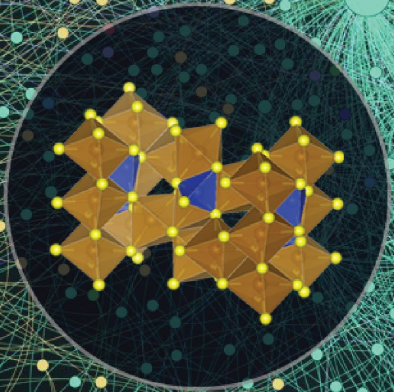
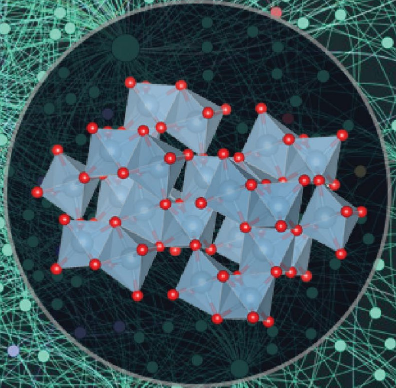
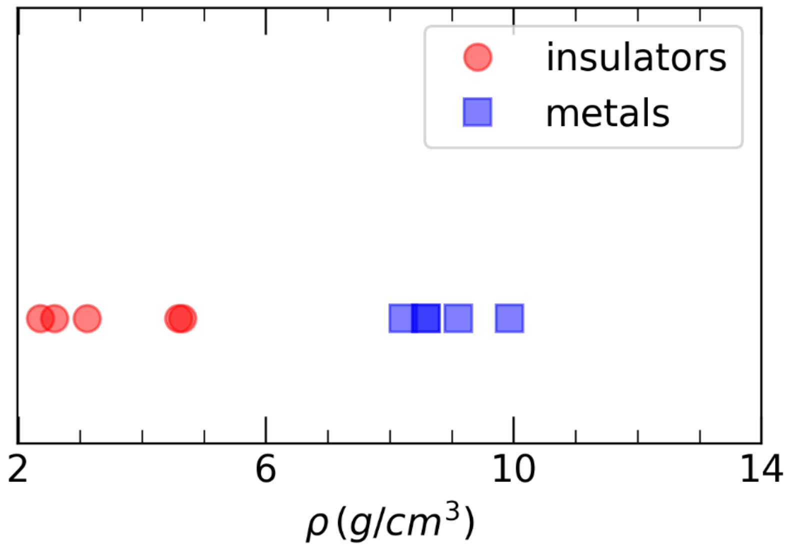
Support Vector Machines (SVMs)

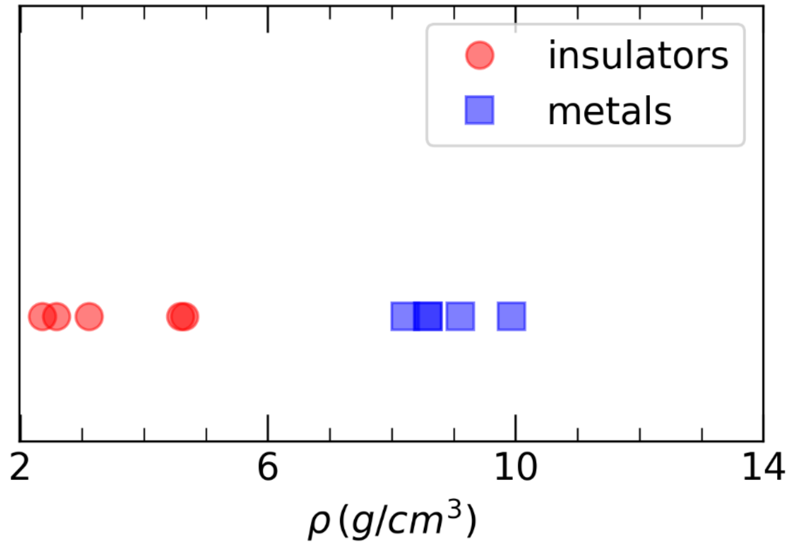



Consider a number line with different categories of data plotted.



