CS 221

Problem Set 2

1 (a)

```
w = Association[{"pretty"→0, "good"→0, "bad"→0, "plot"→0, "not"→0, "scenery"→0}];

phi[1] = Association[{"pretty"→1, "good"→1}];
phi[2] = Association[{"bad"→1, "plot"→1}];
phi[3] = Association[{"bad"→1, "not"→1}];
phi[4] = Association[{"pretty"→1, "scenery"→1}];

y[1] = 1;
y[2] = -1;
y[3] = 1;
y[4] = 1;
```

```
assocDot[assoc1_Association, assoc2_Association]:=
    Total[
        Map[
            Function[key,
                Lookup[assoc1, key] * Lookup[assoc2, key, 0]
            ],
            Keys[assoc1]
        ]
    ];

score[w_, phi_] := assocDot[w, phi];

assocMult[assoc_Association, num_?NumericQ]:=
    Module[{assocCopy},
        assocCopy = Association[{}];
        Map[
            Function[key,
                AssociateTo[assocCopy,
                    Rule[key, Lookup[assoc, key] * num]
                ]
            ],
            Keys[assoc]
        ];
        assocCopy
    ];


addMissingFeatures[assoc_Association, keys_List]:=
    Module[{assocCopy},
        assocCopy = Association[{}];
        Map[
            Function[key,
                AssociateTo[assocCopy,
                    Rule[key, Lookup[assoc, key, 0]]
                ]
            ],
            keys
        ];
        assocCopy
    ];



margin[phi_, y_, w_] := assocDot[w, phi]*y


hingeLoss[phi_Association, y_Integer, w_Association]:=
    Max[
        0,
        1 - margin[phi, y, w] (* This is the residual *)
    ];


dHingeLoss[phi_, y_, w_]:=
    Module[{hloss},
        hloss = hingeLoss[phi, y, w];
        Which[
            hloss > 0,
                -assocMult[phi, y],
            hloss <= 0,
                assocMult[phi, 0]
        ]
    ];


updatedWeight[weight_Association, phi_Association, y_Integer, stepSize_] :=
    (weight - stepSize * dHingeLoss[phi, y, weight]);


StochasticGradientDescent[weight_Association, phi_, y_, stepSize_?NumericQ]:=
    Module[{allFeatureNames, phiBuffered, newWeight},
        allFeatureNames = Keys[weight];
        newWeight = weight;
        Table[
            phiBuffered = addMissingFeatures[phi[i], allFeatureNames];
            newWeight = updatedWeight[newWeight, phiBuffered, y[i], stepSize];
            Print["After iteration ", i, ", new weight is ", newWeight];,
            {i, 1, 4}
        ];
        newWeight
    ]
```

```
assocDot[phi[1], phi[4]]
```

```
1
```

```
assocMult[phi[1], 4]
```

```
<|pretty → 4, good → 4|>
```

```
StochasticGradientDescent[w, phi, y, 1]
```

```
After iteration 1, new weight is
 <|pretty → 1, good → 1, bad → 0, plot → 0, not → 0, scenery → 0|>
After iteration 2, new weight is
 <|pretty → 1, good → 1, bad → -1, plot → -1, not → 0, scenery → 0|>
After iteration 3, new weight is
 <|pretty → 1, good → 1, bad → 0, plot → -1, not → 1, scenery → 0|>
After iteration 4, new weight is
 <|pretty → 1, good → 1, bad → 0, plot → -1, not → 1, scenery → 0|>
<|pretty → 1, good → 1, bad → 0, plot → -1, not → 1, scenery → 0|>
```

I (b)

```
w = Association[{"not"→0, "good"→0, "bad"→0}];

phi[1] = Association[{"not"→1, "good"→1}]; y[1] = -1;
phi[2] = Association[{"not"→1, "bad"→1}];  y[2] = 1;
phi[3] = Association[{"bad"→1}];           y[3] = -1;
phi[4] = Association[{"good"→1}];          y[4] = 1;
```

Intuitive proof that no weight vector can be correct 100% of the time with this feature set:

The weight for “bad” must be -1 to make test case 3 work

{not_weight, good_weight, bad_weight} . {0, 0, 1} == -1

bad_weight == -1

The weight for “good” must be 1 to make test case 4 work

{not_weight, good_weight, bad_weight} . {0, 1, 0} == 1

good_weight == 1

So we know cases 3 and 4 work - let’s focus on 1 and 2.

