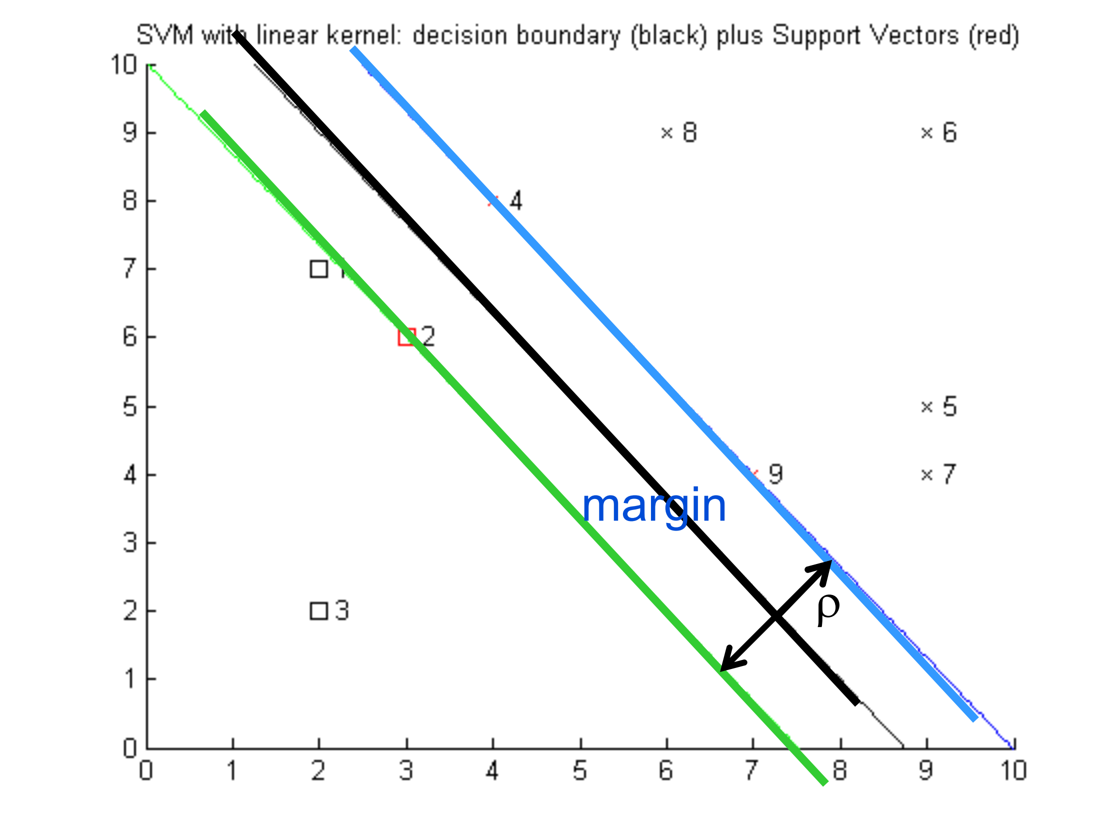
CSSE463: Image Recognition Day 15

- Today:
  - Project intro
  - Wrap up SVM and do demo
- Near future:
  - Neural nets
  - lightning talks (see other slides now), Lab 5 due
  - lab for sunset detector
- Tuesday 6th week: Mid-term exam

