Machine Learning and Data Analytics
ME 5013- Fall 2019

Lectures 09 and 10

- Support Vector Machine

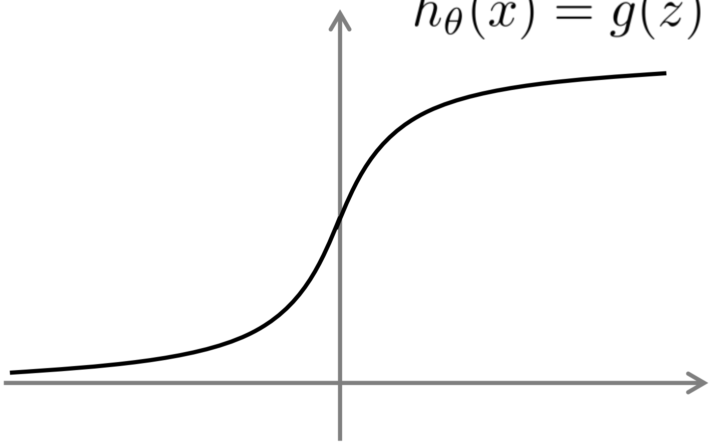
UTSA®
The University of Texas at San Antonio™

Adel Alaeddini, PhD
Associate Professor of Mechanical Engineering
Advanced Data Engineering Lab
adel.alaeddini@utsa.edu

Disclosure: the slides are adopted from Dr. Andrew Ng Machine Learning Course

Alternative view of logistic regression

$$h_{\beta}(x) = \frac{1}{1 + e^{-x\beta}}$$



$z = x\beta$

If $y = 1$, we want $h_{\beta}(x) \approx 1$, $x\beta \gg 0$

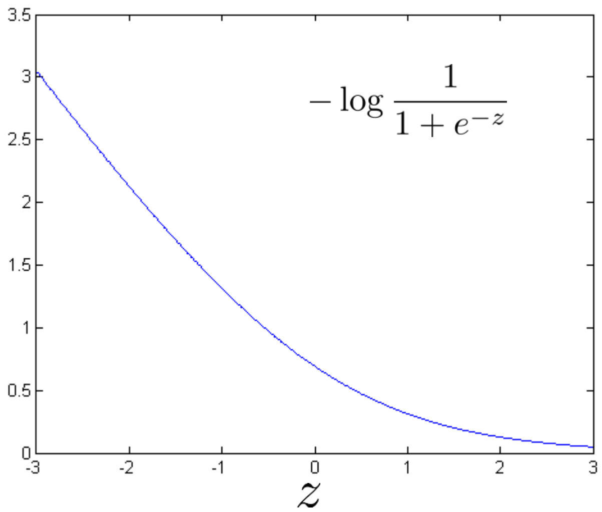
If $y = 0$, we want $h_{\beta}(x) \approx 0$, $x\beta \ll 0$

Alternative view of logistic regression

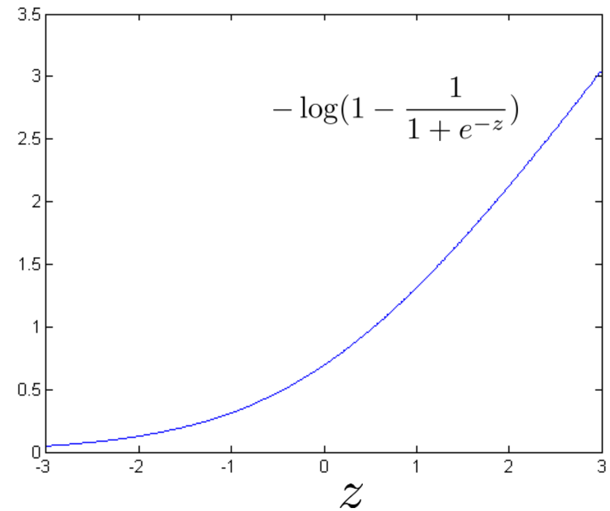
Cost of example: $-(y \log h_\beta(x) + (1-y)\log(1-h_\beta(x)))$

$$= -y \log \frac{1}{1+e^{-x\beta}} - (1-y)\log(1-\frac{1}{1+e^{-x\beta}})$$

If $y = 1$ (want $x\beta \gg 0$):



If $y = 0$ (want $x\beta \ll 0$):



