Revision

DECISION BOUNDARIES

Hyperplane

- A hyperplane is a subspace whose dimension is one less than that of its ambient space.
  - If a space is 3-dimensional, its hyperplanes are the 2-dimensional planes
  - If a space is 2-dimensional, it's hyperplanes are the 1-dimensional lines

$$\boxed{\boldsymbol{a_1 x_1 + a_2 x_2 +, \ldots, a_n x_n = b}}$$

- When the coordinates are real numbers, this hyperplane separates the space into two half-spaces given by the inequalities

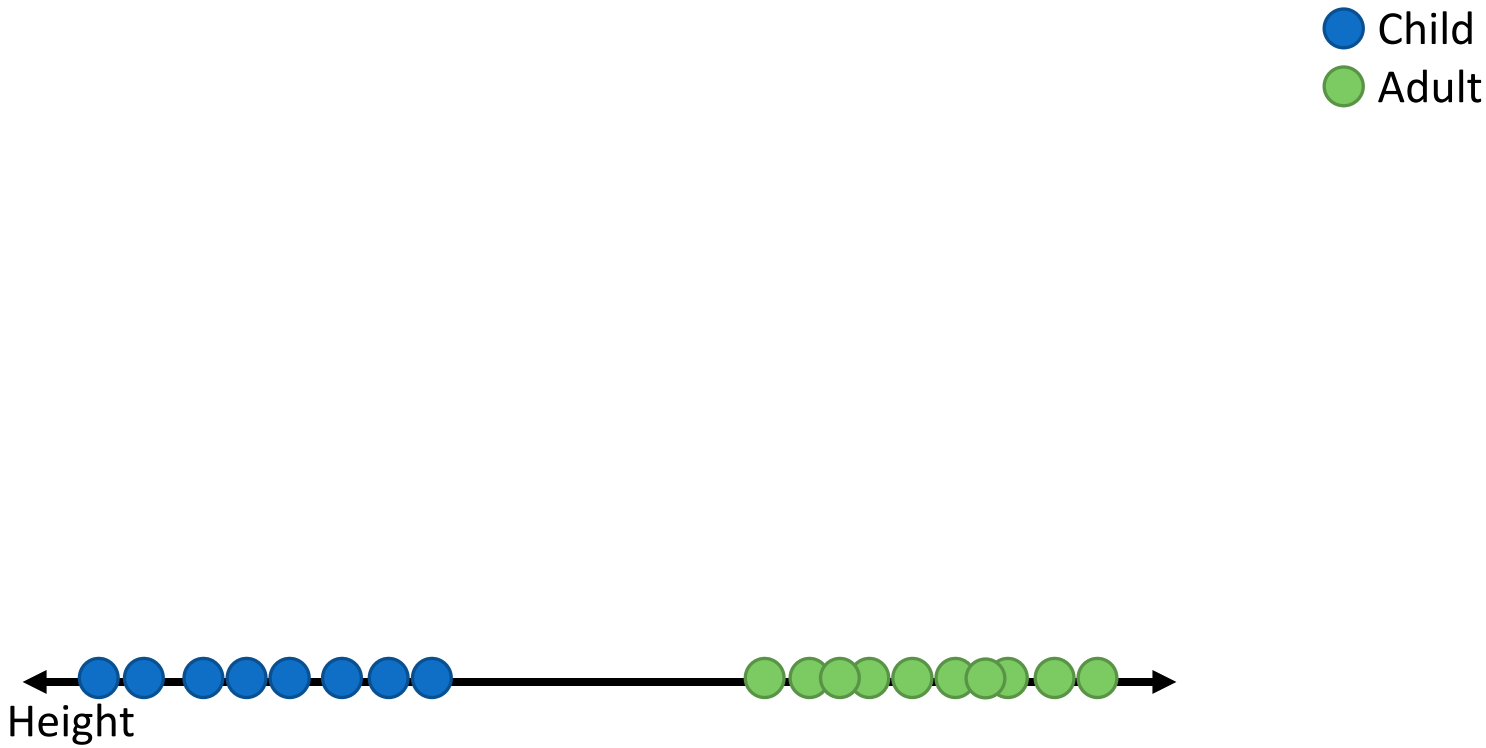
$$\boxed{\boldsymbol{a_1 x_1 + a_2 x_2 +, \ldots, a_n x_n < b}}$$

$$\boxed{\boldsymbol{a_1 x_1 + a_2 x_2 +, \ldots, a_n x_n \geq b}}$$

There could be infinite hyperplanes possible! How to get most optimal hyperplane?

