Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

1)
$x_1 = (0,1,0)$ y_1 = good
$x_2 = (1,0,1)$ y_2 = bad
$x_3 = (1,1,1)$ y_3 = ugly
$x_4 = (1,0,0)$ y_4 = bad
$x_5 = (0,0,1)$ y_5 = good

$w_{good} = (0,0,0)$ $w_{bad} = (0,0,0)$ $w_{ugly} = (0,0,0)$

x_1:

class γ	$w_\gamma \cdot x$	
good	0	[$w_{good} \cdot x = (0, 0, 0) \cdot (0,1,0)$]
bad	0	[$w_{bad} \cdot x = (0, 0, 0) \cdot (0,1,0)$]
ugly	0	[$w_{ugly} \cdot x = (0, 0, 0) \cdot (0,1,0)$]

yhat = good = y*, no update

x_2:

class γ	$w_\gamma \cdot x$	
good	0	[$w_{good} \cdot x = (0, 0, 0) \cdot (1,0,1)$]
bad	0	[$w_{bad} \cdot x = (0, 0, 0) \cdot (1,0,1)$]
ugly	0	[$w_{ugly} \cdot x = (0, 0, 0) \cdot (1,0,1)$]

yhat = good ≠ bad = y*, update:
$w_{bad} = w_{bad} + x = (0,0,0) + (1,0,1) = (1,0,1)$
$w_{good} = w_{good} – x = (0,0,0) – (1,0,1) = (-1,0,-1)$

x_3:

class γ	$w_\gamma \cdot x$	
good	-2	[$w_{good} \cdot x = (-1, 0, -1) \cdot (1,1,1)$]
bad	2	[$w_{bad} \cdot x = (1, 0, 1) \cdot (1,1,1)$]
ugly	0	[$w_{ugly} \cdot x = (0, 0, 0) \cdot (1,1,1)$]

yhat = bad ≠ ugly = y*, update:
$w_{ugly} = w_{ugly} + x = (0,0,0) + (1,1,1) = (1,1,1)$
$w_{bad} = w_{bad} – x = (1,0,1) – (1,1,1) = (0,-1,0)$

x_4:

class γ	$w_\gamma \cdot x$	
good	-1	[$w_{good} \cdot x = (-1,0,-1) \cdot (1,0,0)$]
bad	0	[$w_{bad} \cdot x = (0,-1,0) \cdot (1,0,0)$]
ugly	1	[$w_{ugly} \cdot x = (1,1,1) \cdot (1,0,0)$]

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

yhat = ugly ≠ bad = y*, update:
$w_{bad} = w_{bad} + x = (0,-1,0) + (1,0,0) = (1,-1,0)$

$w_{ugly} = w_{ugly} - x = (1,1,1) - (1,0,0) = (0,1,1)$

x_5:

class γ	$w_\gamma \cdot x$	
good	-1	$[w_{good} \cdot x = (-1,0,-1) \cdot (0,0,1)]$
bad	0	$[w_{bad} \cdot x = (1,-1,0) \cdot (0,0,1)]$
ugly	1	$[w_{ugly} \cdot x = (0,1,1) \cdot (0,0,1)]$

yhat = ugly ≠ good = y*, update:
$w_{good} = w_{good} + x = (-1,0,-1) + (0,0,1) = (-1,0,0)$
$w_{ugly} = w_{ugly} - x = (0,1,1) - (0,0,1) = (0,1,0)$

Weights at the end of first iteration:

$w_{good} = (-1,0,0)$ $w_{bad} = (1,-1,0)$ $w_{ugly} = (0,1,0)$

2)
a)
$x_1 = (0,1,0)$, y_1 = good = +1
$x_2 = (0,1,0)$, y_2 = bad = -1
$x_3 = (0,1,0)$, y_3 = good = +1
$x_4 = (0,1,0)$, y_4 = bad = -1
$x_5 = (0,1,0)$, y_5 = good = +1

$$a = \sum_m \alpha_m \Phi(x_m) \cdot \Phi(\mathrm{x_n})$$

$x_1 = (0,1,0)$
$a = 0((0,1,0) \cdot (0,1,0))$
$+ 0((1,0,1) \cdot (0,1,0))$
$+ 0((1,1,1) \cdot (0,1,0))$
$+ 0((1,0,0) \cdot (0,1,0))$
$+ 0((0,0,1) \cdot (0,1,0))$
$= 0 \leq 0$
$\Rightarrow \alpha_1 = \alpha_1 + y_1$
$= 0 + 1$
$= 1$

