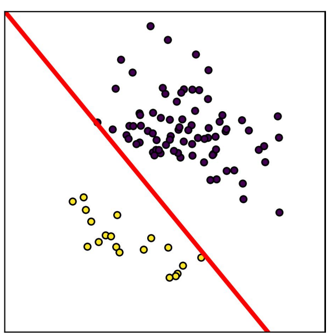
Support Vector Machines

Support Vector Machines (SVMs)

- Binary classification
- Output is not probability (real number) but binary 0 or 1
- Map data to higher dimensional space
- Large margin classification

Large Margin Classification

Data are linearly separable

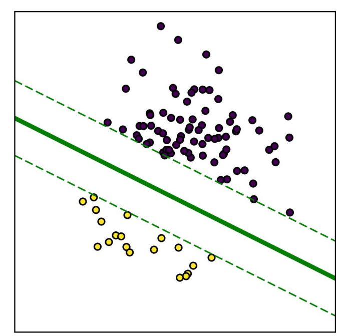


There are many possible linear decision boundaries

We might expect some decision boundaries to generalize better than others

Large Margin Classification

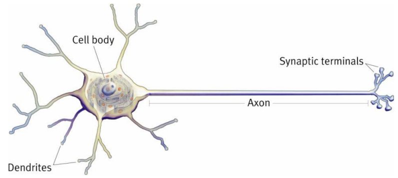
Margin is distance between decision boundary and nearest data points on each side

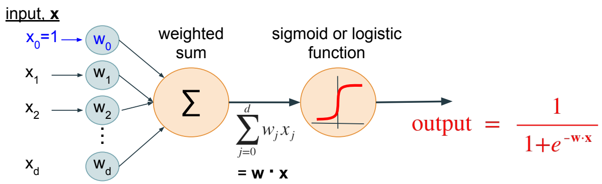


The data points at the margin boundary are called *support vectors*.

Aim for decision boundary with large margin

Recall: Logistic Regression

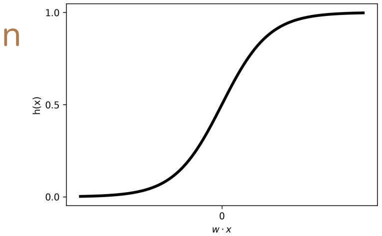


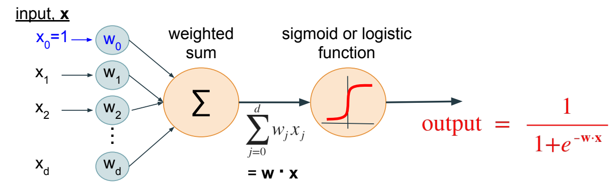


Recall: Logistic Regression

- In L.R., interpret $h(\boldsymbol{x})$ as probability
- If $\boldsymbol{w} \cdot \boldsymbol{x} \geq 0$ then output 1
- If $\boldsymbol{w} \cdot \boldsymbol{x} < 0$ then output 0

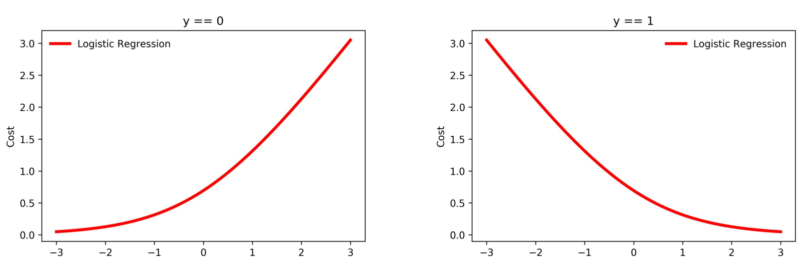
Hypothesis, $h(x)$, for SVM





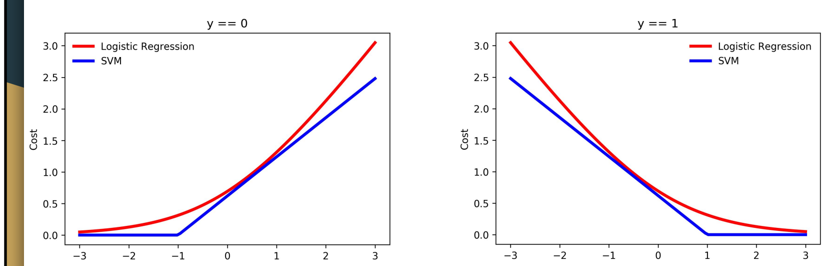
Logistic Regression Cost Function

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} y^{(i)}(-\log(h(x^{(i)}))) + (1-y^{(i)})(-\log(1-h(x^{(i)})))$$