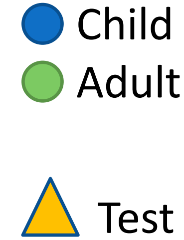
References

- StatQuest by Josh Starmer, Support Vector Machines, Clearly Explained!!!
  - https://www.youtube.com/watch?v=efR1C6CvhmE&t=439s

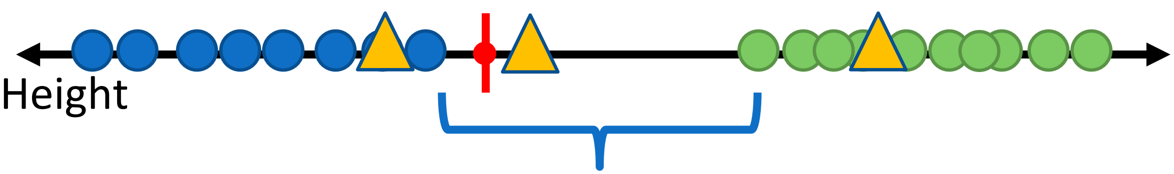


How many dimensions are there in this labeled dataset?

Where should we draw a "boundary" that separates each class based on this dimension?

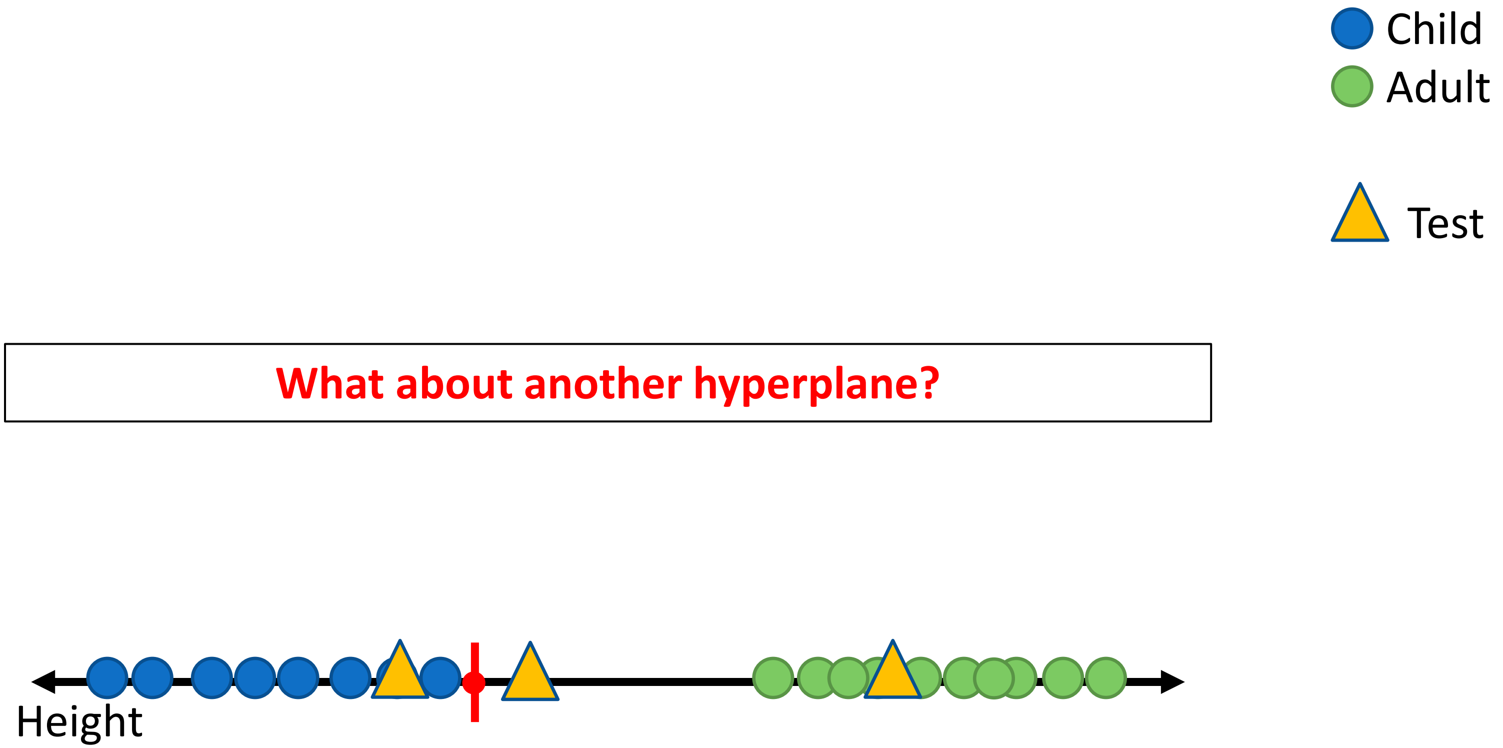


Suppose we have trained a perceptron which separates negative and positive examples. Where that hyperplane could be?



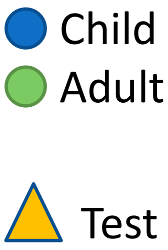
A perceptron can learn to draw a hyperplane anywhere in this region.

What about another hyperplane?



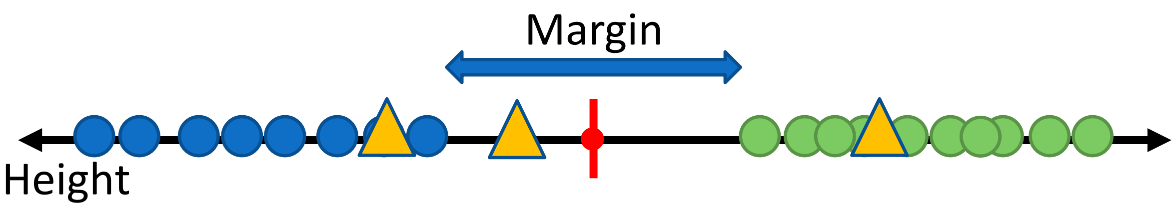
So some hyperplanes are better than others!

But what would be the best hyperplane?



The classifiers that maximize the margin between classes to draw a hyperplane are called "Maximal Margin Classifiers"

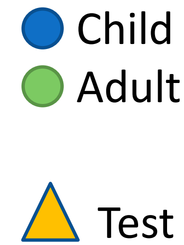
Margin between hyperplane and both classes should be equal and maximum!



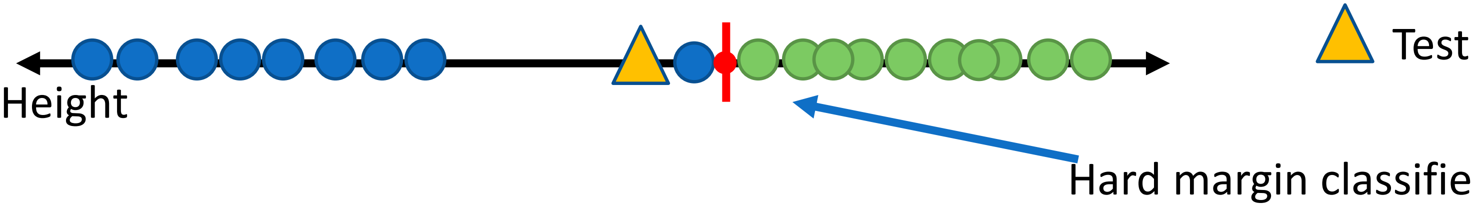
Hyperplane in the center of two known classes is probably the best hyperplane!

But all is not well!

Would bias and variance be high or low for this classifier?