$x_2 = (1,0,1)$
$a = 1((0,1,0) \cdot (1,0,1))$

$+ 0((1,0,1) \cdot (1,0,1))$
$+ 0((1,1,1) \cdot (1,0,1))$
$+ 0((1,0,0) \cdot (1,0,1))$
$+ 0((0,0,1) \cdot (1,0,1))$
$= 0 \leq 0$
$\Rightarrow \alpha_2 = \alpha_2 + y_2$
$= 0 + (-1)$
$= (-1)$

$x_3 = (1,1,1)$
$a = 1((0,1,0) \cdot (1,1,1))$
$+ (-1)((1,0,1) \cdot (1,1,1))$
$+ 0((1,1,1) \cdot (1,1,1))$
$+ 0((1,0,0) \cdot (1,1,1))$
$+ 0((0,0,1) \cdot (1,1,1))$
$=1-2$
$= -1 \leq 0$
$\Rightarrow \alpha_3 = \alpha_3 + y_3$
$= 0 + 1$
$= 1$

$x_4 = (1,0,0)$
$a = 1((0,1,0) \cdot (1,0,0))$
$+ (-1)((1,0,1) \cdot (1,0,0))$
$+ 1((1,1,1) \cdot (1,0,0))$
$+ 0((1,0,0) \cdot (1,0,0))$
$+ 0((0,0,1) \cdot (1,0,0))$
$=0+(-1)+1$
$= 0 \leq 0$
$\Rightarrow \alpha_4 = \alpha_4 + y_4$
$= 0 + (-1)$
$= (-1)$

$x_5 = (0,0,1)$
$a = 1((0,1,0) \cdot (0,0,1))$
$+ (-1)((1,0,1) \cdot (0,0,1))$
$+ 1((1,1,1) \cdot (0,0,1))$
$+ 0((1,0,0) \cdot (0,0,1))$
$+ 0((0,0,1) \cdot (0,0,1))$
$=0+(-1)+1+0+0$
$= 0 \leq 0$
$\Rightarrow \alpha_5 = \alpha_5 + y_5$
$= 0 + 1$

= 1

$\alpha_1 = 1, \alpha_2 = -1, \alpha_3 = 1, \alpha_4 = -1, \alpha_5 = 1$

b)
$k(x,x') = (x \cdot x'+1)^3$
$\Rightarrow (x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1$
x_1:
$0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')($
$x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 +$
$3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 +$
$3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (0,1,0), x_1' = (0,1,0)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,0,1), x_1' = (0,1,0)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,1,1), x_1' = (0,1,0)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,0,0), x_1' = (0,1,0)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (0,0,1), x_1' = (0,1,0)\}$

$= 0+0+0+0+0$
$= 0 \leq 0$
$\alpha_1 = \alpha_1 + y_1 = 0 + 1$
$\alpha_1 = 1$

x_2:

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

$1((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (0,1,0), x_1' = (1,0,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,0,1), x_1' = (1,0,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,1,1), x_1' = (1,0,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,0,0), x_1' = (1,0,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (0,0,1), x_1' = (1,0,1)\}$

$= 1+0+0+0+0$
$= 1 > 0$
$\alpha_2 = \alpha_2 = 0$
$\alpha_2 = 0$

x_3:
$1((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (0,1,0), x_1' = (1,1,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,0,1), x_1' = (1,1,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$

$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (1,1,1),\ x_1' = (1,1,1)\}$

$+\ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (1,0,0),\ x_1' = (1,1,1)\}$

$+\ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (0,0,1),\ x_1' = (1,1,1)\}$

$= 5+0+0+0+0$
$= 5 > 0$
$\alpha_3 = \alpha_3 = 0$
$\alpha_3 = 0$

x_4:
$1((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')($
$x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 +$
$3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 +$
$3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (0,1,0),\ x_1' = (1,0,0)\}$

$+\ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (1,0,1),\ x_1' = (1,0,0)\}$

$+\ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (1,1,1),\ x_1' = (1,0,0)\}$

$+\ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (1,0,0),\ x_1' = (1,0,0)\}$

$+\ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 +$
$6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') +$
$(x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') +$
$(x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)\ \{x_1 = (0,0,1),\ x_1' = (1,0,0)\}$