Support vector machine

Logistic regression:

$$\min_{\beta} \frac{1}{n} \left[\sum_{i=1}^{n} y^{(i)} \left(-\log h_{\beta}(x^{(i)}) \right) + (1 - y^{(i)}) \left((-\log(1 - h_{\beta}(x^{(i)})) \right) \right] + \frac{\lambda}{2\,n} \sum_{j=1}^{p} \beta_j^2$$

Regularization term

Support vector machine:

$$\min_{\beta} C \sum_{i=1}^{n} \left[y^{(i)} cost_1(x^{(i)} \beta) + (1 - y^{(i)}) cost_0(x^{(i)} \beta) \right] + \frac{1}{2} \sum_{j=1}^{p} \beta_j^2$$

Regularization term

Hypothesis:

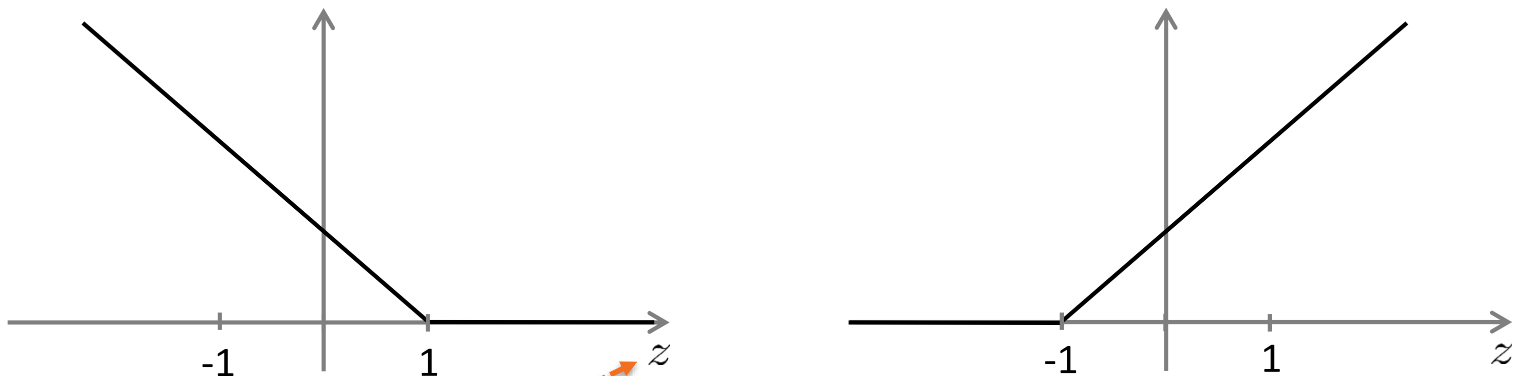
$$h_{\beta}(x) = \begin{cases} 1 & if\ x^{(i)}\beta \geq 0 \\ 0 & Otherwise \end{cases}$$

- No division by n
- C vs λ

Large Margin Intuition

Support Vector Machine

$$\min_{\beta} C \sum_{i=1}^{n} \left[y^{(i)} cost_1(x^{(i)}\beta) + (1 - y^{(i)}) cost_0(x^{(i)}\beta) \right] + \frac{1}{2} \sum_{j=1}^{p} \beta_j^2$$



If $y = 1$, we want $x\beta \geq 1$ (not just ≥ 0)

If $y = 0$, we want $x\beta \leq -1$ (not just < 0)

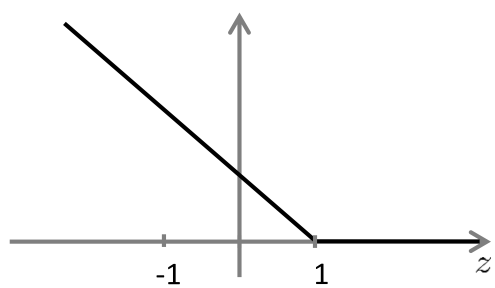
(extra) safety margin factor

SVM Decision Boundary

$$\min_{\beta} C \sum_{i=1}^{n} \left[y^{(i)} cost_1(\; x^{(i)}\beta \;) + (1 - y^{(i)}) cost_0(\; x^{(i)}\beta \;) \right] + \frac{1}{2} \sum_{j\,=\,1}^{p} \beta_j^2$$

