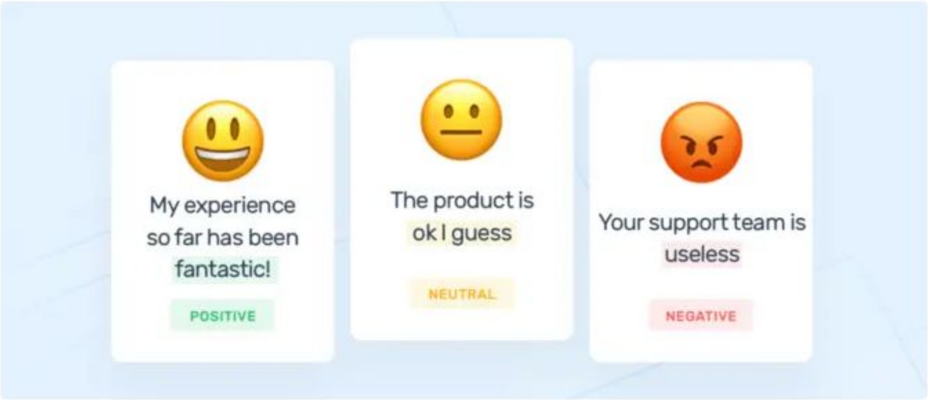
Sentiment Analysis

Basic Concepts and SVM

Basic concepts

What is Sentiment Analysis?

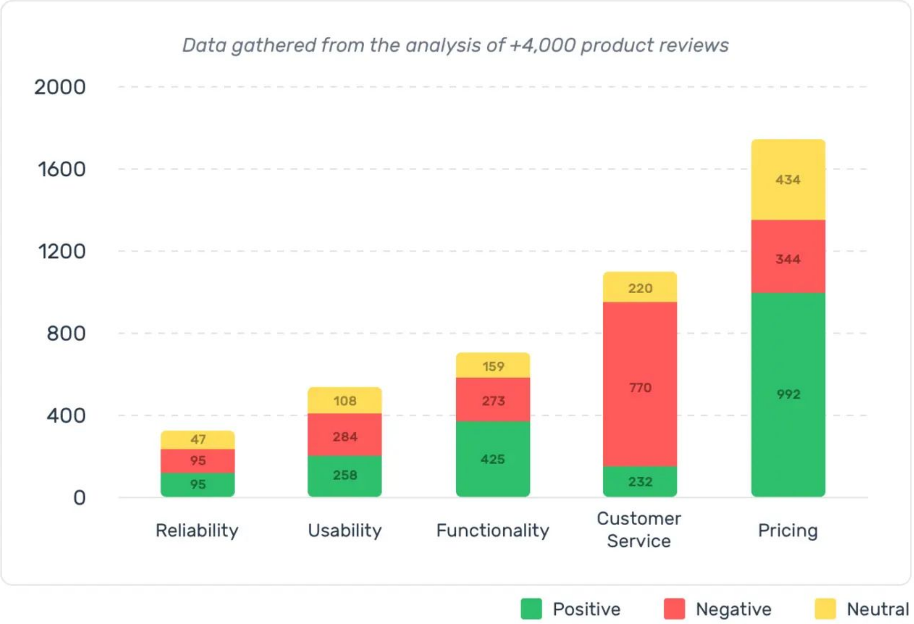
The practice of applying **Natural Language Processing** and **Text Analysis** techniques to identify *polarity* within a text (e.g. a *positive* or *negative* or *neutral* opinion), where it can be a whole document, paragraph, or sentence.



Source: Monkeylearn

One example

Automatically analyze 4,000+ reviews about a product, and discovered that customers were happy about their `pricing` but complained a lot about their `customer service`:



Source: Monkeylearn

General applications of sentiment analysis?

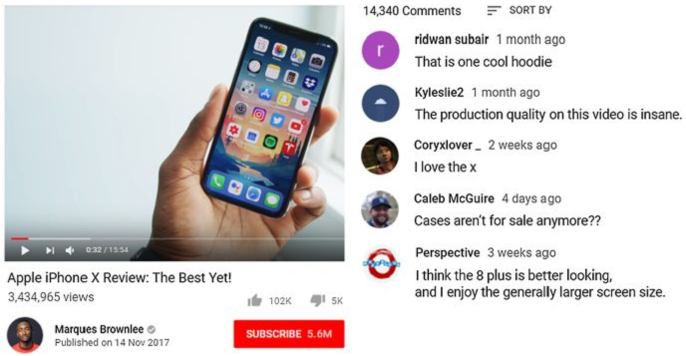
- *Movie*: is this review positive or negative?