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

$= 0+0+0+0+0$
$= 0 \leq 0$
$\alpha_4 = \alpha_4 + y_4 = 0 + (-1)$
$\alpha_4 = -1$

x_5:
$1((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (0,1,0), x_1' = (0,0,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,0,1), x_1' = (0,0,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,1,1), x_1' = (0,0,1)\}$

$+ (-1)((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (1,0,0), x_1' = (0,0,1)\}$

$+ 0((x_1,x_1')^3 + 3(x_1,x_1')^2(x_2x_2') + 3(x_1,x_1')^2(x_3x_3') + 3(x_1,x_1')^2 + 3(x_1,x_1')(x_2x_2')^2 + 6(x_1,x_1')(x_2x_2')(x_3x_3') + 6(x_1,x_1')(x_2x_2') + 3(x_1,x_1')(x_3x_3')^2 + 6(x_1,x_1')(x_3x_3') + 3(x_1,x_1') + (x_2,x_2')^3 + 3(x_2,x_2')^2(x_3,x_3') + 3(x_2,x_2')^2 + 3(x_2,x_2')(x_3,x_3')^2 + 6(x_2,x_2')(x_3,x_3') + 3(x_2,x_2') + (x_3,x_3')^3 + 3(x_3,x_3')^2 + 3(x_3,x_3') + 1)$ $\{x_1 = (0,0,1), x_1' = (0,0,1)\}$

$= 0+0+0+0+0$
$= 0 \leq 0$
$\alpha_5 = \alpha_5 + y_5 = 0 + 1$
$\alpha_5 = 1$

$\alpha_1 = 1, \alpha_2 = 0$ $\alpha_3 = 0, \alpha_4 = -1, \alpha_5 = 1$

3)
The distance between two points in high dimensional space can be written as:

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

$$D(x,z) = \sqrt{\sum_{i=1}^{d}(x_i - z_i)^2}$$

It is given that:

$$\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d} x_i - \frac{1}{\sqrt{d}}\sum_{i=1}^{d} z_i\right)^2 \le \sum_{i=1}^{d}(x_i - z_i)^2$$

Which can be written as:

$$\left(\frac{1}{\sqrt{d}}\left(\sum_{i=1}^{d} x_i - \sum_{i=1}^{d} z_i\right)\right)^2 \le \sum_{i=1}^{d}(x_i - z_i)^2$$

Which is:

$$\left(\frac{1}{\sqrt{d}}\left(\sum_{i=1}^{d} x_i - \sum_{i=1}^{d} z_i\right)\right)^2 \le \left(D(x,z)\right)^2$$

By taking the root of both sides:

$$\frac{1}{\sqrt{d}}\left(\sum_{i=1}^{d} x_i - \sum_{i=1}^{d} z_i\right) \le D(x,z)$$

Which is:

$$\frac{1}{\sqrt{d}}\left(\sum_{i=1}^{d} x_i - \sum_{i=1}^{d} z_i\right) \le \sqrt{\sum_{i=1}^{d}(x_i - z_i)^2}$$

Which will save one square and one root operation per set of points.

4)

If each leaf node is one if-then rule, then no, it is not possible because the leaf node can based on a decision that was made because of another decision that is not reflected in the single if-then-rule and multiple labels can be represented by multiple leaf nodes.

For example, given:

(a,b,c) = label
(0,0,0) = Bad
(0,0,1) = Bad
(0,1,0) = Good
(1,0,0) = Bad
(0,1,1) = Bad
(1,0,1) = Good
(1,1,0) = Bad

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

(1,1,1) = bad

Which will generalize as: if a = c ≠ b, then good, else bad.

But, the if-then-rules at each leaf will be:
If a = b, then bad
If a ≠ b, then good
If a ≠ c, then bad

This could cause the constructed tree to declare a good when it finds that a ≠ b without considering if a = c which could lead to a misclassification. Some rules depend on other rules and cannot be regarded alone without the necessary pre-qualifying rule applied so a set of final rules may not build into a successful decision tree.

This is only if each rule has only one simple if-then. If each rule allows multiple criteria that does account for additional variable evaluations, such as, if a = c and a ≠ b, then yes, it will work.

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

6)

In “Hidden Technical Debt in Machine Learning Systems”, Sculley et al. analyze the results of quickly developing and deploying ML systems by examining the long-term effects of their maintenance.

Since ML systems are used in cases where abstraction boundaries between data are unclear and often must be merged to produce good results, this makes improvements difficult because of the interdependencies. Using existing solutions to one dataset as a basis for the solution to a new, similar dataset can result in a complex model that requires more time and effort to analyze and improve. Solutions that provide predictions are also often used in other systems without explicitly recording them so modifications to a system can cause a downstream failure in multiple systems that were relying on the original solution method and are not built to accommodate new, improved solutions.