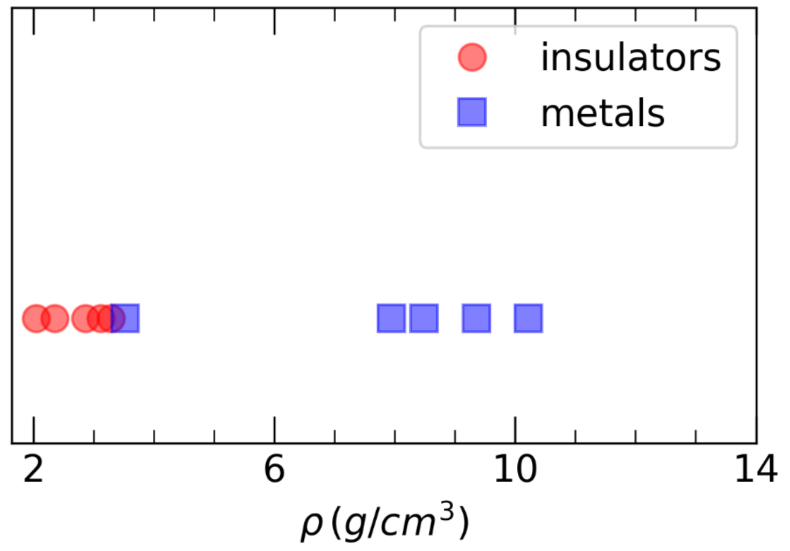
We could select a threshold to classify data into one category or another

Distance from nearest observation to the threshold is the **margin**

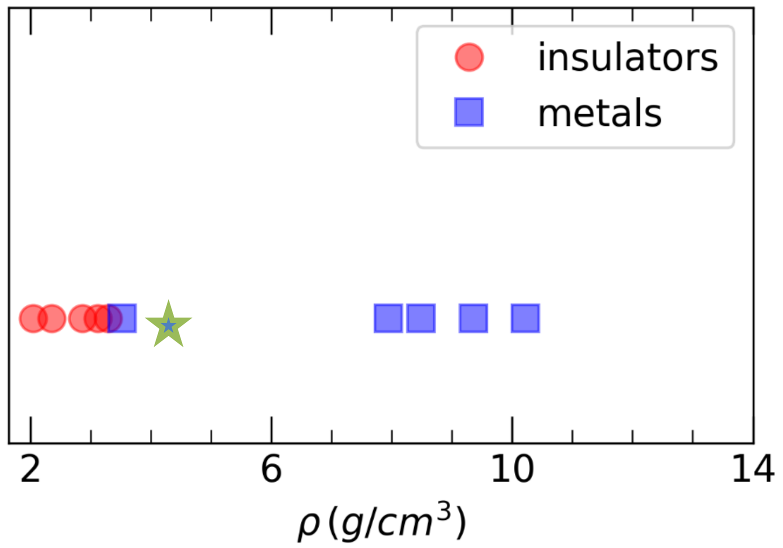


"maximal margin classifier"

Outliers will really mess up a maximal margin classifier

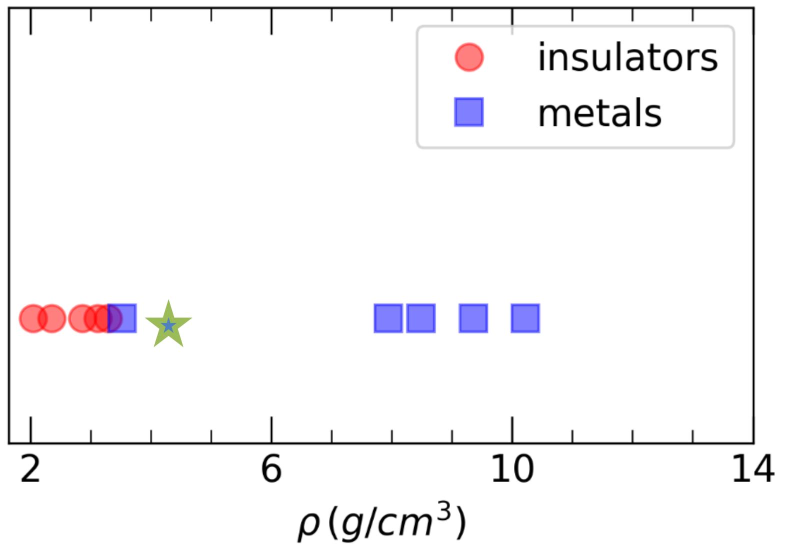


Green star would get classified as insulator…. Is this reasonable?



