Introduction to Data Science

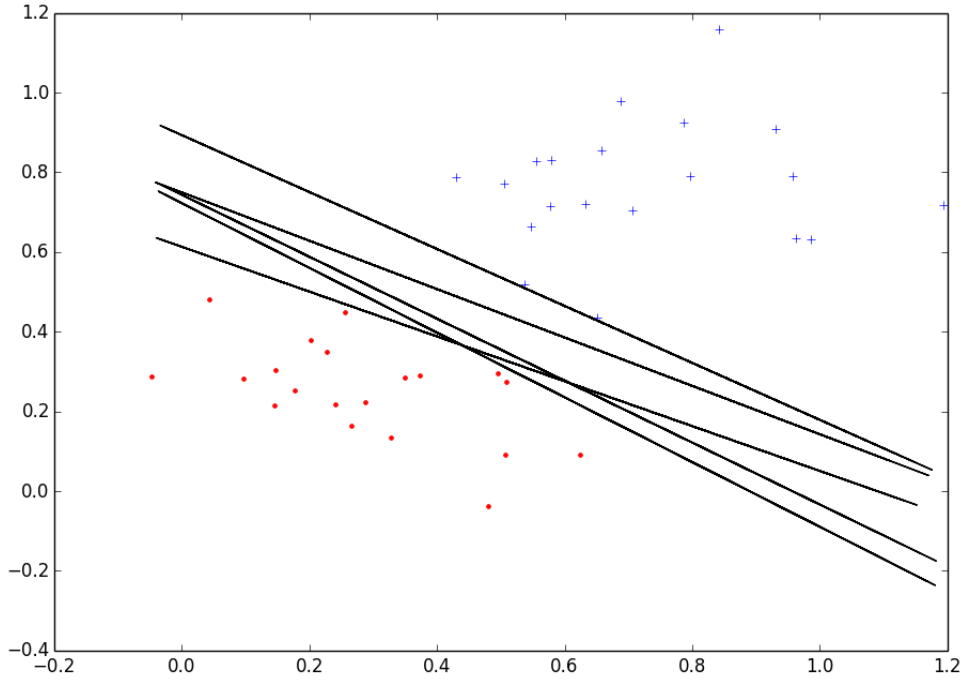
SUPPORT VECTOR MACHINES

BRIAN D'ALESSANDRO

SEPARATING HYPERPLANES

When data is separable, there is often more than one hyper-plane that can perfectly separate the data.

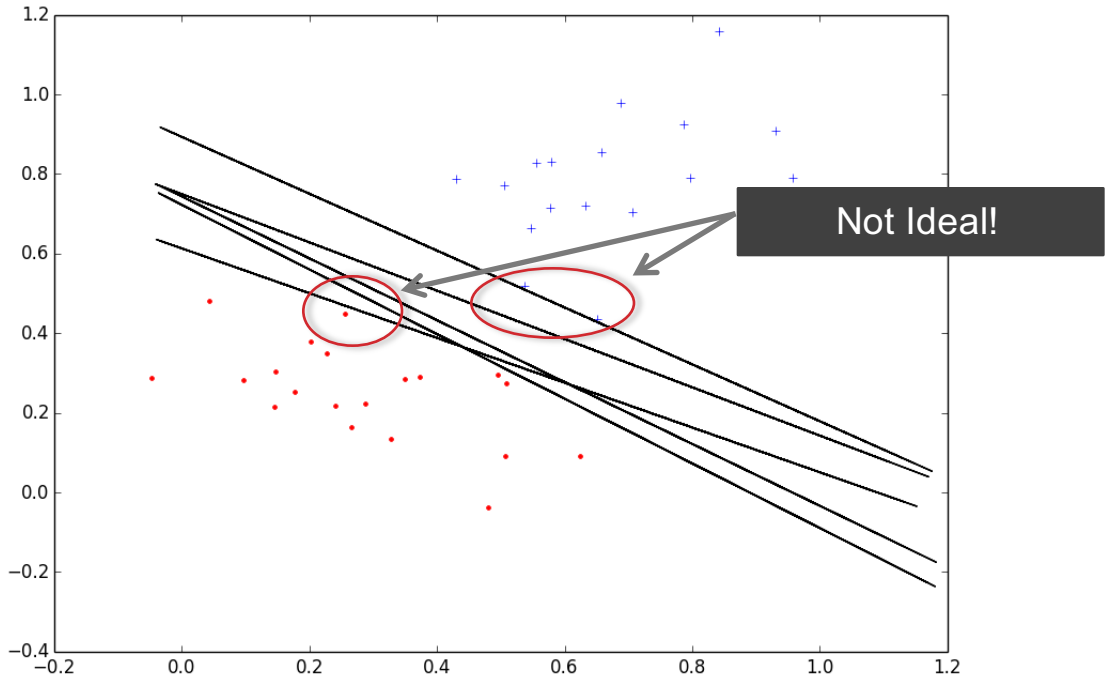
Given so many choices, it would be nice to have a principled way to choose an optimal one.



