May 29, 2021

Solving for θ:

Use SVM software packages to solve for θ.

Need to Specify:

→ choice of C

→ choice of kernel (similarity function)

e.g. Linear kernel (no kernel)

↳ predict $y=1$ if $\theta^T x \geq 0$, $x \in \mathbb{R}^{n+1}$

↳ good for large # features (n) w/ small training set (m)

e.g. Gaussian kernel

↳ $f_i = \exp\left(-\frac{\|x-\ell^{(i)}\|^2}{\sigma^2}\right)$, where $\ell^{(i)} = x^{(i)}$, $x \in \mathbb{R}^n$

↳ need to choose σ^2

↳ good for small n and/or large m

Kernal Code:

$f^{(i)}$ $x^{(i)}$ $l^{(i)}$

function $f = \text{kernel}(x_1, x_2)$

$$f = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right)$$

return

✡ Perform feature scaling <u>before</u> using Gaussian kernel.

Other Kernals:

Note: Not all similarity functions make valid kernels. All valid kernels need to satisfy Mercer's Theorem to ensure SVM packages' optimizations run correctly and do not diverge.

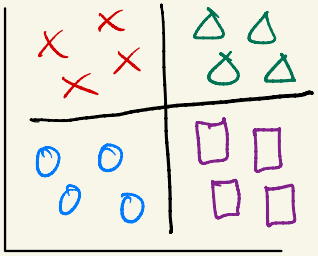
e.g. Polynomial kernel

↳ $k(x, l) = (x^T l)^2, (x^T l)^3, (x^T l + 1)^3$, etc

✡ $(x^T l + \text{constant})^{\text{degree}}$

Also: String kernels, chi-square kernel, histogram intersection kernel ...

Multi-Class Classification:



$y \in \{1, 2 \dots K\}$

※ Most SVMs have built-in multi-class functionality

→ Otherwise, use One-vs-all method.

↳ i.e. Train K SVMs for $y = 1, 2 \dots K$, get $\theta^{(1)}, \dots \theta^{(K)}$.
Pick class i w/ largest $(\theta^{(i)})^T x$.

Logistic Regression vs SVMs:

n = # features ($x \in \mathbb{R}^{n+1}$)

m = # training examples

1. If n is large relative to m, use logistic regression or SVM w/ linear kernel.
   ↳ e.g. $n = 10000$, $m = 10 \dots 1000$
2. If n is small and m is intermediate, use SVM w/ Gaussian kernel.
   ↳ e.g. $n = 1\text{-}1000$, $m = 10\text{-}10000$

3. If n is small, m is large, create/add more features and use logistic regression or linear SVM.
   - e.g. $n = 1-1000$, $m = 50000+$

* A neural network could also work for all three, but may be slower to train.