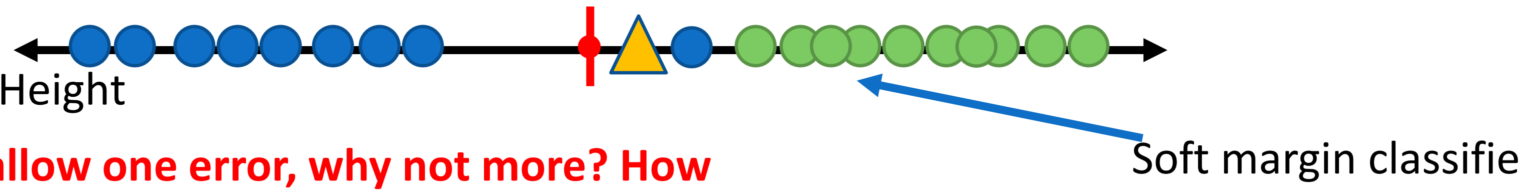


It’s low bias and high variance model!



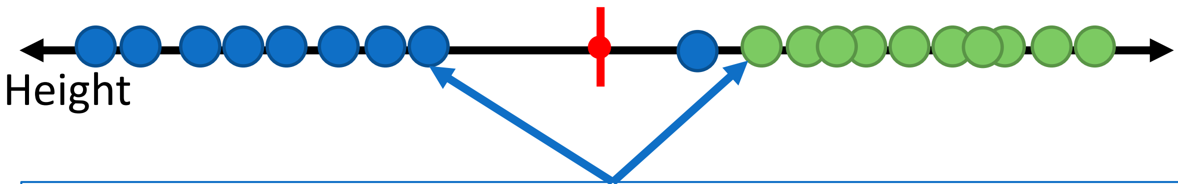
Let’s allow some error for the training examples.

Now it’s high bias and low variance model!



But why just allow one error, why not more? How to decide how much more bias should be added?

Use cross validation!