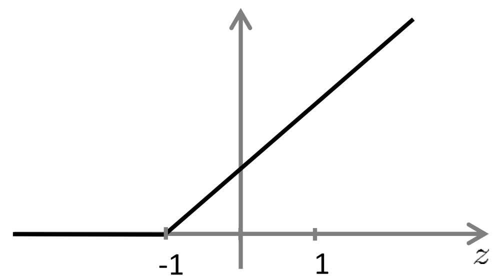
Whenever $y^{(i)} = 1$:

$$x^{(i)}\beta \geq 1$$



Whenever $y^{(i)} = 0$:

$$x^{(i)}\beta \leq -1$$



For very large C, i.e. C=10000

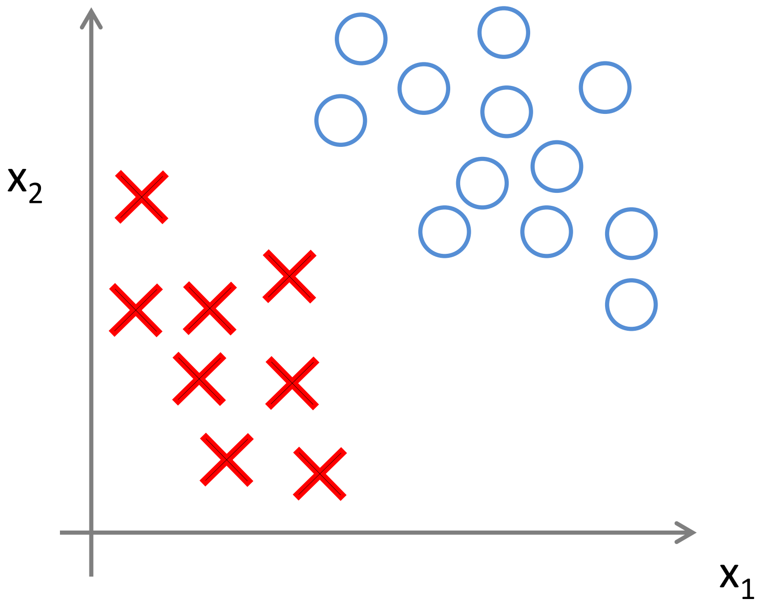
$$\min \frac{1}{2} \sum_{j=1}^{p} \beta_j^2$$

$s.t.$

$$x^{(i)}\beta \geq 1 \qquad if \;\; y^{(i)} = 1$$

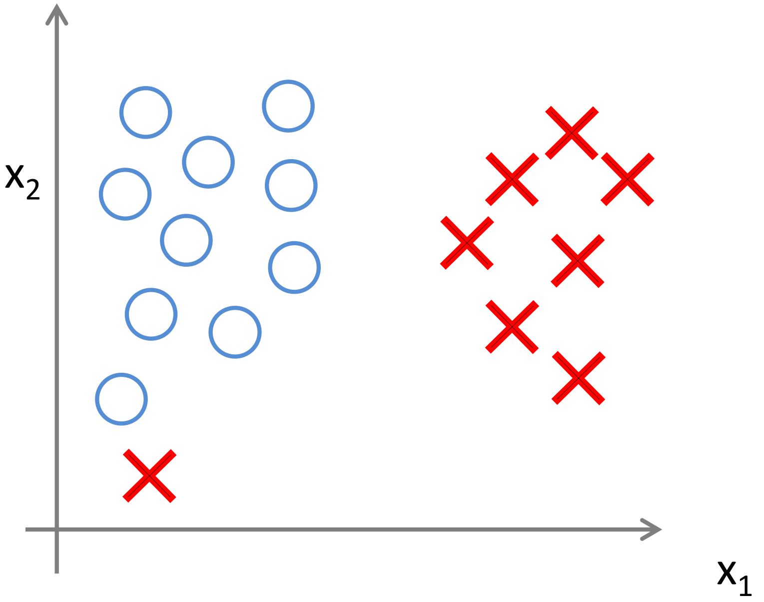
$$x^{(i)}\beta \leq -1 \qquad if \;\; y^{(i)} = 0$$

Linearly separable case



Large margin classifier

Large margin classifier in presence of outliers



SVM with Kernels

Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$,
choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \ldots, l^{(m)} = x^{(m)}$.