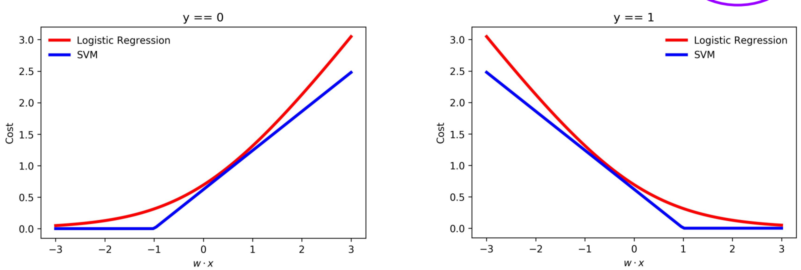
SVM Cost Function

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} y^{(i)}(-\log(h(x^{(i)}))) + (1-y^{(i)})(-\log(1-h(x^{(i)})))$$



SVM Cost Function with Regularization

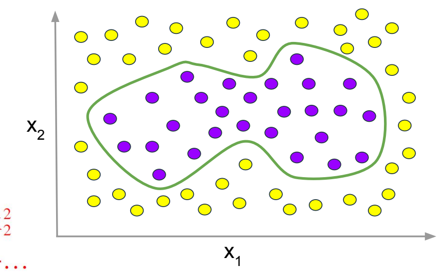
$$J(w) = \left[\frac{1}{n} \sum_{i=1}^{n} y^{(i)} \boxed{-\log(h(x^{(i)}))} + (1-y^{(i)}) \boxed{-\log(1-h(x^{(i)}))} \right] + \frac{\lambda}{2n} \sum_{j=1}^{d} w_j^2$$



Non-Linear Decision Boundaries

One possibility is to add higher order polynomial features

$$h(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^3 x_2^2 + w_4 x_1^5 + w_5 x_1^2 x_2^4 + w_6 x_2^9 + \ldots$$



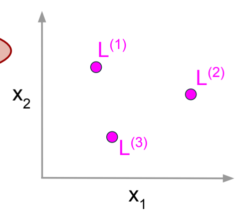
Kernels

Radial Basis Function Kernel

$$f_1 = \text{similarity}(x, L^{(1)}) = \exp\left(-\frac{\|x - L^{(1)}\|^2}{2\sigma^2}\right)$$

$$f_2 = \text{similarity}(x, L^{(2)}) = \exp\left(-\frac{\|x - L^{(2)}\|^2}{2\sigma^2}\right)$$

$$f_3 = \text{similarity}(x, L^{(3)}) = \exp\left(-\frac{\|x - L^{(3)}\|^2}{2\sigma^2}\right)$$



For each data point *x*, compute new features based on the proximity of *x* to the landmarks

Example

Predict 1 if $w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 \geq 0$

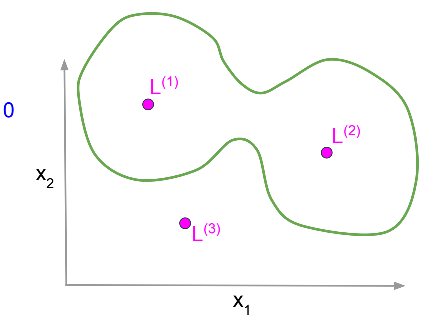
Predict 0 if $w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 < 0$

Suppose the SVM is trained and it learns the parameters:

$w_0 = -0.5$ $w_1 = 1$ $w_2 = 1$ $w_3 = 0$

$w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3$

$= -0.5 + 1 * f_1 + 1 * f_2 + 0 * f_3$



Example

Predict 1 if $w_0 + w_1f_1 + w_2f_2 + w_3f_3 \geq 0$

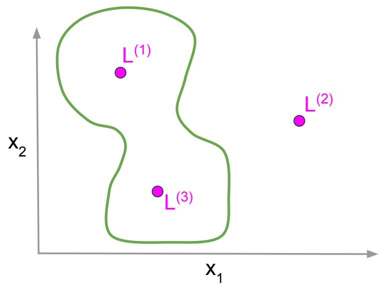
Predict 0 if $w_0 + w_1f_1 + w_2f_2 + w_3f_3 < 0$

Suppose the SVM is trained and it learns the parameters:

$w_0 = -0.5$ $w_1 = 1$ $w_2 = 0$ $w_3 = 1$

$w_0 + w_1f_1 + w_2f_2 + w_3f_3$

$= -0.5 + 1*f_1 + 0*f_2 + 1*f_3$



Example

Predict 1 if $w_0 + w_1f_1 + w_2f_2 + w_3f_3 \geq 0$

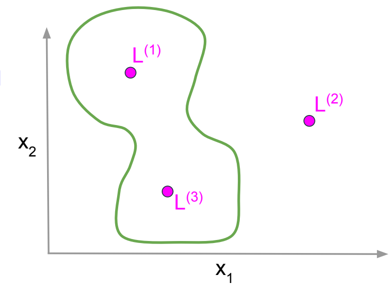
Predict 0 if $w_0 + w_1f_1 + w_2f_2 + w_3f_3 < 0$