The only way we perform improvement overall is to allow misclassifications!

Yet another example of tradeoff between variance and bias!

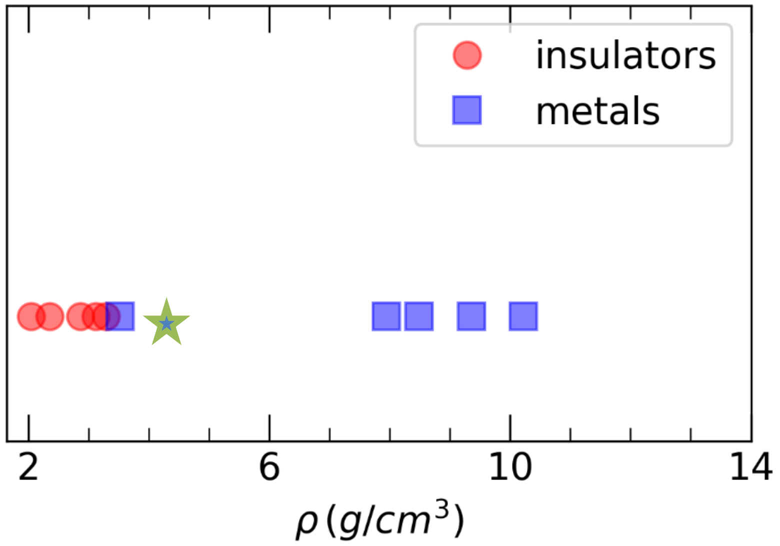


Threshold made from outliers is low

Bias, high variance

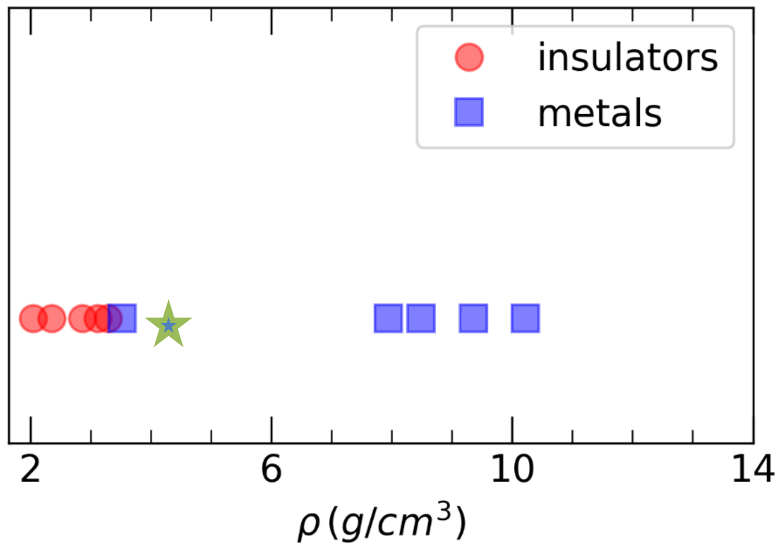
Soft margins is distance from threshold to observations ignoring misclassifications