SEPARATING HYPERPLANES

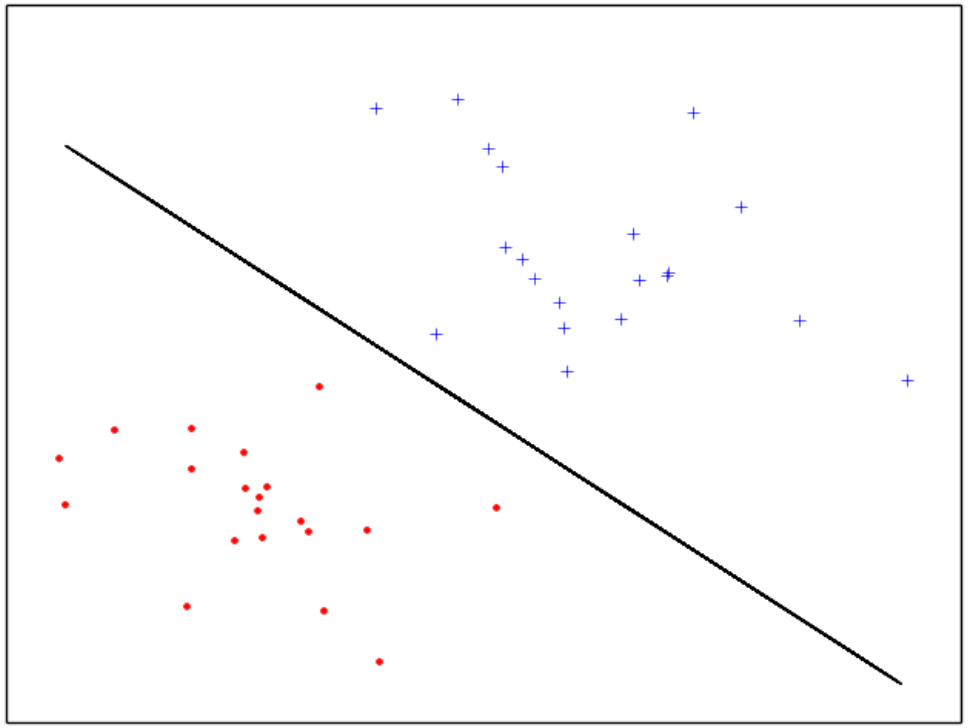
Ideally, we want a line that is as far from all points as possible.

A safe bet would be a line that sits as close to the middle of the empty separating the two classes



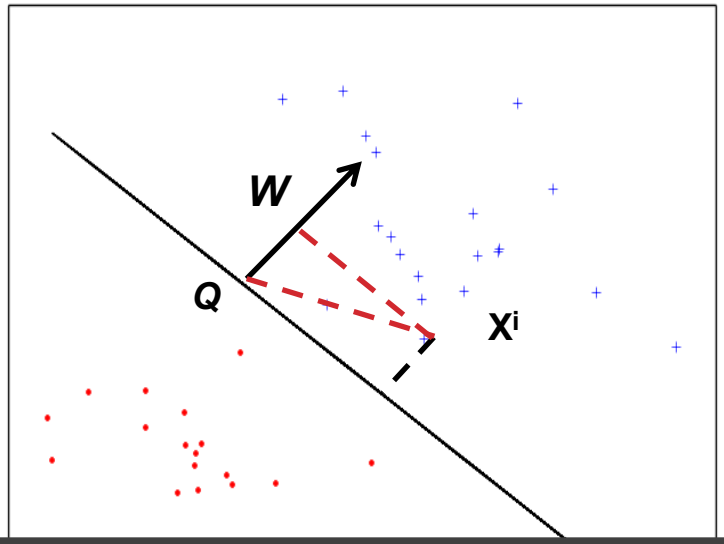
MAXIMAL MARGIN HYPERPLANE

We can formalize our intuition of searching for a plane that is maximally far from as many points as possible.