Positive	**Negative**
GREAT movie and the family will love it!! If kids are bored one day just pop the tape in and you'll be so glad you did!!! ~~~Rube i luv raven-s!	The script for this movie was probably found in a hair-ball recently coughed up by a really old dog. Mostly an amateur film with lame FX. For you Zeta-Jones fanatics: she has the credibility of one Mr. Binks.
Did Sandra (yes, she must have) know we would still be here for her some nine years later? See it if you haven't, again if you have; see her live while you can.	I would love to have that two hours of my life back. It seemed to be several clips from Steve's Animal Planet series that was spliced into a loosely constructed script. Don't Go, If you must see it, wait for the video ...
Verry classic plot but a verry fun horror movie for home movie party Really gore in the second part This movie proves that you can make something fun with a small budget. I hope that the director will make another one	This is without a doubt the worst movie I have ever seen. It is not funny. It is not interesting and should not have been made.

Source: Towards Data Science

General applications of sentiment analysis?

- *Products*: what do people think about the new iPhone?

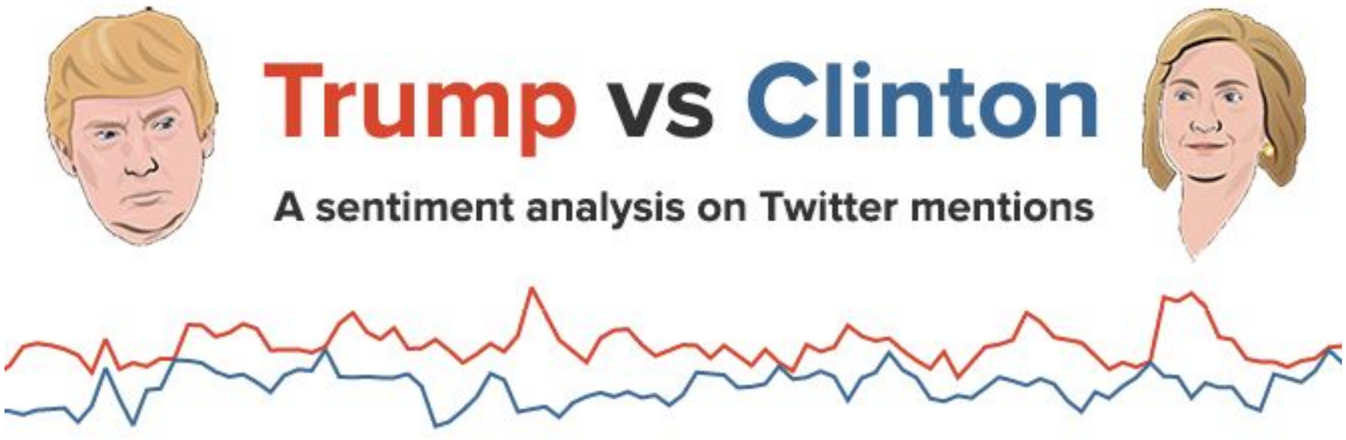


Source: Springer

- *Public sentiment*: how is consumer confidence? Is despair increasing?
  - "**Consumer confidence** is an economic indicator which measures the degree of optimism that **consumers** feel about the overall state of the economy and their personal financial situation." (from *Wikipedia*)

General applications of sentiment analysis?

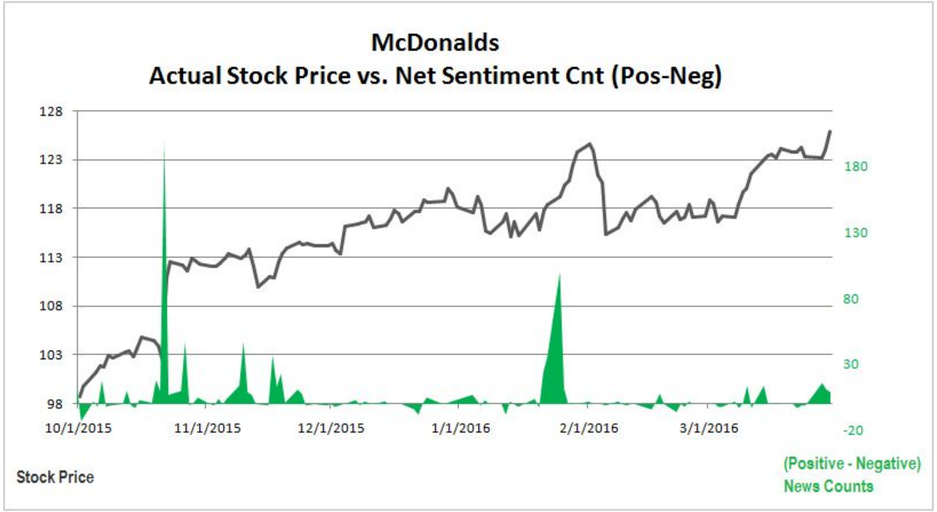
- *Politics*: what do people think about this candidate or issue?