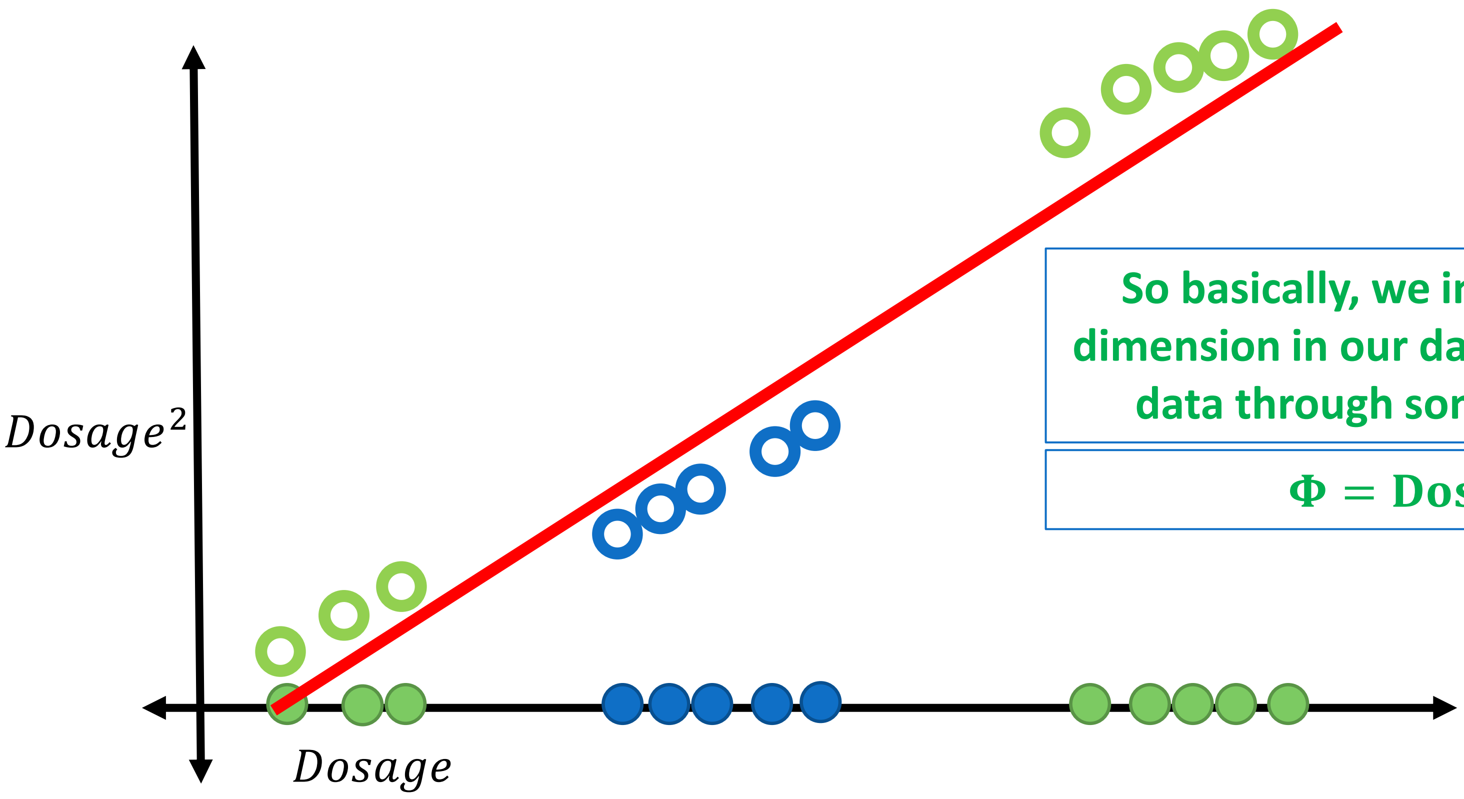


The data points that are considered to draw the hyperplane are known as “support vectors”

Where should we draw a hyperplane to separate both classes now?

Now a linear classifier cannot do this separation!

What to do now?



So basically, we introduced new dimension in our data by passing the data through some function $\boldsymbol{\Phi}$

$$\boldsymbol{\Phi = \text{Dosage}^2}$$

Kernel

A Kernel Function…

Could be simple

- $\Phi(x) = x$
- Called linear kernel

Could be complex

- $\Phi(x) = x_1^2 + x_1^3 + x_3 + \sqrt{x_4} \ldots$

Linear Kernel, Polynomial Kernel, Radial Basis Kernel…

- Use cross validation to find which kernel works best.

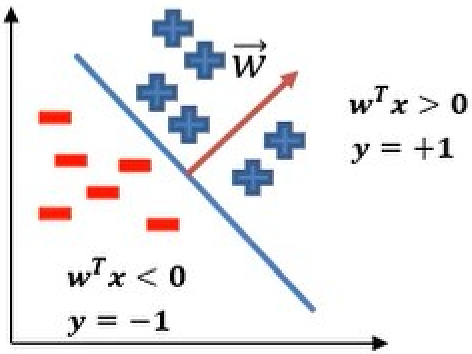