Which observation to use for soft margin is determined via cross-validation

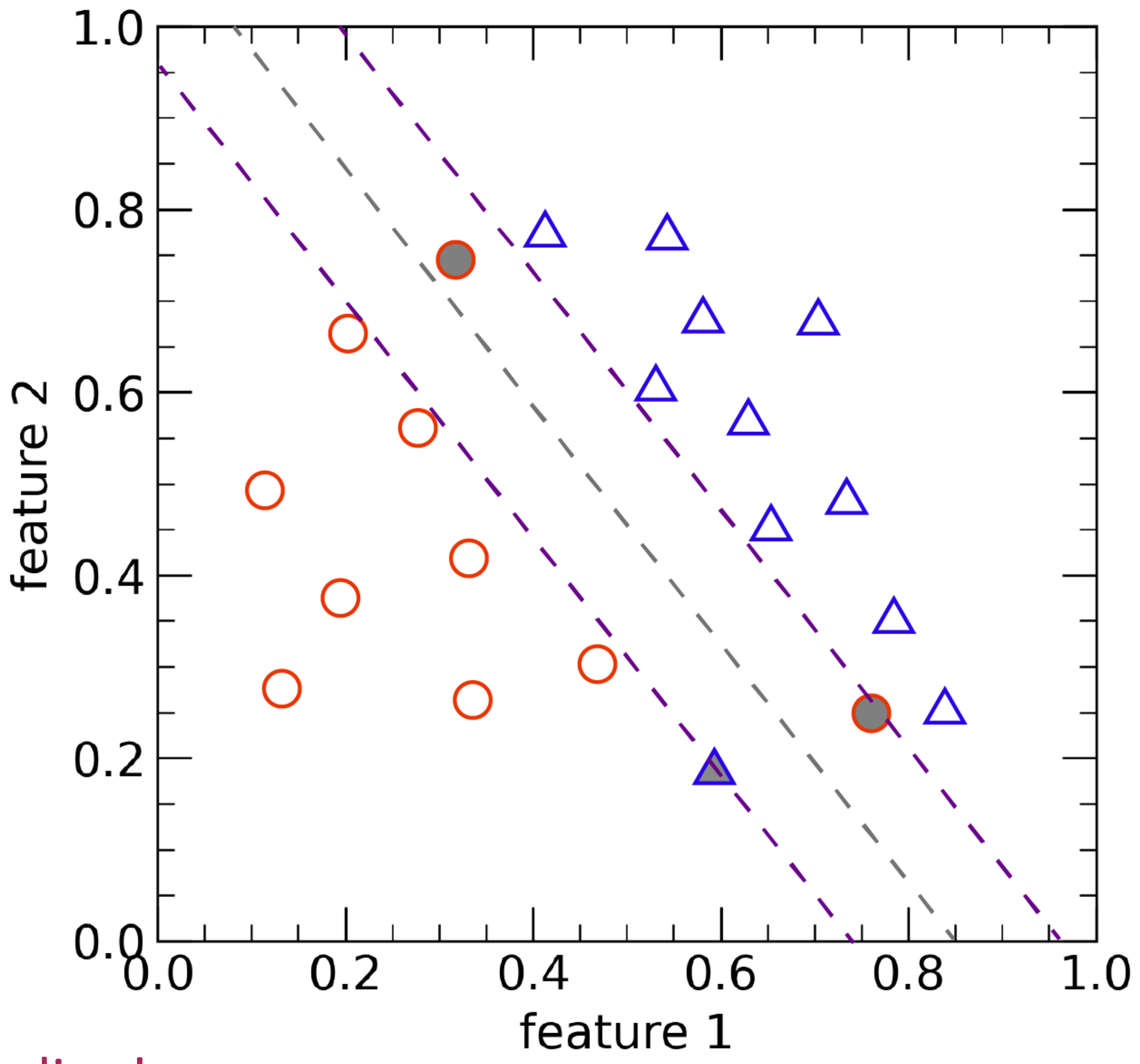


Count misclassifications and observations within soft margin until we get best classification overall