Source: Monkeylearn

General applications of sentiment analysis?

- *Prediction*:
  - predict election outcomes or market trends from sentiment
  - predict stock prices (up and down) with sentiment analysis of user generated content



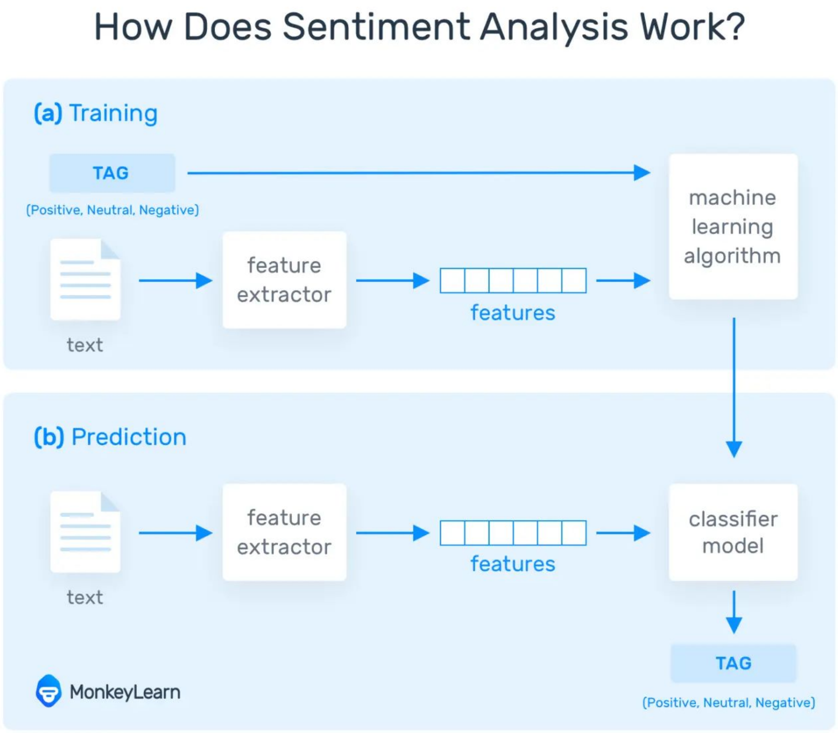
Source: Printerest

Types of Sentiment Analysis

- Polarity (positive, negative, neutral)
- Feelings and emotions (e.g. angry, happy, sad, etc)
- Intentions (e.g. *interested* v. *not interested*)

How it works?

- Input: Text
- Feature Extractor:
  - Bag-of-Words
  - Word embedding
- Classification
  - Logistic Regression
  - Naive Bayes
  - Decision Tree
  - Support Vector Machine



Sentiment Analysis Challenges

Subjective and Tone

What did you like about the event?
What did you DISlike about the event?

Context — *Absolutely nothing!*

Irony — *Yeah, sure. So smooth!*

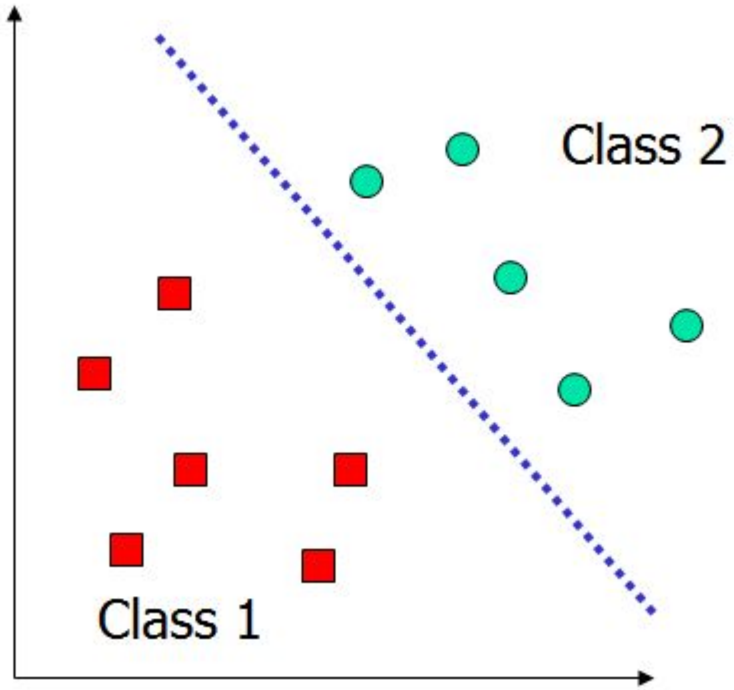
Comparisons — *This is better than older tools.*