We want these two equations to be true:

w . phi[1] == y[1]
w . phi[2] == y[2]

Simplified with known information, these equations are:

{not_weight, 1, -1} . {1, 1, 0} == -1
{not_weight, 1, -1} . {1, 0, 1} == 1

not_weight + 1 == -1
not_weight - 1 == 1

not_weight == -2
not_weight == 2

These two equations are incompatible, therefore there is no weight vector that can satisfy this set of features.

A simple augmentation that could help this feature vector would be an extra feature that is simply the negative of the number of “not”’s in the sentence. The solution w vector (and new phi values) are:

```
w = Association[{"not"→-2, "good"→1, "bad"→-1, "#not * #bad"→4}];



phi[1] = Association[{"not"→1, "good"→1, "bad"→0, "#not * #bad"→0}];           y[1] = -1;
(* Predicted score is -1, actual score is -1 *)

phi[2] = Association[{"not"→1, "good"→0, "bad"→1, "#not * #bad"→1}];           y[2] = 1;
(* Predicted score is 1, actual score is 1 *)

phi[3] = Association[{"not"→0, "good"→0, "bad"→1, "#not * #bad"→0}];           y[3] = -1;
(* Predicted score is -1, actual score is -1 *)

phi[4] = Association[{"not"→0, "good"→1, "bad"→0, "#not * #bad"→0}];           y[4] = 1;
(* Predicted score is 1, actual score is 1 *)
```

3d

```
maxIters = 11
eta = 0.062

Read 3554 examples from polarity.train
Read 3554 examples from polarity.dev
Iteration 1
Train error: 0.212999
Test error: 0.334271

Iteration 2
Train error: 0.138154
Test error: 0.310917

Iteration 3
Train error: 0.100732
Test error: 0.297693

Iteration 4
Train error: 0.073157
Test error: 0.286438

Iteration 5
Train error: 0.064716
Test error: 0.294598

Iteration 6
Train error: 0.077659
Test error: 0.304446

Iteration 7
Train error: 0.053180
Test error: 0.288689

Iteration 8
Train error: 0.058807
Test error: 0.296567

Iteration 9
Train error: 0.028700
Test error: 0.284187

Iteration 10
Train error: 0.019977
Test error: 0.273495

Iteration 11
Train error: 0.014069
Test error: 0.267867
```

Total Test Error: 0.2679 = 26.79% failure

3e

1.
   a. === home alone goes hollywood , a funny premise until the kids start pulling off stunts not even steven spielberg would know how to do . besides , real movie producers aren't this nice .
   b. Truth: -1, Prediction: 1 [WRONG]
   c. **Response: The prediction thinks the movie is funny and nice, but there are negatives corresponding to each of these words - “until” for funny and “aren’t” for nice. These negatives are missed by the predictor, leading to the incorrect prediction.**
2.
   a. === 'it's painful to watch witherspoon's talents wasting away inside unnecessary films like legally blonde and sweet home abomination , i mean , alabama . '
   b. Truth: -1, Prediction: 1 [WRONG]
   c. **Response: The predictor doesn’t realize that “sweet” and “home” are part of names of things, not words being used in the review.**
3.
   a. === patchy combination of soap opera , low-tech magic realism and , at times , ploddingly sociological commentary .
   b. Truth: -1, Prediction: 1 [WRONG]
   c. **Response: The predictor doesn’t realize that “magic realism” is a proper noun, not 2 descriptor words – it should have tied the use of the word “low-tech” to these two words to see that this was not positive.**
4.
   a. === the best thing i can say about this film is that i can't wait to see what the director does next .
   b. Truth: 1, Prediction: -1 [WRONG]
   c. **Response: The predictor doesn’t understand that the reviewer is being sarcastic because he doesn’t mention anything about the film when he’s talking about the best thing about it. The mere presence of “best” and “film” and “does” make the predictor think the review is automatically positive.**
5.
   a. === . . . standard guns versus martial arts cliche with little new added .
   b. Truth: -1, Prediction: 1 [WRONG]
   c. **Response: The predictor misses the fact that “little” modifies “new” to effectively make it mean the opposite of what it thinks it means.**
6.
   a. === what makes the movie special is its utter sincerity .
   b. Truth: 1, Prediction: -1 [WRONG]

c. **Response: Usually, “utter” comes before something bad, and the predictor doesn’t see it comes before something good this time. Additionally, the predictor thinks the use of the word “movie” is bad because critics often call good movies “films” and bad ones “movies” but this time around it’s different.**

7.

a. === provide[s] nail-biting suspense and credible characters without relying on technology-of-the-moment technique or pretentious dialogue .