Using soft margin classifiers is also called a **support vector classifier**



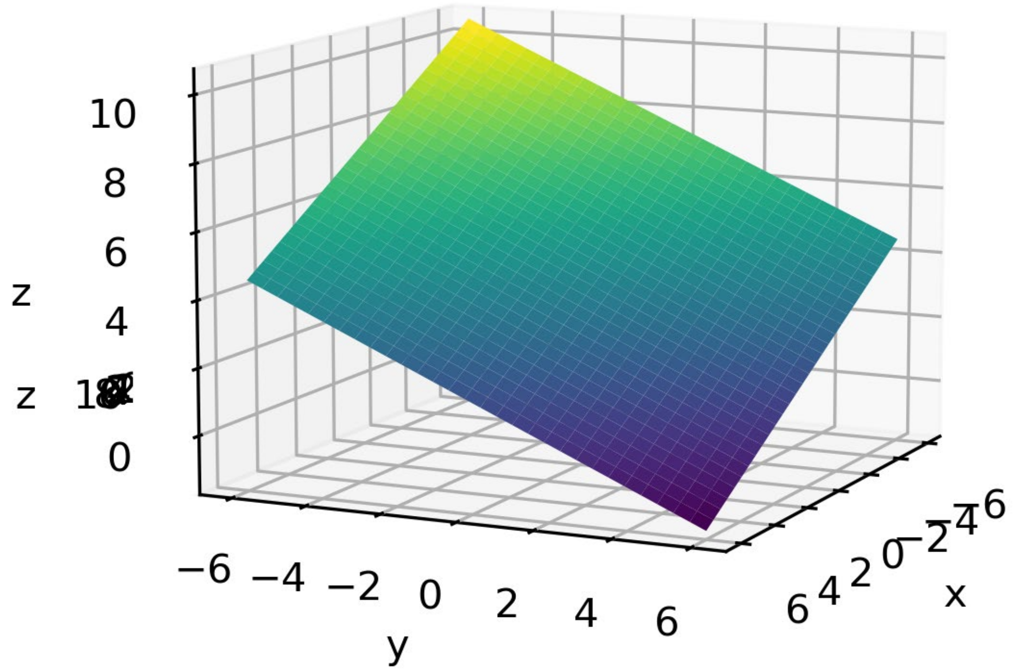
Instead of just 1 dimensional data, we could have higher order data



Soft margin becomes a line!

3 dimensional data would be a plane, but the same principle holds with higher dimensions

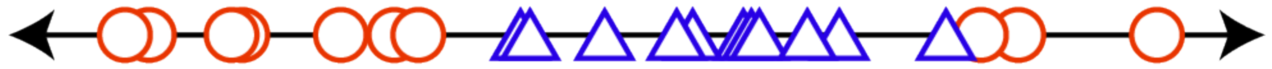
Classifier becomes a plane, instead of a line