Emojis — 😂

Defining Neutral

Support Vector Machine

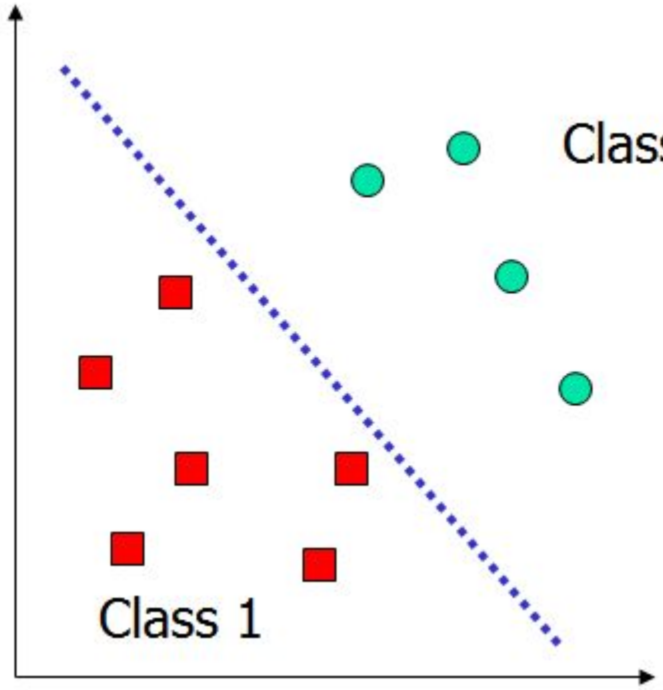
Two Class Problem: Linear Separable Case

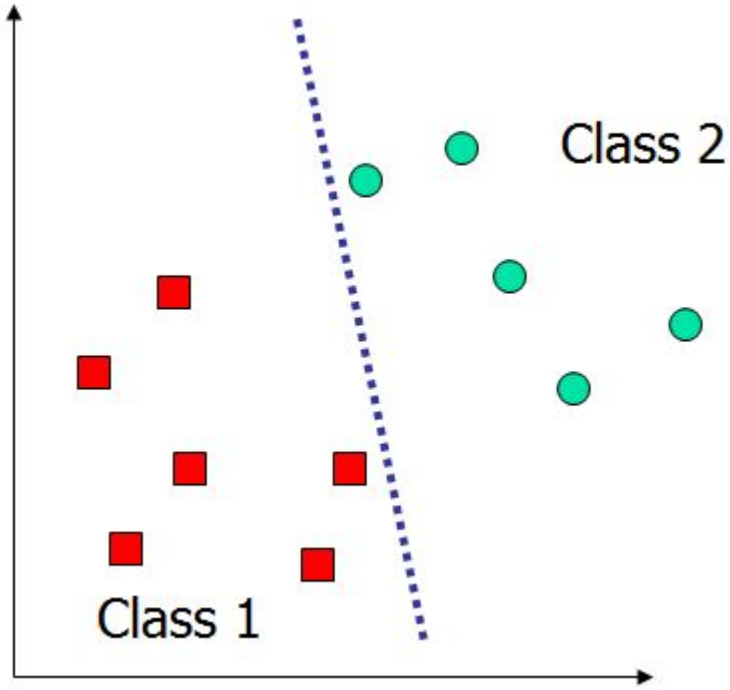


Many decision boundaries can separate these two classes

Which one should we choose?

Example of Bad Decision Boundaries

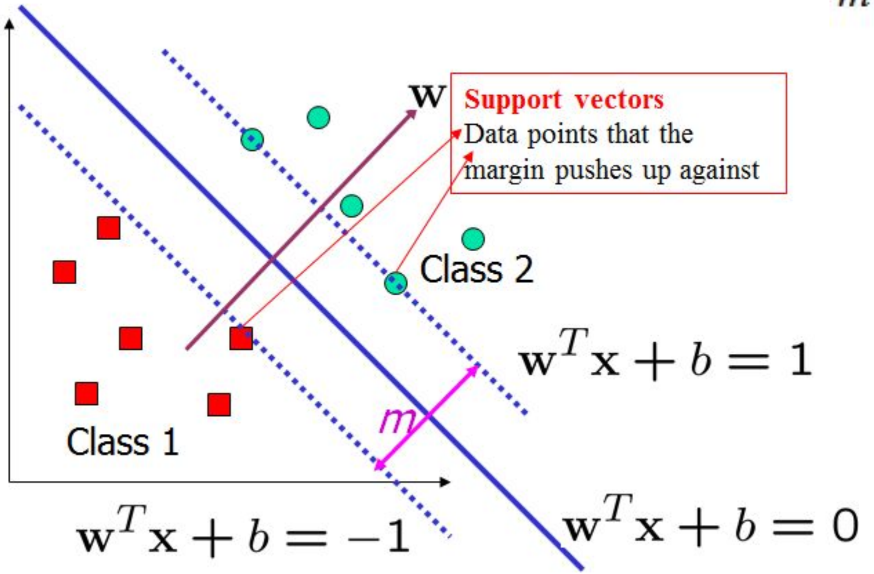




Good Decision Boundary: Margin Should Be Large

The decision boundary should be as far away from the data of both classes as possible