SOME FUNDAMENTALS

We'll start with some basic linear algebra



The distance of X^i to the line is given by the projection of the segment QX^i onto W and is given by:

We have a feature space:

$$X =< x_1, ...x_k >$$

A hyper-plane:

$$W \cdot X + t = 0$$

A normal vector:

$$W =< w_1, ...w_k >$$

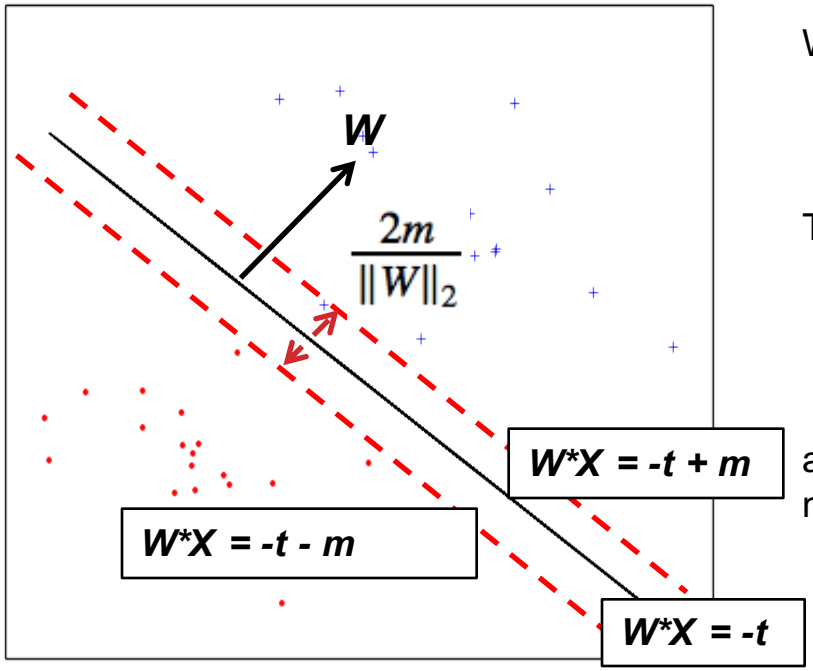
Distance(X^i, plane)

$$\frac{W \cdot X^i + t}{\|W\|_2}$$

ADDING CONSTRAINTS

If we add some constraints, the problem becomes easier to solve.

M^+ = distance of closest positive point to W
M^- = distance of closest negative point to W



With the constraint:

$$M^+ = M^- = m = 1$$

The margin then becomes:

$$\frac{2m}{\|W\|_2}$$

and this is what we seek to maximize.

THE OPTIMIZATION PROBLEM

Maximizing: $\frac{2m}{\|W\|_2}$ is equivalent to the following optimization problem:

$$W^*, t^* = \underset{W,t}{\operatorname{argmin}} \frac{1}{2} \|W\|_2^2 \quad \text{subject to } y_i(W * X_i + t) \geq 1$$