Higher order dimensions would use a “hyperplane”
of features-1 dimension

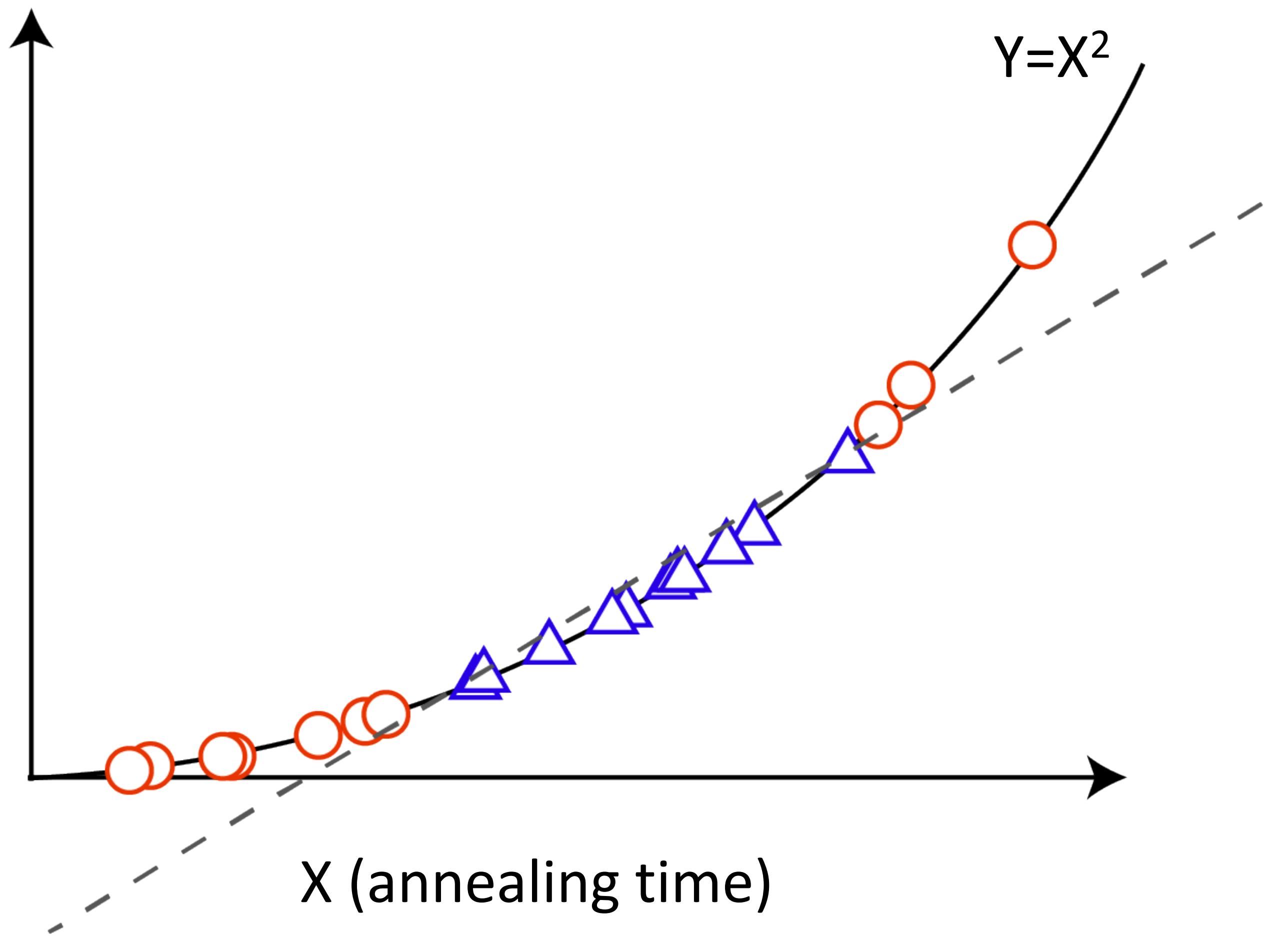
Sometimes our data can't be split nicely with any margin choice

