Learning with Kernels

Matthieu Bulté

06-06-2017

Are are we doing?

Mathematics is about transforming hard problems into problems trivial to solve

What is our problem?

We have data We have labels We want to infer a rule to label new data

Less abstract please

Decision function Decision Boundary (Plots of many different decision boundaries)

A trivial problem

Linearly seperale data set

Solving the trivial problem

The Line

Demo 1 - which line?

Margin maximization

Picking the best trivial solution (opt problem)

Demo 1 - cont'd

Da dual

(lagrangian)

Support Vectors

H only depends on support vectors!

A not so trivial problem

non linearly separable datasets

Time to be smart

How do we make this problem easier?

Transforming the problem

Space travel
show a pic of a rocket -> not this kind of space travel…

Space Travel

Project the data to another space where the problem is easy
Solve the problem Bring the easy solution to the hard problem

Demo 2

Not yet there

Projection is expensive (polynomial has factorial grows)

Could we avoid it?

$\phi(x) = (x_1^2, x_2^2, \text{sqrt}(2)x_{1x2})$ $<\phi(x), \phi(y)> = \ldots = (<x, y>)^2$
$:= k(x, y)$

Hello from the other side

Change our PoV, we have
(phi, space) -> kernel

Hello from the other side

Change our PoV, we want
(phi, space) <- kernel
mind blown

Mercer & co

Many different space can be constructed, all based on the same idea. We chose to present the easy one and left out the useful one. Intuition is king.

In other words

Plug & Play

Demo 3 - some kernels

https://cs.stanford.edu/people/karpathy/svmjs/demo/

the good, the bad, ...

no uggly :)

bad

- parameters tuning
- training time
- domain knowledge (what do I know when talking about very complex problems?)

good

- E[P(error)] <= ...
- performance on small data sets
- domain knowledge

Thank you