$$m = \frac{2}{\sqrt{w.w}} \qquad m = \frac{2}{||\mathbf{w}||}$$



We should maximize the margin, *m*

The maximum margin linear classifier is the linear classifier with the maximum margin.
This is the simplest kind of SVM (Called an Linear SVM)

The Optimization Problem

Let $\{x_1, ..., x_n\}$ be our data set and let y_i $\{1,-1\}$ be the class label of x_i

The decision boundary should classify all points correctly

A constrained optimization problem

$$m = \frac{2}{||\mathbf{w}||}$$

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

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

Lagrangian of Original Problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to } 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0 \qquad \text{for } i = 1, \ldots, n$$

The Lagrangian is

Lagrangian multipliers

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n} \alpha_i \left(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

Setting the gradient of *L* w.r.t. **w** and b to zero, we have

$$\mathbf{w} + \sum_{i=1}^{n} \alpha_i(-y_i)\mathbf{x}_i = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \alpha_i \geq 0$$

The Dual Optimization Problem

We can transform the problem to its dual

Dot product of X

$$\max.\ W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to $\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$

α's → New variables
(Lagrangian multipliers)

KKT:

$$y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 \geq 0 \qquad i = 1, \ldots, n$$
$$a_i \geq 0 \qquad \forall i$$
$$a_i(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) = 0 \qquad \forall i$$

This is a convex quadratic programming (QP) problem

- Global maximum of a_i can always be found
- Well established tools for solving this optimization problem (e.g. cplex)

Primal and Dual Problems

Assume N is the number of training samples, and d is the dimension of the data

Primal

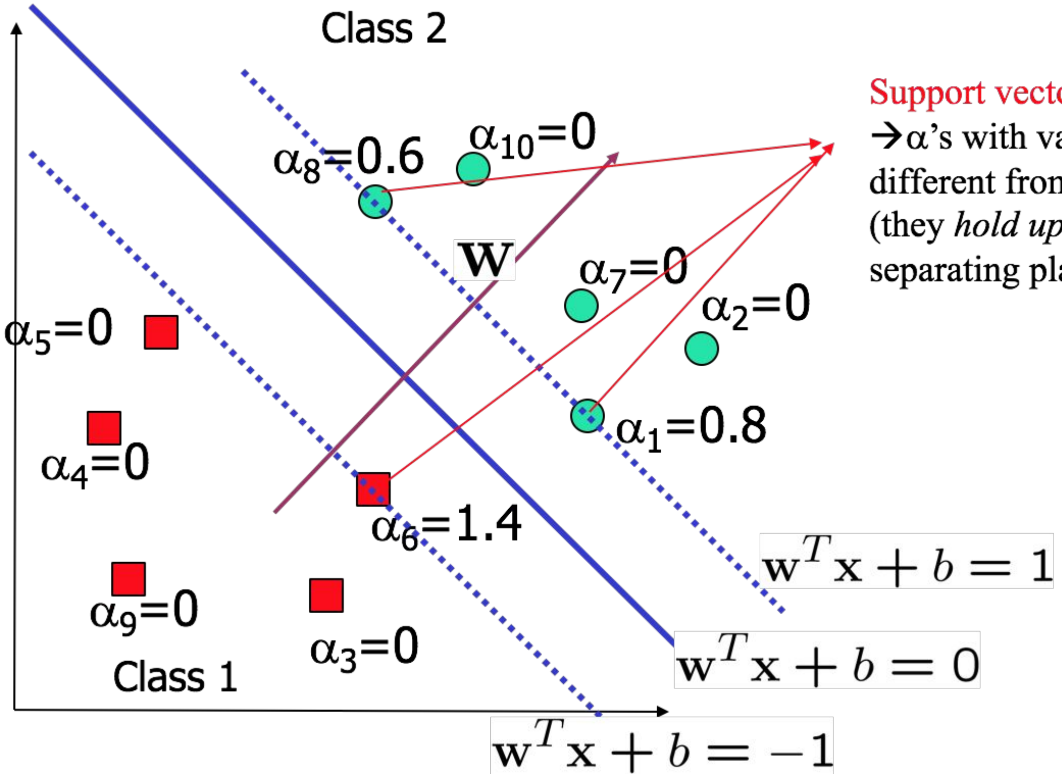
$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$
$$\text{subject to } 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b) \leq 0 \qquad \text{for } i = 1, \ldots, n$$