Annealing time on x axis, phase of interest as classification

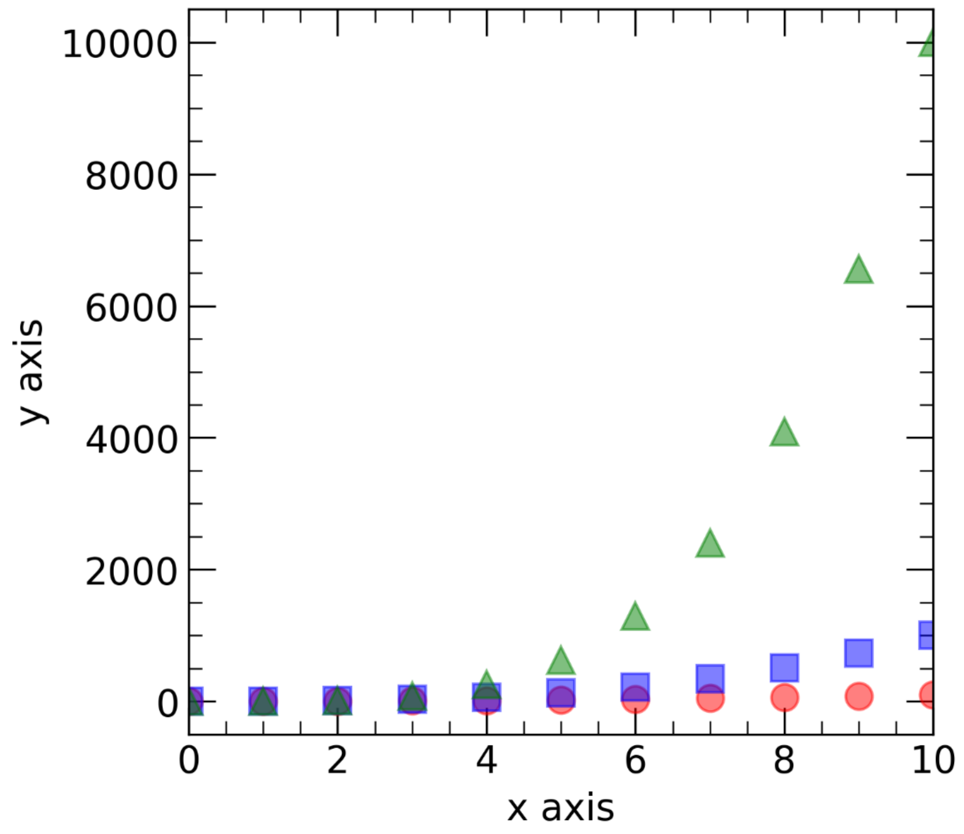


Both maximal margin classifiers **and** Support vector classifiers will fail here....
So we need **support vector machines!**

Support vector machines take advantage of different transformations of the data

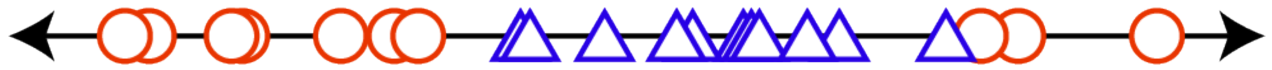


$Y=X^2$ works, but what else could have worked? Many transformations possible!



The support vector machine systematically increases the order of polynomial and calculates a support vector classifier for each kernel

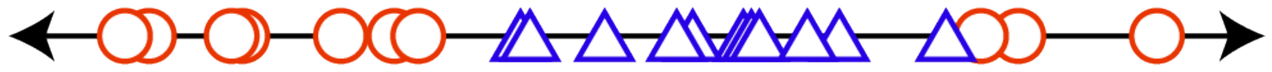
Radial basis function is another popular kernel for SVMs



Points near the unknown point have a weighted advantage on determining classification.

Similar to "Weighted nearest neighbor"

The kernel "trick" is that the data is never transformed!



Data is never actually transformed,
but calculations are done as if the transformation had taken place
(reduces computational requirements)

Let's look at the math behind polynomial kernel

General equation for polynomial kernel is

$$(a \times b + r)^d$$

So for 2nd order polynomial this becomes

$$\left(a \times b + \frac{1}{2}\right)^2 = \left(a \times b + \frac{1}{2}\right)\left(a \times b + \frac{1}{2}\right) = ab + a^2b^2 + \frac{1}{4}$$

Which is same as this dot product

$$\left(a, a^2, \frac{1}{2}\right) \cdot \left(b, b^2, \frac{1}{2}\right)$$

These are the data coordinates! (x axis cords, y axis cords, z axis cords which we ignore since they are the same for both)

This gives us an easy transformation, so no real transformation required

Let’s look at the math behind the radial kernel

General equation for polynomial kernel is

$$\exp(-\gamma(a-b)^2)$$

a and b are the two points being correlated, so distance squared shows why it is weighted to near neighbors.

γ is learned by cross validation to scale the influence of distance squared

The RBF has infinite dimensions… how?

$$\exp(-\gamma(a-b)^2) = \exp\left(-\gamma\left(a^2+b^2-2ab\right)\right) = \exp\left(-\gamma\left(a^2+b^2\right)\right)\exp(\gamma 2ab)$$

If $\gamma = \frac{1}{2}$ then we have

$$\exp\left(-\frac{1}{2}\left(a^2+b^2\right)\right)\exp(ab)$$

Second term can be replaced by a Taylor series expansion

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

Support vector machines have tunable hyperparameters

C parameter adds a penalty for each misclassified point (variance / bias tradeoff)

Typically: $0.1 < c < 100$

γ dictates how much influence near neighbors should have (low γ puts more data points together, high γ fewer points together because radius is smaller)

Typically: $0.0001 < \gamma < 10$

https://towardsdatascience.com/hyperparameter-tuning-for-support-vector-machines-c-and-gamma-parameters-6a5097416167

Extrapolation in Materials Informatics