Given example x:

$$f_1 = \text{similarity}(x, l^{(1)})$$
$$f_2 = \text{similarity}(x, l^{(2)})$$

Hypothesis: Given x, compute features $f \in \mathbb{R}^{m+1}$

Predict "y=1" if $f^{(i)}\beta \geq 0$

Training:

$$\min_{\beta} C \sum_{i=1}^{n} \left[y^{(i)} cost_1(f^{(i)}\beta) + (1 - y^{(i)}) cost_0(f^{(i)}\beta) \right] + \frac{1}{2} \sum_{j=1}^{n} \beta_j^2$$

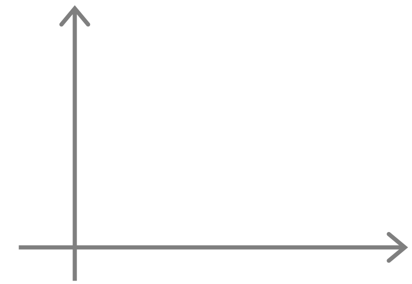
SVM parameters:

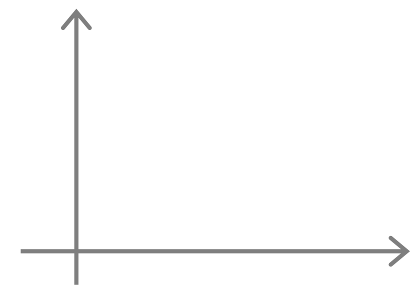
Large C: Lower bias, high variance

Small C: Higher bias, low variance.

Large σ^2: Features f_i vary more smoothly.

Higher bias, lower variance.





Small σ^2: Features f_i vary less smoothly.

Lower bias, higher variance.

Using an SVM

Use SVM software package to solve for parameters β.

Need to specify:

Choice of parameter C.

Choice of kernel (similarity function):

Linear kernel (no kernel)

Predict "y = 1" if $x\beta \geq 0$

- Large number of features
- small training set

Gaussian kernel:

$$f_i = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right)$$, where $l^{(i)} = x^{(i)}$.

- Small number of features
- Large training set

Need to choose σ^2.

Kernel (similarity) functions:

```
function f = kernel(x1,x2)
```

$$f = \exp\left(-\frac{||\,\mathbf{x1} - \mathbf{x2}\,||^2}{2\sigma^2}\right)$$

```
return
```

Note: Do perform feature scaling before using the Gaussian kernel.

Note of choices of kernel

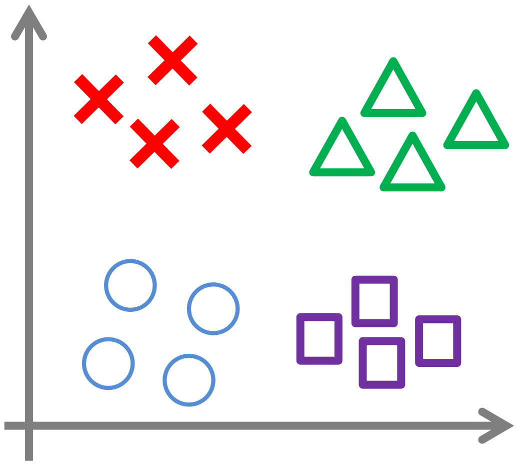
Other choices of kernel

Note: Not all similarity functions $\text{similarity}(x, l)$ make valid kernels. (Need to satisfy technical condition called "Mercer's Theorem" to make sure SVM packages' optimizations run correctly, and do not diverge).

Many off-the-shelf kernels available:

- Polynomial kernel: $(xl + constant)^2$
- More esoteric: String kernel, chi-square kernel, histogram intersection kernel, …

Multi-class classification



$$y \in \{1, 2, 3, \ldots, k\}$$

Many SVM packages already have built-in multi-class classification functionality.

Otherwise, use one-vs.-all method. (Train k SVMs, one to distinguish $y = i$ from the rest, for $l = 1, \ldots, k$), get $\beta^{(1)}, \ldots, \beta^{(k)}$

Pick class l with largest $x\beta^{(l)}$

Logistic regression vs. SVMs

p = number of features ($x \in \mathbb{R}^{n+1}$), n = number of training examples

If p is large (relative to n) (i.e. p=10,000, n<1000) :

Use logistic regression, or SVM without a kernel ("linear kernel")

If p is small (1-1000), n is intermediate (10-10,000):

Use SVM with Gaussian kernel

If p is small (1-1000), n is large (>100,000):

(manually) create/add more features, then use logistic regression or SVM without a kernel

Neural network likely to work well for most of these settings, but may be slower to train.