Hard Margin SVM

References

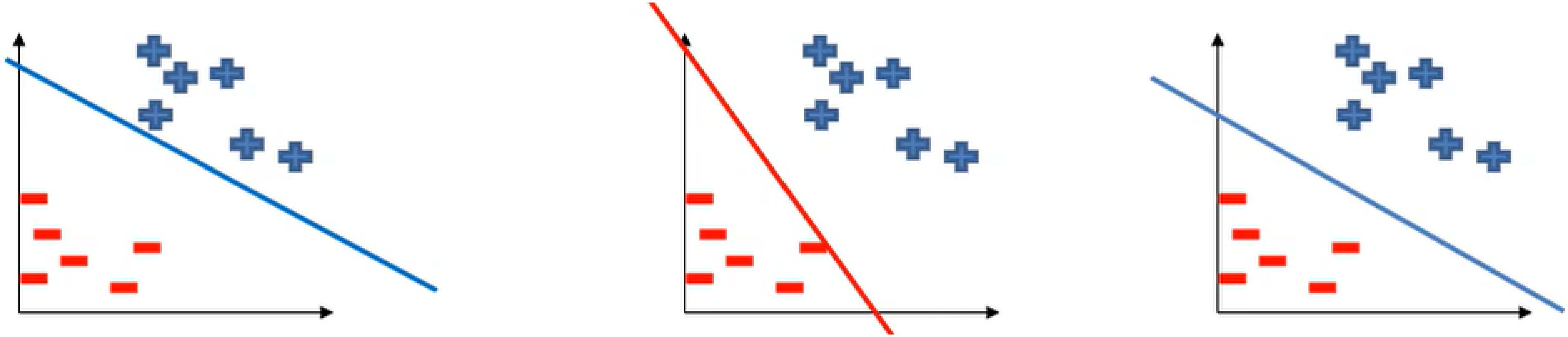
- Machine Learning for Intelligent Systems, Kilian Weinberger, Cornell, Lecture 9,
  - https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote09.html
- Support Vector Machine: how it really works,
  - https://www.youtube.com/watch?v=A7FeQekjd9Q , Victor Lavrenko, Assistant Professor at the University of Edinburgh
- Support Vector Machine Python Example, Cory Maklin,
  - https://towardsdatascience.com/support-vector-machine-pythonexample-d67d9b63f1c8
- Support Vector Machine: Complete Theory, Saptashwa Bhattacharyya,
  - https://towardsdatascience.com/understandingsupport-vector-machine-part-1-lagrange-multipliers-5c24a52ffc5e