Dual

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$
$$\text{subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Need to learn d parameters for primal and N for dual
- If N<<d then more efficient to solve for dual
- Dual form only involves dot product of x. We will return to why this is an advantage when we look at *kernels*

A Geometrical Interpretation

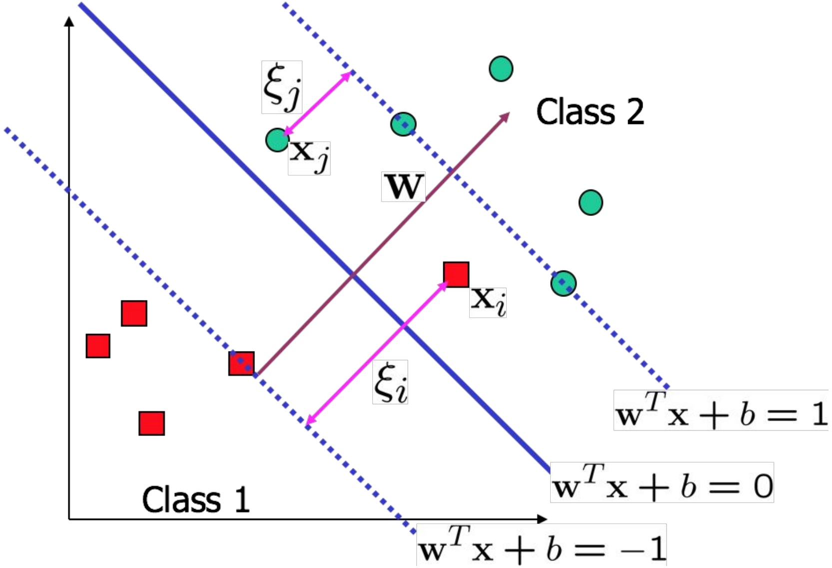


Support vectors
→α's with values different from zero (they *hold up* the separating plane)!

Non-linearly Separable Problems

We allow “error” in classification; it is based on the output of the discriminant function wx+b

Approximates the number of misclassified samples



New objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

C : tradeoff parameter between error and margin;
chosen by the user;
large C means a higher penalty to errors

The Optimization Problem

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

The only difference with the linear separable case is that there is an upper bound C on a_i

Once again, a QP solver can be used to find a_i efficiently!

Extension to Non-linear SVMs (Kernel Methods)

Non-Linear SVM

How could we generalize this procedure to non-linear data?

Vapnik in 1992 showed that transforming input data **x_i** into a higher dimensional makes the problem easier.

Similar to Hidden Layers in ANN

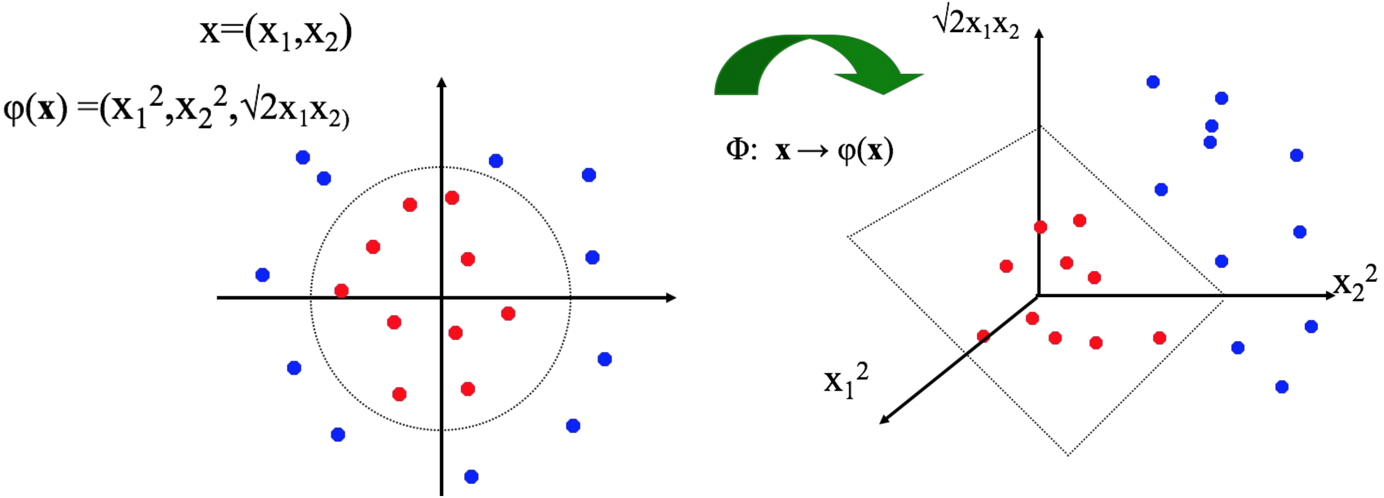
- We know that data appears only as dot products (**x_i, x_j**)
- Suppose we transform the data to some (possibly infinite dimensional) space **H** via a mapping function Φ such that the data appears of the form Φ**(x_i)**Φ(**x_j**)

Why?