b. Truth: 1, Prediction: -1 [WRONG]

c. **Response: The predictor doesn’t see that the words after “without relying on” are going to mean the opposite of what they would mean if they were taken at face-value. So the move is not “pretentious”. Additionally, “nail-biting suspense” should have been a key giveaway that it was positive.**

8.

a. === gangs of new york is an unapologetic mess , whose only saving grace is that it ends by blowing just about everything up .

b. Truth: -1, Prediction: 1 [WRONG]

c. **Response: The predictor is weighing some heavy words here, but it doesn’t realize that “gangs” “new” and “york” are all part of the movie’s title, so they shouldn’t count towards the score. Also, “only saving grace” should have indicated that “grace” was negative, not positive.**

9.

a. === [siegel] and co-writers lisa bazadona and grace woodard have relied too much on convention in creating the characters who surround frankie .

b. Truth: -1, Prediction: 1 [WRONG]

c. **Response: The predictor doesn’t realize that “grace” is the name of a person, not a positive adjective.**

10.

a. === an incoherent jumble of a film that's rarely as entertaining as it could have been .

b. Truth: -1, Prediction: 1 [WRONG]

c. **Response: The predictor doesn’t realize that “entertaining” should be negated by “rarely as”.**

Austin Ray Pset 2 CS 221

Problem 2

a.

$$\text{Loss}(x, y, w) = \text{Loss}_{\text{squared}}(x, y, w)$$

$$= (\text{residual}(x, y, w))^2$$

$$= (\text{prediction} - \text{target})^2$$

$$= (\sigma(w \cdot \text{phi}(x)) - y)^2$$

$$= ((1 + e^{-w \cdot \phi(x)})^{-1} - y)^2$$

$$\boxed{\text{Loss}(x, y, w) = ((1 + e^{-w \cdot \phi(x)})^{-1} - y)^2}$$

HW2

Problem 2 (cont'd)

2. $\text{Loss}(x, y, w) = \left(\left(1+e^{-w\cdot\phi(x)}\right)^{-1} - y\right)^2$

$\nabla_w \text{Loss}(x, y, w)$

$$= 2\left(\left(1+e^{-w\cdot\phi(x)}\right)^{-1} - y\right)\left(-\left(1+e^{-w\cdot\phi(x)}\right)^{-2}\cdot\left(-e^{-w\cdot\phi(x)}\cdot(\phi(x))\right)\right)$$

$$\boxed{\nabla_w \text{Loss}(x, y, w) = 2\cdot\phi(x)\cdot\exp(-w\cdot\phi(x))\cdot\left(1+\exp(-w\cdot\phi(x))\right)^{-2}\cdot\left(\left(1+\exp(-w\cdot\phi(x))\right)^{-1} - y\right)}$$

define: $z = w\cdot\phi(x)$

$$\sigma(z) = \left(1+e^{-z}\right)^{-1}$$

$$\nabla_w \text{Loss}(x, y, w) = 2\cdot\phi(x)\cdot\exp(-z)\cdot(\sigma(z))^2(\sigma(z) - y)$$

$$= 2\phi(x)\left((\sigma(z))^{-1} - 1\right)(\sigma(z))^2(\sigma(z) - y)$$

$$= 2\phi(x)\,\sigma(z)(1-\sigma(z))(\sigma(z) - y)$$

Problem 2 cont'd

C.) $y=0$

want $\|\nabla_w Loss(x,y,w)\| = \text{minimum} = 0$, since norms are non-neg

Occurs when $\nabla_w Loss(x,y,w) = 0$

$$\sigma(z) = \frac{1}{1+e^{-w\cdot\phi(x)}} = 0$$

or

$$1-\sigma(z) = \frac{e^{-w\cdot\phi(x)}}{1+e^{-w\cdot\phi(x)}} = \frac{1}{1+e^{w\cdot\phi(x)}} = 0$$

or

$\sigma(z) - y = \sigma(z) - 0 = \sigma(z) = 0$ (already known)

or

$\phi(x) = 0$ (null case; unhelpful)

so $\frac{1}{1+e^{-w\cdot\phi(x)}} = 0 \Rightarrow 1+e^{-w\cdot\phi(x)} \to \infty$

$$\frac{1}{e^{w\cdot\phi(x)}} \to \infty$$

$$w\cdot\phi(x) \to -\infty$$

Problem 2c cont'd

$\sigma(x) = 0$:

$$\nabla_w Loss = \vec{0} \Rightarrow \|\nabla_w Loss\| = 0$$

$\sigma(x) = \frac{2}{3}$:

$$\nabla_w Loss = 2\Phi(x)\left(\tfrac{2}{3}\right)\left(1-\tfrac{2}{3}\right)\left(\tfrac{2}{3}-0\right)$$

$$= \frac{8}{27}\Phi(x)$$

$$\|\nabla_w Loss\| = \boxed{\frac{8}{27}\|\Phi(x)\| = \|\nabla_w Loss\|_{max}}$$