Support Vector Machines

The Perceptron



But there are other possibilities as well!



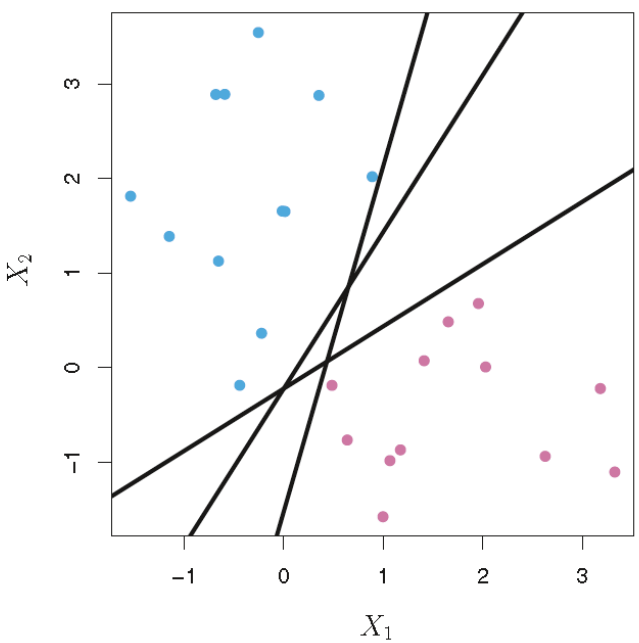
All hyperplanes are perfectly fine for a Perceptron, but not practically!

Support Vector Machines

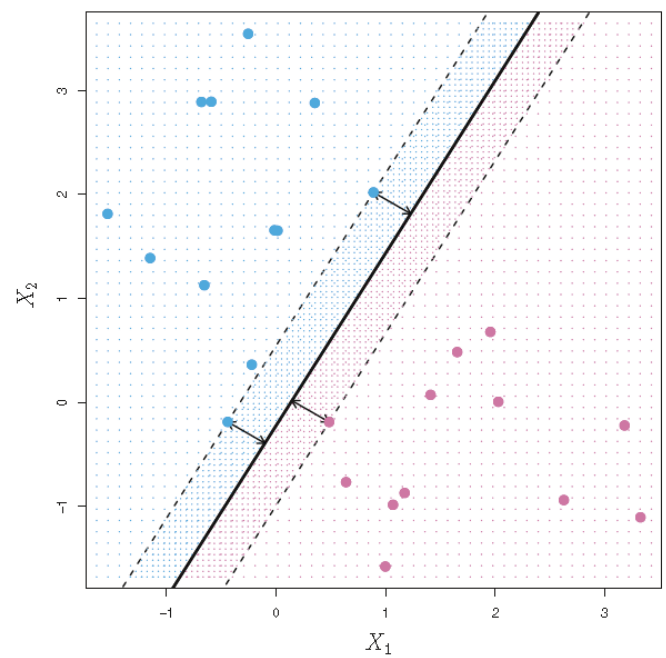
- The Support Vector Machine (SVM) is a linear classifier that can be viewed as an extension of the Perceptron developed by Rosenblattt in 1958.
  - The Perceptron guaranteed that you find a hyperplane if it exists.
  - The SVM finds the **maximum margin separating hyperplane**.