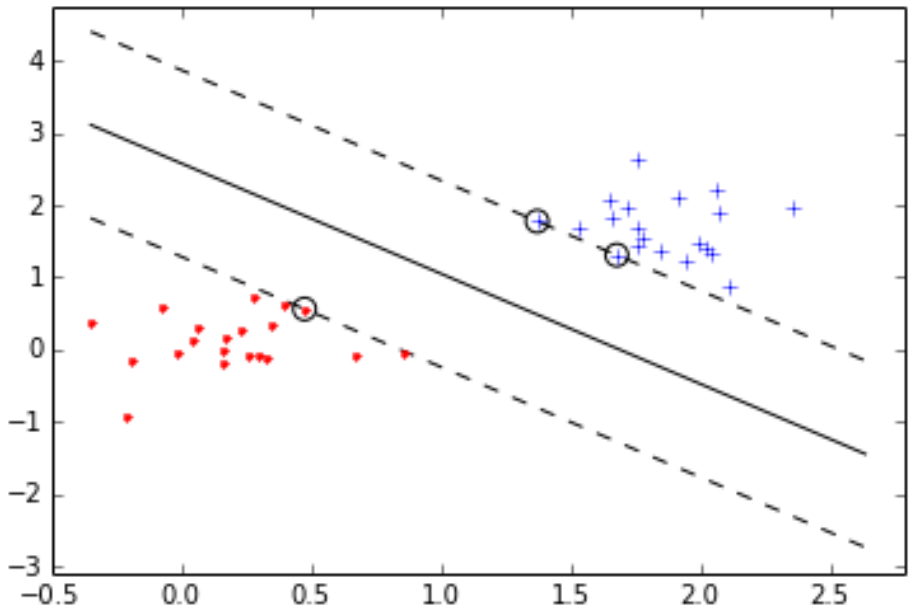
- The constraints ensure that all points are on the correct side of the line.
- This can be solved with Convex Optimization and the method of Langrange Multipliers (beyond the scope of this class)

ET VOILA!

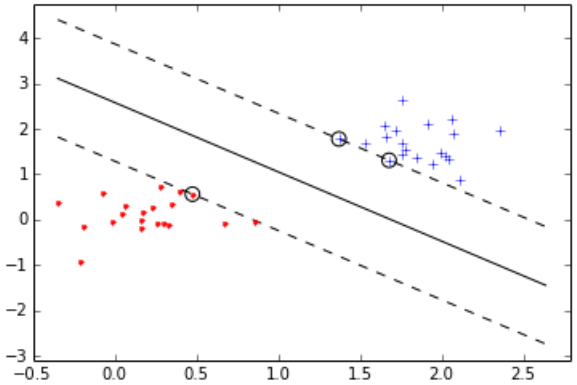
An optimal separating hyper-plane along with margin planes and support vectors identified.

Classification of a new point X then becomes: f(X) = sign(W*X + t).

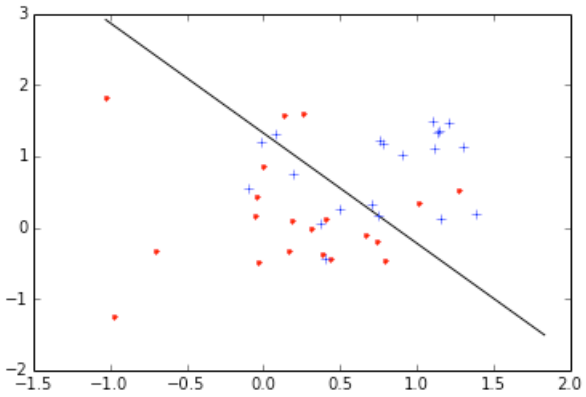
Effectively, we are finding which side of the line a new point is on.