Fast system development of ML systems can lead to the introduction of solutions that offer little benefit, sometimes at the expense of high processing time, to quickly solve a problem and enable fast deployment. If such systems are analyzed, they could be re-written to be more efficient or even eliminated altogether to improve the overall system.

ML systems sometimes use feedback loops that will influence their behavior. This can be directly, from the system itself, or indirectly from related systems that themselves have data modifying feedback loops. Such loops make it difficult to analyze the future behavior of the system.

ML systems frequently utilize existing libraries and open-source packages to manage the data which results in “Glue Code” which is code that is specific to one package to format the data that can’t be changed for any future improvements of the overall system and becomes a limiting factor.

Early development of an ML system will often utilize experimental code that is rarely removed and can cause bloat in a system making it hard to maintain or to transfer to a new team.

There are underlying problems such as the use of generic datatypes to describe specific data that could be more robust if it is more accurately described. Fast development will sometimes cause developers to use existing packages to solve specific problems even if those packages are in different languages resulting in software that uses more than one coding language. This is difficult to maintain. Developers will also sometimes become over-reliant on early prototypes without focusing on long-term system integration ultimately causing unstable systems.

Early system development can often disregard the necessary system configurations resulting in finished systems that require large, specific configuration files where any incorrect configuration setting can cause a system failure.

Since ML systems deal with data in and from the outside world, if developers used fixed thresholds, they may not translate well or at all to new, shifting data.

To be effective, ML systems require live monitoring of behavior in real time. If the real world behavior experiences a sudden shift, systems need to be able to detect this and respond accordingly. For systems with costly results of errors, limits need to be set and enforced such as in automated bidding systems. If a system depends on data from another system, that system must be subject to the same monitoring with the needs of the affected system(s) taken in to consideration.

Input data should be tested periodically for accuracy, this takes time and processing resources but is necessary for good system maintenance. As a system grows to process multiple models with different configurations and priorities, the system must be able to manage its resources accordingly. System maintainers should focus on deleting unnecessary features and reducing complexity just as much as they focus on system improvements to maintain the overall system health.

In "The ML Test Score", Eric Breck, et al. discuss that the reliability of ML systems is critical but they are more challenging to test than coded systems due to their dependency on data that is unknown ahead of time. Based on years of practical work at Google, they propose a test rubric of 28 individual tests that can be used to calculate a single score to indicate if an ML system is ready to go live.

The tests that they propose comprise four categories: Tests for features and data, Tests for model development, Tests for ML infrastructure, and Monitoring tests for ML. Each test has a variety of criteria to test to.

The tests for features and data evaluates whether all of the features are beneficial and that their processing time is reasonable and not excessive. Included in this test is an elementary test that all of the code is tested. It appears then that some ML system developers choose not to test code that they deem "simple" necessitating the inclusion of this criteria.

The tests for model development evaluate if the specifications for the models themselves are properly reviewed, and that all hyperparameters have been properly adjusted so that the offline testing is a good match to real-world application. This test includes trying to determine if the model is too complex and that a simpler solution exists and that it includes all necessary elements.

The ML infrastructure tests evaluate the models ability to handle data and make accurate predictions and that it adequately provides for returning to a previously known working version if errors are found.

Monitoring tests are evaluated for live systems to see how stable they are over time.

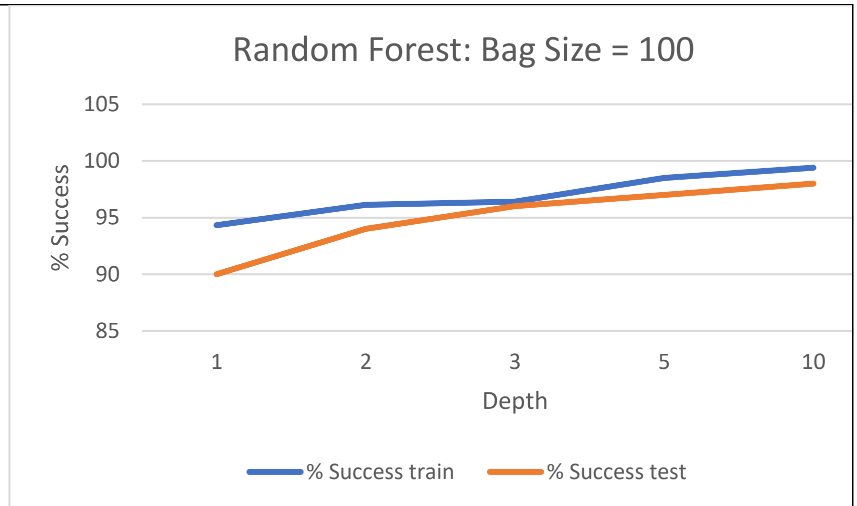
Companies like Google have implemented certification programs for their ML systems that encourage (require) their ML systems to be adequately evaluated to this rubric to demonstrate that they are ready to go live. They are encouraging other ML system developers to adopt similar requirements. As systems grow in complexity and are made to work on growing datasets, adequate testing is crucial for success.