SVM creates road!!

Perceptron



SVM

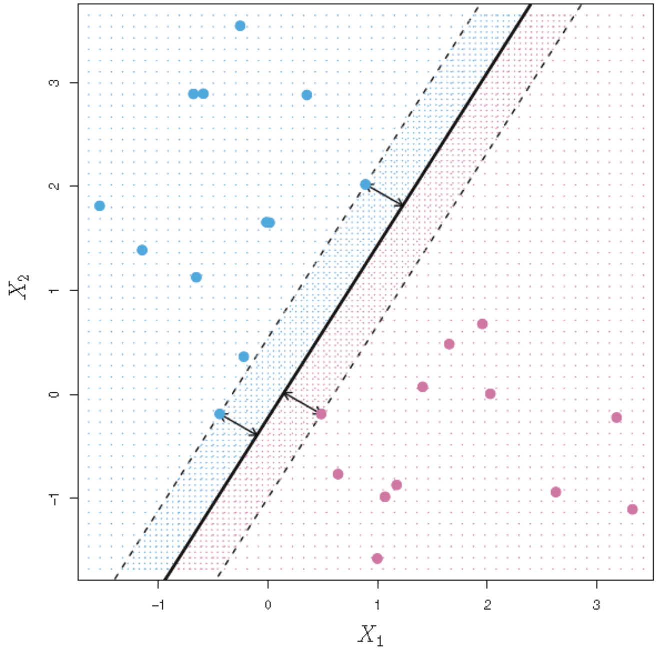


Credit: Machine Learning for Intelligent Systems, Kilian Weinberger, Cornell, Lecture 9
Images Credit: https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f

Support Vector Machines

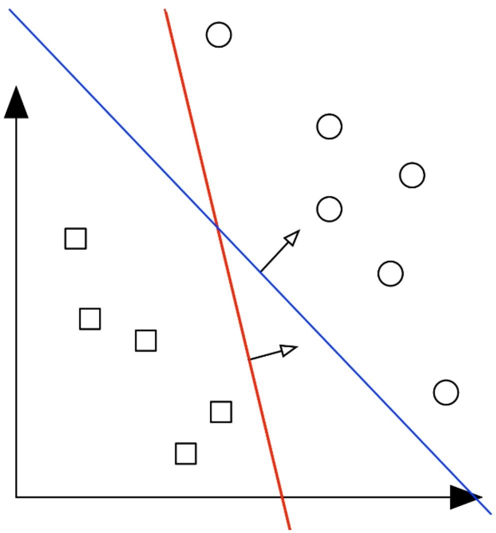
- Support vectors define how wide the “road” could be!
- The margin of hyperplane from the closest support vector of either class is denoted by γ
  - If we have drawn a maximum margin hyperplane, the distance of all γ from that hyperplane is equal!

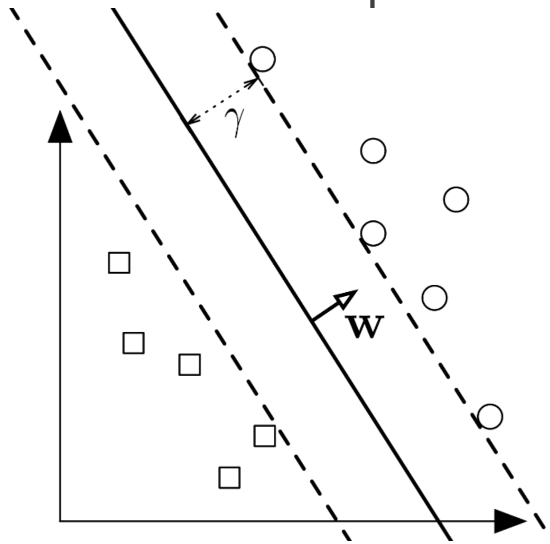
SVM



Support Vector Machines

- **Setting:** We define a linear classifier $h(x) = sign(W^T x + b)$ or and we assume a binary classification setting with labels $y \in \{+1, -1\}$
- **If data is linearly separable,** typically there are infinitely many separating hyperplanes. What is the best separating hyperplane?
- **SVM Answer:** The hyperplane that maximizes the distance to the closest data points from both classes – the hyperplane with **<u>maximum margin</u>**
- A hyperplane is defined as a set of points such that $H = \{x | W^T x + b = 0\}$. The margin γ is the distance from the hyperplane to the closest point across both classes.