REALITY



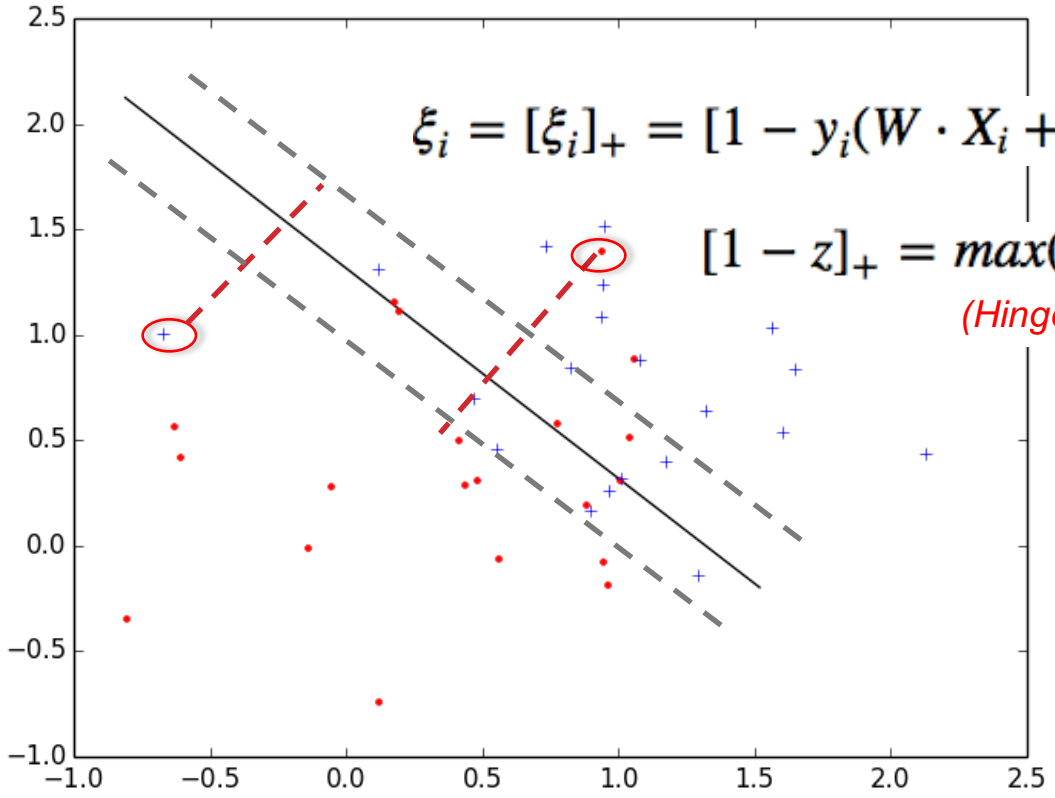
This is somewhat of an unrealistic scenario. Data is seldom this cleanly separated.



Real data looks more like this.

INTRODUCING SLACK VARIABLES

We can account for errors in our problem by introducing slack variables, which measure distance of a misclassified points to the margin.



$$\xi_i = [\xi_i]_+ = [1 - y_i(W \cdot X_i + t)]_+$$

$$[1 - z]_+ = max(0, 1 - z)$$

(Hinge Loss)

SOFT MARGIN SVM

Our optimization problem now can be written:

$$W^*, t^* = \underset{W,t}{\operatorname{argmin}} \frac{1}{2} \|W\|_2^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{subject to } y_i(W \cdot X_i + t) \geq 1 - \xi_i$$

Which can also be expressed as:

$$W^*, t^* = \underset{W,t}{\operatorname{argmin}} \lambda \|W\|_2^2 + \sum_{i=1}^{n} [1 - y_i(W \cdot X_i + t)]_+$$

Where $\lambda = 1/2C$

SOFT MARGIN SVM AS ERM

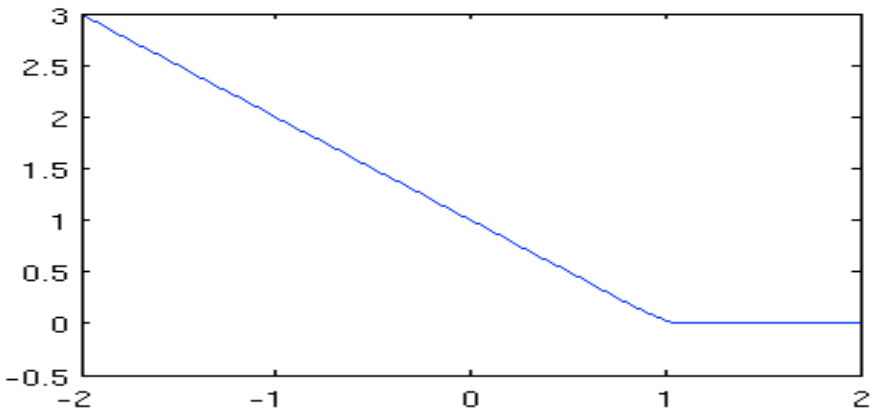
With this formulation we are trying to do two things:

$$W^*, t^* = \underset{W,t}{\operatorname{argmin}} \lambda \|W\|_2^2 + \sum_{i=1}^{n} [1 - y_i(W \cdot X_i + t)]_+$$

Maximize the margin

Minimize the training error (hinge loss)