- Linear operation in **H** is equivalent to non-linear operation in input space.

Non-linear SVMs: Feature Space

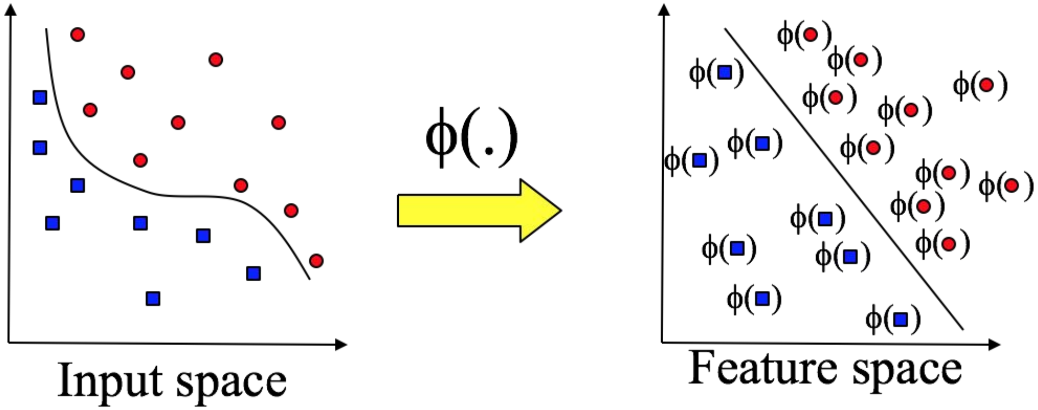
General idea: the original input space (x) can be mapped to some higher-dimensional feature space ($\varphi(\mathbf{x})$)where the training set is separable:



If data are mapped into higher a space of sufficiently high dimension, then they will in general be linearly separable; N data points are in general separable in a space of N-1 dimensions or more!!!

Transformation to Feature Space

- Possible problem of the transformation
  - High computation burden due to high-dimensionality and hard to get a good estimate
- SVM solves these two issues simultaneously
  - "Kernel tricks" for efficient computation
  - Minimize $||\mathbf{w}||^2$ can lead to a "good" classifier



Kernel Trick

Recall:

Note that data only appears as dot products

Minimize $\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=j=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j$

Subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i y_i = 0$

Since data is only represented as dot products, we need not do the mapping explicitly.

Introduce a Kernel Function (*) *K* such that:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

(*) Kernel function – a function that can be applied to pairs of input data to evaluate dot products in some corresponding feature space

Example Transformation

Consider the following transformation

$$\phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2)$$

Define the kernel function *K* (**x**,**y**) as

$$\begin{aligned} \langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle &= (1 + x_1y_1 + x_2y_2)^2 \\ &= K(\mathbf{x}, \mathbf{y}) \end{aligned}$$

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

The inner product can be computed by *K* without going through the map $\phi(\cdot)$ explicitly!!!

Modification Due to Kernel Function

Change all inner products to kernel functions,

Original

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

With Kernel

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

Examples of Kernel Functions

- Polynomial kernel with degree *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Radial basis function kernel

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

- Hyperbolic tangent kernel

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

- Research on different kernel functions in different applications is very active

Example

- Suppose we have 5 1D data points
  - $x_1=1$, $x_2=2$, $x_3=4$, $x_4=5$, $x_5=6$, with 1, 2, 6 as class 1 and 4, 5 as class 2, $y_1=1$, $y_2=1$, $y_3=-1$, $y_4=-1$, $y_5=1$
- We use the polynomial kernel of degree 2
  - $K(x,y) = (xy+1)^2$
  - C is set to 100
- We first find a_i (i=1, …, 5) by

$$\text{max.} \quad \sum_{i=1}^{5} \alpha_i - \frac{1}{2} \sum_{i=1}^{5} \sum_{j=1}^{5} \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2$$

$$\text{subject to } 100 \geq \alpha_i \geq 0, \sum_{i=1}^{5} \alpha_i y_i = 0$$

Example

- By using a QP solver, we get

  a_1=0, a_2=2.5, a_3=0, a_4=7.333, a_5=4.833

  - Verify (at home) that the constraints are indeed satisfied
  - The support vectors are {x_2=2, x_4=5, x_5=6}
- The discriminant function is