Suppose the SVM is trained and it learns the parameters:

$w_0 = 0.5$ $w_1 = -1$ $w_2 = 0$ $w_3 = -1$

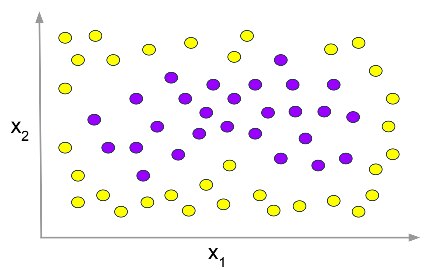
$w_0 + w_1f_1 + w_2f_2 + w_3f_3$

$= 0.5 + -1*f_1 + 0*f_2 + -1*f_3$



SVMs with Kernels

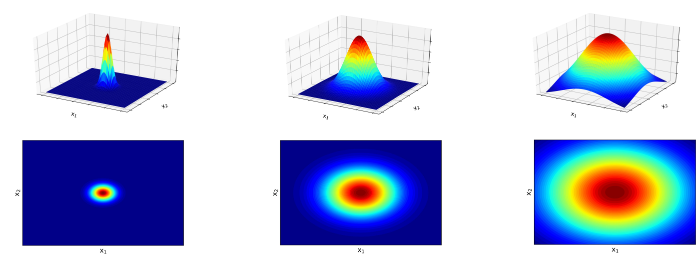
- Suppose there are n training examples with d features
- Use each of the n training examples as a landmark
- So there will be n features (for a data point x, compute its similarity to each of the n landmarks)
- Thus, data are mapped to a high dimensional space prior to using our large margin SVM classifier



Kernel Variations

Using no kernel is called a "linear kernel"

Modifying σ parameter in RBF kernel



Multi-Class Classification: one-vs.-all

Train *K* separate classifiers, one for each of the *K* classes. For a data point, classify it based on which of the classifiers output the highest value, e.g., which SVM output the largest $w \cdot x$.

<u>Song genres:</u>
Blues, Country, Hip Hop, Jazz, Pop, Rock

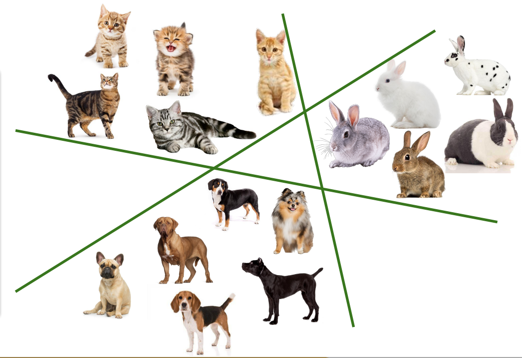
<u>Handwritten digits:</u>
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

<u>Email labeling:</u>
Family, School, Summer, Friends, CS305

Family vs not Family
CS305 vs not CS305
Summer vs not Summer
Friends vs not Friends
School vs not School

Multi-Class Classification: one-vs.-all

Train *K* separate classifiers, one for each of the *K* classes. For a data point, classify it based on which of the classifiers output the highest value, e.g., which SVM output the largest $w \cdot x$.



Multi-Class Classification: one-vs.-one

Train $K*(K-1)/2$ separate classifiers, one for each <u>pair</u> of the *K* classes. For a data point, classify it based on which class received the highest number of positive "+1" predictions from the $K*(K-1)/2$ classifiers.

<u>Song genres:</u>
Blues, Country, Hip Hop, Jazz, Pop, Rock

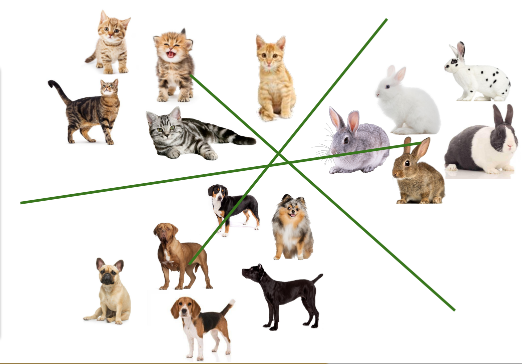
<u>Handwritten digits:</u>
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

<u>Email labeling:</u>
Family, School, Summer, Friends, CS305

Friends vs CS305
Family vs School
Family vs Friends
Summer vs CS305
School vs CS305
Summer vs Friends
Family vs Summer
Family vs CS305
School vs Summer
School vs Friends

Multi-Class Classification: one-vs.-one

Train $K*(K-1)/2$ separate classifiers, one for each <u>pair</u> of the *K* classes. For a data point, classify it based on which class received the highest number of positive "+1" predictions from the $K*(K-1)/2$ classifiers.



Comparing Classifiers

- Number of training examples *n* relative to number of features *d*
- Efficiency. Interpretability.
- Is it the classifiers or the data that matter?

Overview