Problem 2 cont'd

$$\frac{1}{1+e^{w\cdot\phi(x)}} = 0 \Rightarrow w\cdot\phi(x) \rightarrow \infty$$

so $w\cdot\phi(x) \longrightarrow -\infty$

or $w\cdot\phi(x) \longrightarrow \infty$

so $\|w\| \longrightarrow \infty$ will trigger either of these

so basically at least one entry in w $(w_j) \longrightarrow \infty$

Problem 2 (cont'd)

$y = 0$

d.i) $\max\left[\|\nabla_w Loss(x,y,w)\|\right]$

occurs when $\nabla_w^2 Loss(x,y,w) = 0$

let's maximize w/r t W

~~which is equivalent to maximizing by w since $\phi(x) = const$~~

$$\frac{d}{dW}\left[\nabla_w Loss(x,y,w)\right]$$

$$= \frac{d}{dW}\left[2\phi(x)\left((\sigma(z))^2 - (\sigma(z))^3\right)\right]$$

$$= 2\phi(x)\left(2\sigma(z) - 3(\sigma(z))^2\right)\frac{d\sigma(z)}{dW}$$

$$\frac{d\sigma(z)}{dW} = \frac{d}{dW}\left[\frac{1}{1+e^{-z}}\right] = \frac{\phi(x)e^{-z}}{(1+e^{-z})^2} = \frac{\phi(x)(1-\sigma(z))}{\sigma(z)^3} = 0$$

$\sigma(z) \to \infty$ (known to be min)

or

$\phi(x) \to 0$ (null answer)

or

$\sigma(2) = 0 \Leftarrow 2\sigma(z) = 3\sigma(z)^2 \Rightarrow \sigma(z) = \frac{2}{3}$

$$\frac{1}{1+e^{-z}} = \frac{2}{3} \Rightarrow \frac{3}{2} - 1 = e^{-z} \Rightarrow e^{z} = 2$$

$\boxed{w \cdot \phi(x) = 2}$ or $\boxed{\sigma(2) = \frac{2}{3}}$

HW2

Problem 4

a.] $D_{train} = \left\{ \begin{matrix} [1,0] \\ [2,1] \\ [0,0] \\ [0,2] \end{matrix} \right\}$

1. $\mu_1 = [0,-1]$ $\mu_2 = [2,2]$

iter 1:
step 1:

	$\|\|\phi(x_i)-\mu_1\|\|^2$	$\|\|\phi(x_i)-\mu_2\|\|^2$
x_1	$[1,1] \Rightarrow 2$	$[-1,-2] \Rightarrow 5$
x_2	$[2,2] \Rightarrow 8$	$[0,-1] \Rightarrow 1$
x_3	$[0,1] \Rightarrow 1$	$[-2,-2] \Rightarrow 8$
x_4	$[0,3] \Rightarrow 9$	$[2,0] \Rightarrow 4$

$z_1 = 1$ $z_2 = 1$ $z_3 = 1$ $z_4 = 2$

step 2: $\mu_1 = \frac{1}{2}([1,0]+[0,0]) = \boxed{[\frac{1}{2}, 0] = \mu_1}$

$\mu_2 = \frac{1}{2}([2,1]+[0,2]) = \boxed{[1, \frac{3}{2}] = \mu_2}$

Problem 4a. (cont'd)

iter 2:

$$\boxed{z_1 = 1 \quad z_2 = 2 \quad z_3 = 1 \quad z_4 = 2}$$

2. $\mu_1 = [2,0] \quad \mu_2 = [-1,0]$

iter 1:

$z_1 = 1 \quad z_2 = 1 \quad z_3 = 2 \quad z_4 = 2$

$\mu_1 = \frac{1}{2}([3,1]) = \left[\frac{3}{2}, \frac{1}{2}\right] = \mu_1$

$\mu_2 = \frac{1}{2}([0,2]) = [0,1] = \mu_2$

iter 2:

$$\boxed{z_1 = 1 \quad z_2 = 1 \quad z_3 = 2 \quad z_4 = 2}$$