tilde{\mathbf{w}*}^T \mathbf{x}_i + w^{(0)*}) \geq 1$, $\xi_i^* = 0$ (because $\xi_i \geq 0$)
    - For the samples i that don't, $\xi_i^* = 1 - y_i \cdot (\tilde{\mathbf{w}*}^T \mathbf{x}_i + w^{(0)*})$

SVM: Loss function

- This observation allows us to re-write the SVM problem as

$$\min_{\tilde{\mathbf{w}}, w^{(0)}} \frac{1}{2C} \|\tilde{\mathbf{w}}\|^2 + \sum_{i=1}^{N} \max\left(0, 1 - y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})\right)$$

- The term

$$\max\left(0, 1 - y_i \cdot (\tilde{\mathbf{w}}^T \mathbf{x}_i + w^{(0)})\right)$$

is referred to as the *hinge loss*

Linear models

- We have seen 3 ways to use a linear model for classification:
  - Least-square classification
  - Logistic regression
  - Support Vector Machine

- All three methods rely on the **same** linear model, yet they yield different classifiers
- Question: Why? What is the main difference between them?