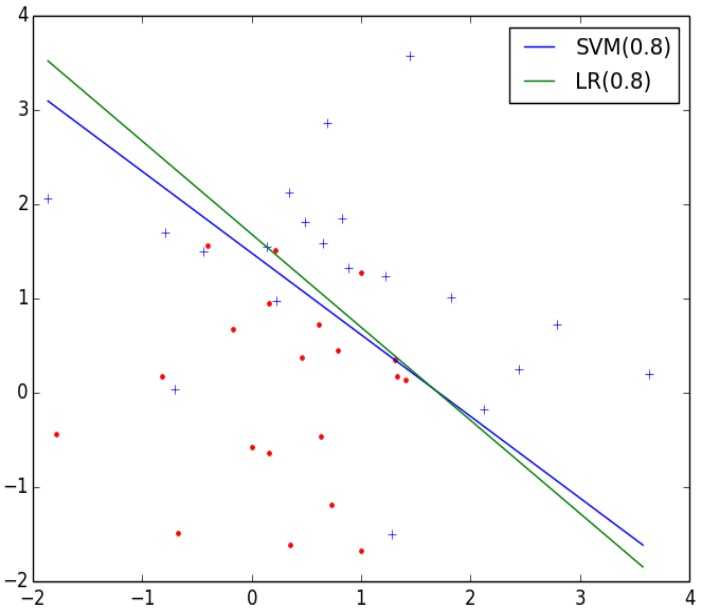
Example Hinge Loss



SVM VS. LOGREG

When dealing with linear models for classification, SVMs and Logistic Regression have very similar solutions. When to use which is a matter of needing to exploit each's unique properties.

SVM vs. LR on Simulated Data



SVM

- Strong geometric interpretation
- Easily extendable to non-linear decision surfaces
- Strong learning guarantees

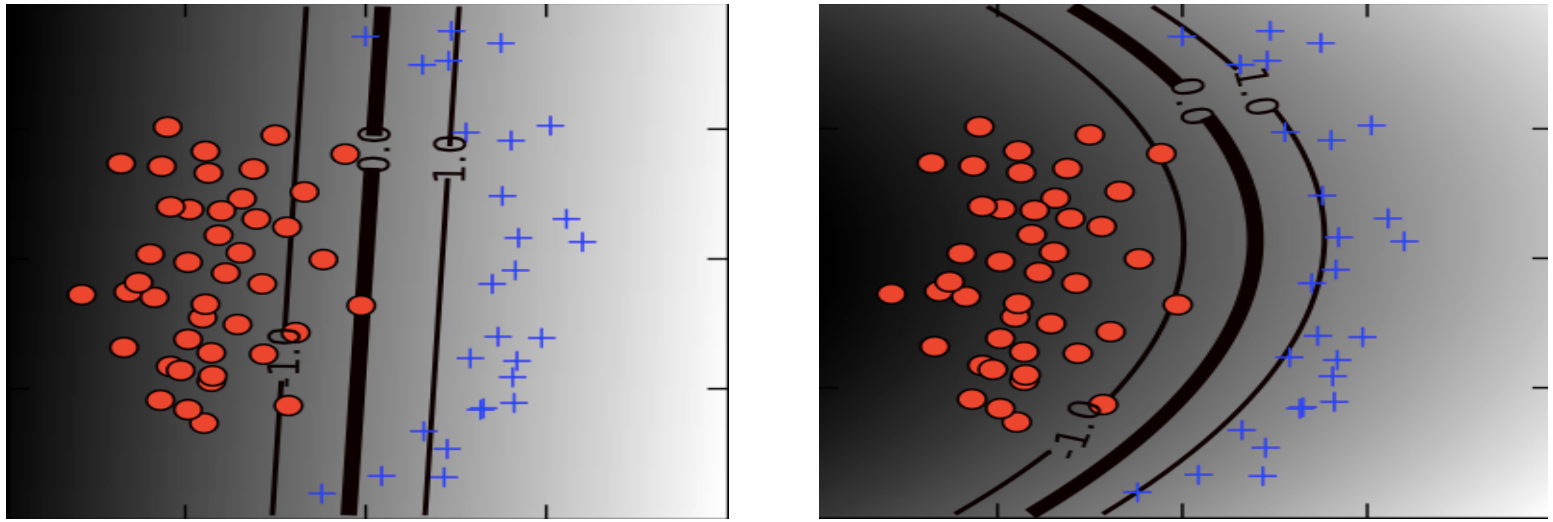
LR

- Designed for P(Y|X)
- Strong statistical properties
- Suited for ranking

SVMS FOR NON-LINEAR PROBLEMS

In practice, most supervised learning problems do better with non-linear separating boundaries. While SVMs are always linear problems, they can effectively solve non-linear problems using clever transformations of the input space.

