# Support Vector Machines

## Large Margin Classification *(Support Vector Machine Algorithm)*

Optimization Objective

* Alternative view of logistic regression
* Modify cost function
* SVM hypothesis output either 0 or 1, not probability

Large Margin Intuition

* Find largest margin around decision boundary

Mathematics Behind Large Margin Classification

* Vector inner product: max p 🡪 min norm

## Kernels

Kernels I

* Non-linear decision boundaries (Gaussian Kernel)
* Similarity functions become new features
* σ2 has the effect of λ (opposite to C)
* Similarity functions become new features

Kernels II

* Choose landmarks same as training examples
* One feature (of the very example) will be 1
* SVM with kernels
* Cost: solve function of θTf instead of θT*x*
* Regularization:
* Ignoring θ0, , which SVM implementations change to rather than minimizing norm, allowing scaling to much bigger training sets
* Determine optimum bias-variance tradeoff by changing C and σ2

## SVM in Practice

Using SVM

* liblinear, libsvm
* Choice of parameter (C), and choice of kernel
* Linear kernel (no kernel), Gaussian, polynomial, string, chi-square, histogram intersection
* Use Mercer’s Theorem to ensure optimizations run correctly, do not diverge
* Perform feature scaling before using Gaussian kernel
* Multi-class classification
* SVM built-in multi-class functionality
* One-vs.-all method – train K SVM’s
* LR (or SVM with linear kernel) or SVM
* Large n (compared to m) 🡪 LR/SVM w/ no kernel
* Small n, intermediate m 🡪 Gaussian
* Small n, large m 🡪 LR/SVM w/ no kernel (slow w/ SVM)
* Neural networks likely to work, but really slowly
* SVM optimization will always find global minima (or close)
* Linear kernel: large n, small m?
* Mentioned before obtaining more data in case of large n helps with overfitting

*REFER TO SLIDES*