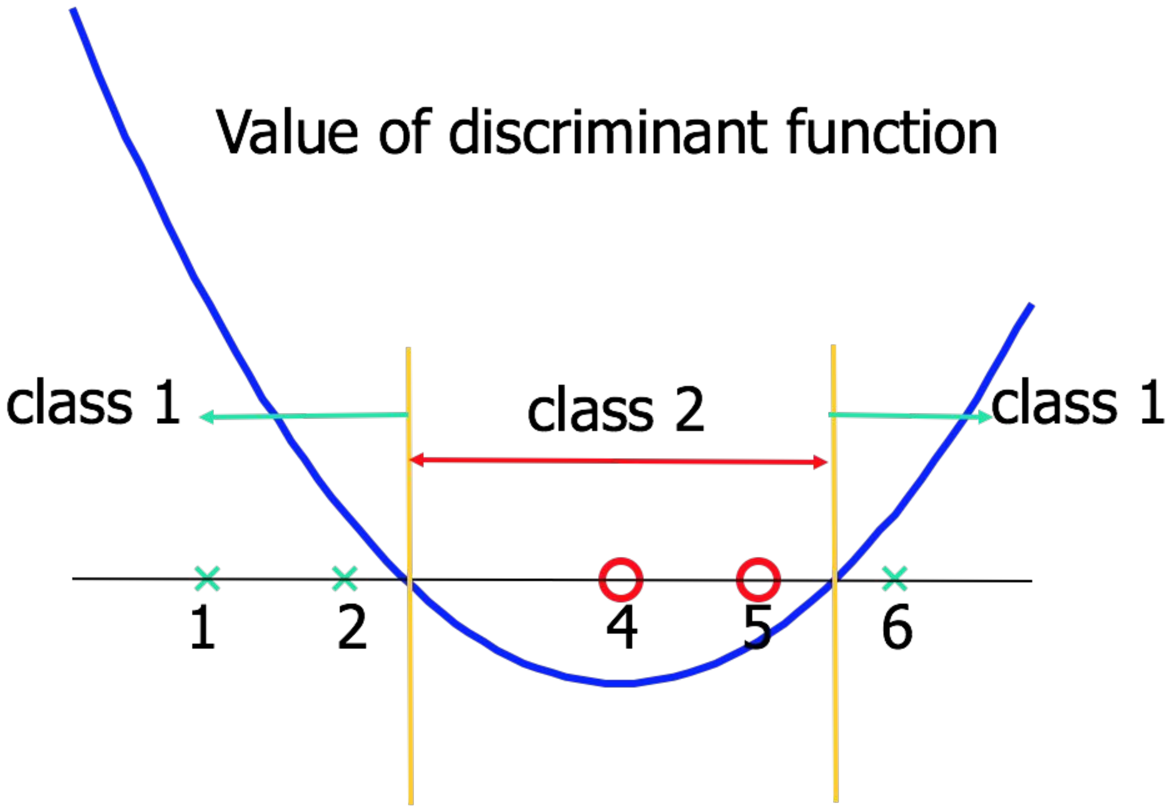
$$f(y) = 2.5(1)(2y+1)^2 + 7.333(-1)(5y+1)^2 + 4.833(1)(6y+1)^2 + b$$
$$= 0.6667x^2 - 5.333x + b$$

b is recovered by solving f(2)=1 or by f(5)=-1 or by f(6)=1, as x_2, x_4, x_5 lie on and all give b=9

$$y_i(\mathbf{w}^T \phi(z) + b) = 1$$

$\Longrightarrow$ $$f(y) = 0.6667x^2 - 5.333x + 9$$

Example



Value of discriminant function
class 1
class 2
class 1
1
2
4
5
6

Weaknesses

- Training (and Testing) is quite slow compared to ANN
  - Because of Constrained Quadratic Programming
- Essentially a binary classifier
  - However, there are some tricks to evade this.
- Very sensitive to noise
  - A few off data points can completely throw off the algorithm
- Biggest Drawback: The choice of Kernel function.
  - There is no “set-in-stone” theory for choosing a kernel function for any given problem (still in research...)

Strengths

- Training is relatively easy
  - We don’t have to deal with local minimum like in ANN.
  - SVM solution is always global and unique.
- Less prone to overfitting
- Simple, easy to understand geometric interpretation.
  - No large networks to mess around with.

SVM for sentiment analysis

High dimensional features, since they can have up to one for every word that appears in the training data.

Using nonlinear kernels may be a good idea in other cases, having this many features will end up making nonlinear kernels *overfit* the data.

Linear kernel actually results in the best performance in most of cases.