Example linear and non-linear boundaries in original input space



THE DUAL FORM

Optimization theory let's us express certain minimization problems as an alternative maximization problem, called the "Dual" form

Primal Form

$$W^*, t^* = \underset{W,t}{\operatorname{argmin}} \frac{1}{2} \|W\|_2^2 + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i(W \cdot X_i + t) \geq 1 - \xi_i$

Dual Form

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{subject to:} \quad & \sum_{i=1}^{n} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C. \end{aligned}$$

THE DUAL FORM: SOLVED

Solving for the dual form allows us to express the weight vector ***W*** as a linear combination of the input vectors.

Starting again with: $f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x} + b$ and the dual form: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i$

We can now rewrite our SVM as: $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i^\mathsf{T}\mathbf{x} + b$

Which can be generalized to: $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b$

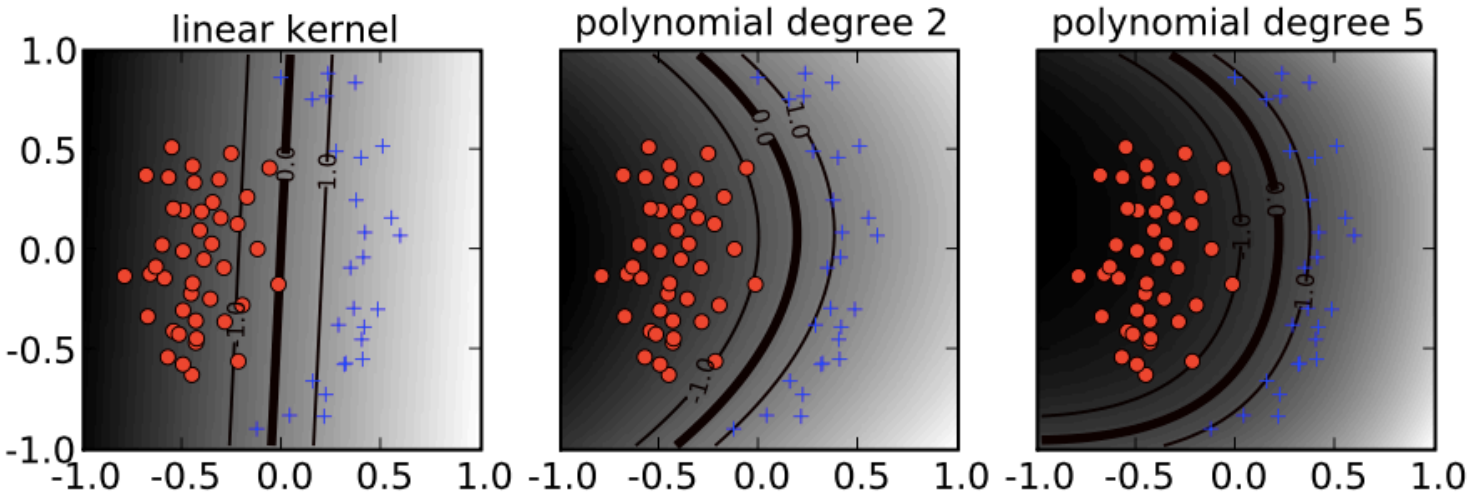
Where: $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\mathsf{T}\phi(\mathbf{x}')$

With this set of steps we transform our features X into a new space ϕ(X), and then build a linear SVM on ϕ(X)

THE POLYNOMIAL KERNEL

The polynomial kernel expands a vector X into a polynomial of degree *d*. Complexity of the kernel is controlled via the hyper-parameter *d*.

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\mathsf{T}\mathbf{x}' + 1)^d$$



THE GAUSSIAN KERNEL

This kernel is the default in Sklearn (which is a strong endorsement), and generally has the most flexibility to model any decision surface. Complexity is controlled via the hyper-parameter γ (larger = more complex)

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$$