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

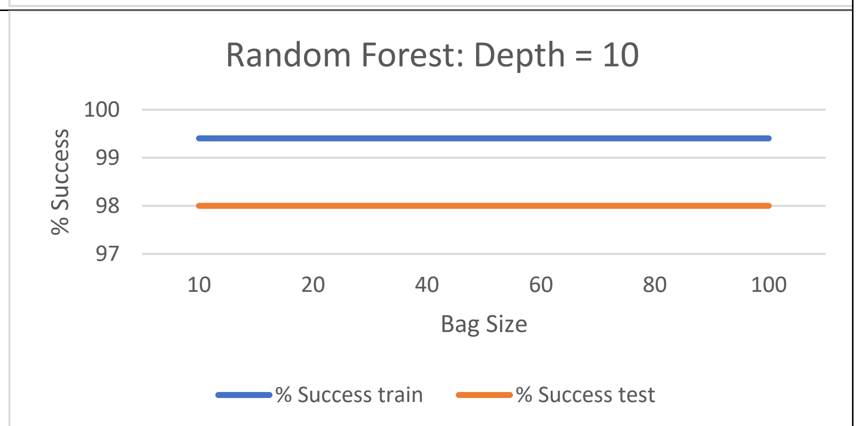
Empirical:

Decision Tree:

Bag Size = 100		
Depth	% Success train	% Success test
1	94.3284	90
2	96.1194	94
3	96.4179	96
5	98.5075	97
10	99.403	98



Depth = 10		
Bag Size	**% Success train**	**% Success test**
10	99.403	98
20	99.403	98
40	99.403	98
60	99.403	98
80	99.403	98
100	99.403	98

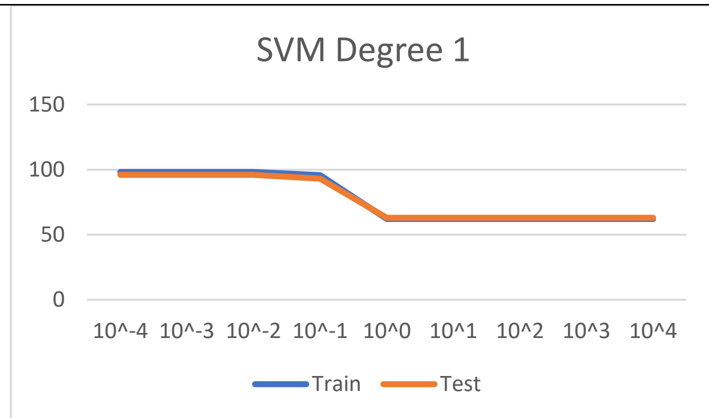


Increasing the tree depth improves successful predictions.
Changing the bag size had no effect on prediction success.

SVM
a)
Degree = 1:

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

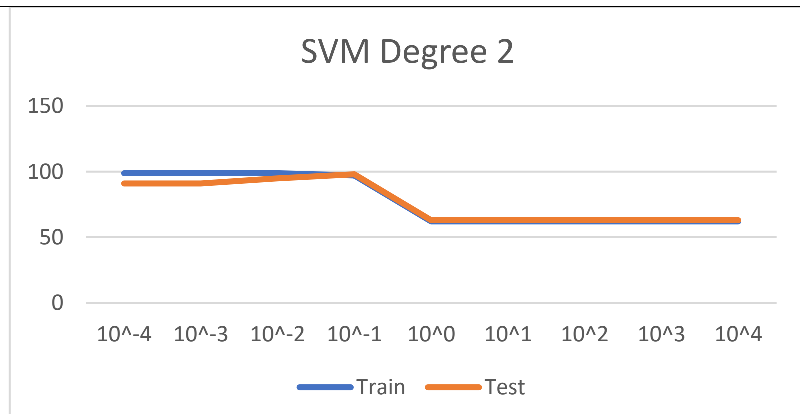
C =	Train	Test
10^-4	98.209	96
10^-3	98.209	96
10^-2	98.209	96
10^-1	95.8209	93
10^0	62.0896	63
10^1	62.0896	63
10^2	62.0896	63
10^3	62.0896	63
10^4	62.0896	63