Review: SVMs: "Best" decision boundary



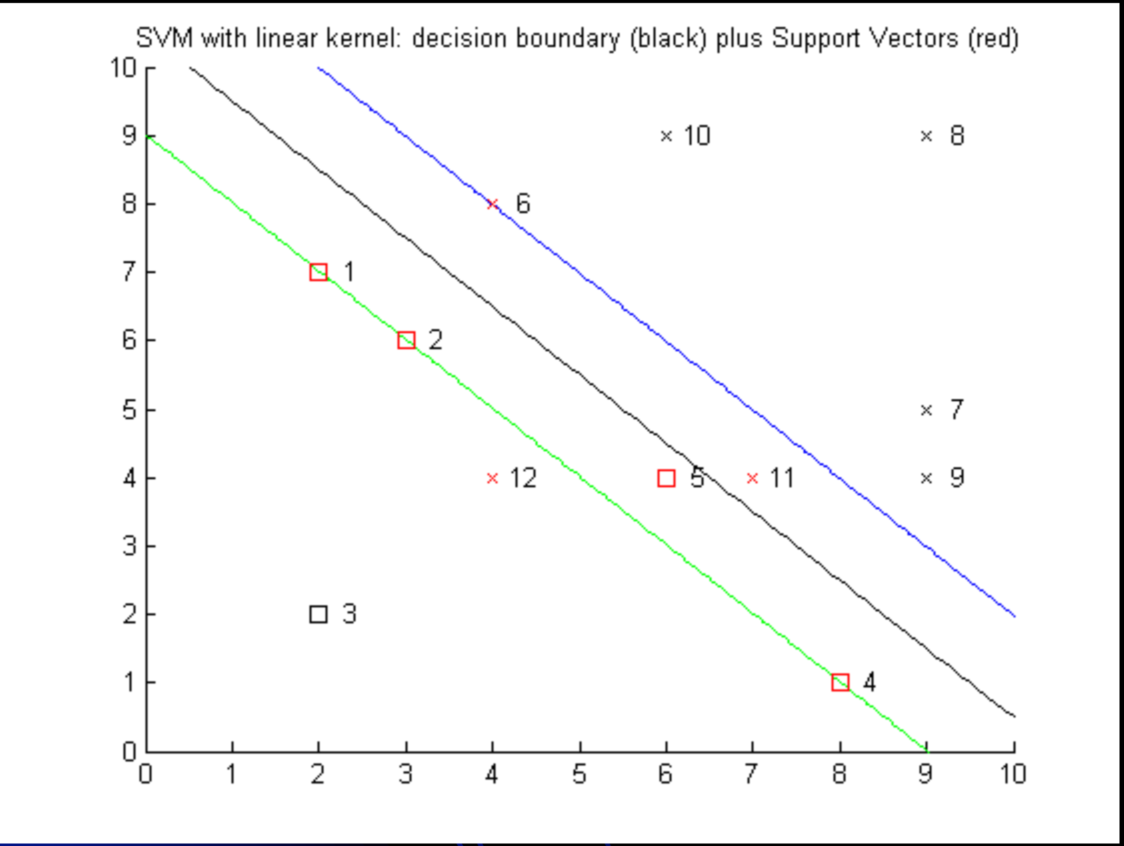
- The "best" hyperplane is the one that *maximizes the margin, ρ,* between the classes. Equivalent to:

$$\min \phi(w) = \frac{1}{2} w^T w$$

$$d_i (w^T x_i + b) \geq 1 \ \ for \ i = 1,2,....N$$

- Solve using quadratic programming

Non-separable data



- Allow data points to be misclassifed
- But assign a cost to each misclassified point.
- The cost is bounded by the parameter C (which you can set)
- You can set different bounds for each class. Why?
  - Can weigh false positives and false negatives differently

Can we do better?

- Cover's Theorem from information theory says that we can map nonseparable data in the input space to a feature space where the data is separable, with high probability, if:
  - The mapping is nonlinear
  - The feature space has a higher dimension
- The mapping is called a *kernel function*.
  - Replace every instance of $x_i x_j$ in derivation with $K(x_i x_j)$
  - Lots of math would follow here to show it works
- Example:
  - separate x_1 XOR x_2 by adding a dimension $x_3 = x_1 x_2$

Most common kernel functions

- Polynomial
- Gaussian Radial-basis function (RBF)
- Two-layer perceptron

$$K(x, x_i) = (x^T x_i + 1)^p$$

$$K(x, x_i) = \exp\left(-\frac{1}{2\sigma^2} \|x - x_i\|^2 \right)$$

$$K(x, x_i) = \tanh\left(\beta_0 x^T x_i + \beta_1 \right)$$

- You choose p, σ, or β_i
- My experience with real data: **use Gaussian RBF!**

Easy	Difficulty of problem	Hard
p=1, p=2,	higher p	RBF

Demo

- Software courtesy of http://ida.first.fraunhofer.de/~anton/software.html (GNU public license)

- Lab 5 (we start today!):
  - Download the Matlab functions that train and apply the SVM.
  - The demo script contains examples of how to call the system
  - Write a similar script to classify data in another toy problem

- Directly applicable to sunset detector

Kernel functions

- Note that a hyperplane (which by definition is linear) in the feature space = a nonlinear boundary in the input space
  - Recall the RBFs
- Note how choice of σ affects the classifier

Comparison with neural nets

- Expensive
  - Training can take a *long* time with large data sets. Consider that you'll want to experiment with parameters…
  - But the classification runtime and space are *O(sd)*, where *s* is the number of support vectors, and d is the dimensionality of the feature vectors.
  - In the worst case, *s* = size of whole training set (like nearest neighbor)
  - But no worse than implementing a neural net with *s* perceptrons in the hidden layer.
- Empirically shown to have good generalizability even with relatively-small training sets and no domain knowledge.

Speaking of neural nets:

- Back to a demo of matlabNeuralNetDemo.m
- Project discussion?