Support Vector Machines

- An Optimization Problem!
  - We maximize the margin: the distance separating the closest pair of data points belonging to opposite classes
  - These points are called Support Vectors because they are the data observations that “support” or determine the decision boundary.
    - These are also the samples that are most difficult to classify.

- To train a SVM classifier, we find the **maximum margin hyperplane**, which optimally separates the two classes in order to generalize to new data and make accurate classification predictions.

Support Vector Machines

- Remember that the shortest distance of a point x from a hyperplane H defined by the normal vector $\vec{w}$

$$distance\ (\vec{w}.\vec{x}) = |\vec{w}.\vec{x}| \qquad \text{(if } \vec{w} \text{ is a unit vector)}$$

- More generally

$$distance\ (\vec{w}.\vec{x}) = \frac{|\vec{w}.\vec{x}|}{||\vec{w}||_2} = \frac{|\vec{w}.\vec{x} + b|}{||\vec{w}||_2} = \frac{|\vec{w}^T.\vec{x} + b|}{||\vec{w}||_2} \qquad \text{(if } \vec{w} \text{ is not a unit vector)}$$

Why do this?

- Here for the margin of H will be defined as:

$$\gamma\ (w.b) = \min_{x \in D} \frac{|\vec{w}^T.\vec{x} + b|}{||\vec{w}||_2}$$

- By definition, the margin and hyperplane are scale invariant

$$\gamma\ (\beta w.\beta b) = \gamma(w, b), \forall \beta \neq 0$$

  - Stretch $\vec{w}$ as much as you like, the plane will stay the same
  - Increasing decreasing b will just move the plane around in n-dims

Support Vector Machines

- **Note:** When γ is maximized, the hyperplane must lie right in the middle of the two classes.
  - Like a two-lane road!

- γ must be the distance to the closest point within both classes.
  - If not, the hyperplane could be moved towards data points of the class, that is further away and increase γ, which contradicts that γ is maximized!

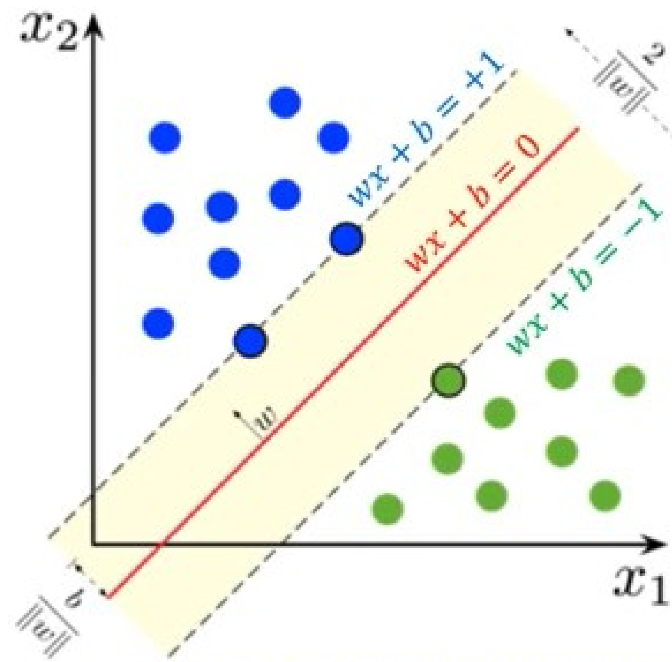


Image ref: https://en.wikipedia.org/wiki/Support-vector_machine

Book Reading

- Murphy – Chapter 1, Chapter 14