As C increases, there is a small change from 0.01 to 0.1 and a large change from 0.1 to 1 with no change for values of c > 1. The success of the training data is higher than the testing data for values of c < 1 and the success of testing data is greater than training data for the same range.

Degree = 2:

C =	Train	Test
10^-4	98.806	91
10^-3	98.806	91
10^-2	98.806	95
10^-1	97.0149	98
10^0	62.0896	63
10^1	62.0896	63
10^2	62.0896	63
10^3	62.0896	63
10^4	62.0896	63

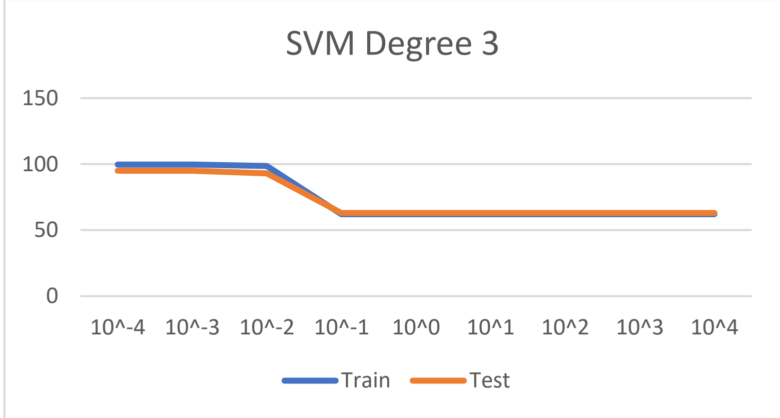


This is curious, I ran the test twice and the testing data is lower at c = 0.0001 and c = 0.001 than at c = 0.01 and it gets higher at c = 0.1 then drops to 63% at c = 1.0 and up to c = 10,000.0

Degree 3:

Marcus Blaisdell
CptS 315
Homework #4
4/16/2018
Professor Doppa

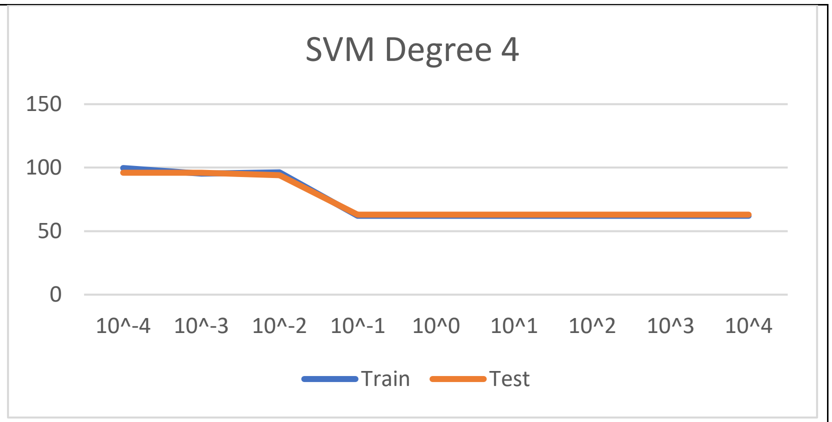
C =	Train	Test
10^-4	99.7015	95
10^-3	99.7015	95
10^-2	98.5075	93
10^-1	62.0896	63
10^0	62.0896	63
10^1	62.0896	63
10^2	62.0896	63
10^3	62.0896	63
10^4	62.0896	63



These results are typical of degree 1, as c increases, success decreases monotonically.

Degree 4:

C =	Train	Test
10^-4	99.7015	96
10^-3	95.2239	96
10^-2	96.4179	94
10^-1	62.0896	63
10^0	62.0896	63
10^1	62.0896	63
10^2	62.0896	63
10^3	62.0896	63
10^4	62.0896	63



Another curiosity, this time with the training data, it decreases from c = 0.01 to c = 0.001 and increases again at c = 0.0001 while the test success rate is monotonic.