Logistic Regression
to
Perceptron

Logistic Regression

- (Log) Linear Model – similar to Naïve Bayes
- Doesn’t assume features are independent
- Correlated features don’t “double count”

NB vs. LR

- Both compute the dot product

- NB: sum of log probabilities

- LR: logistic function

NB vs. LR: Parameter Learning

- Naïve Bayes:
  - Learn conditional probabilities **independently** by counting

- Logistic Regression:
  - Learn weights **jointly**

Features for movie review

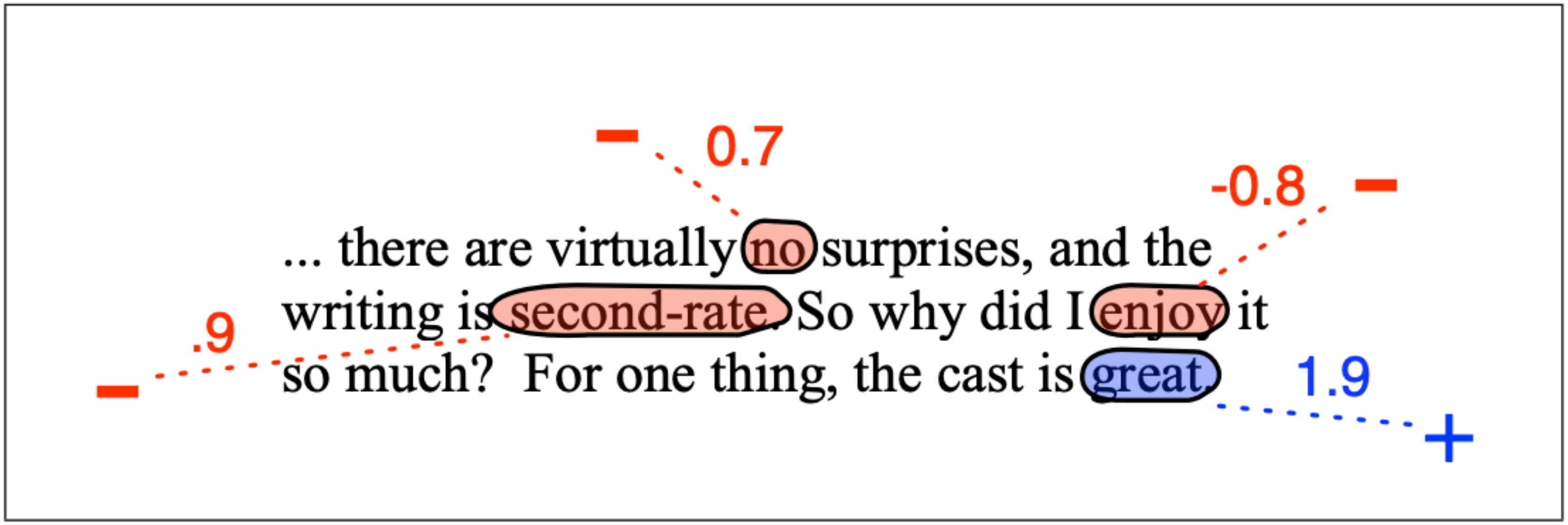
$$f_1(c,x) = \begin{cases} 1 & \text{if “great”} \in x \ \& \ c = + \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(c,x) = \begin{cases} 1 & \text{if “second-rate”} \in x \ \& \ c = - \\ 0 & \text{otherwise} \end{cases}$$

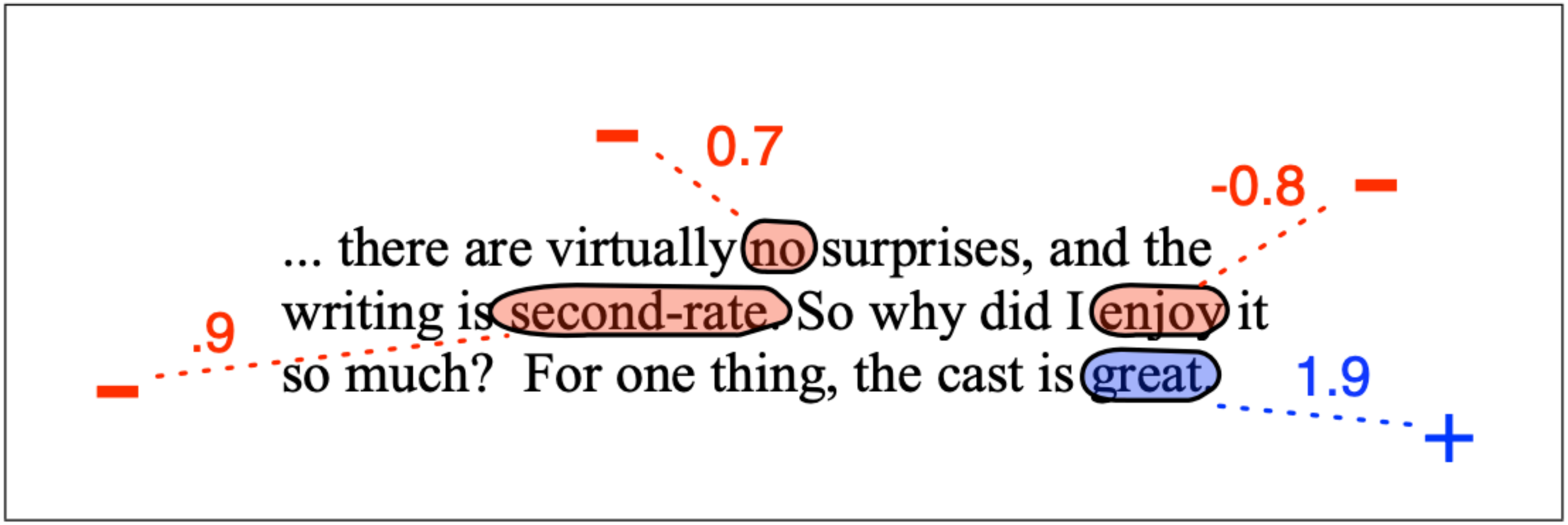
$$f_3(c,x) = \begin{cases} 1 & \text{if “no”} \in x \ \& \ c = - \\ 0 & \text{otherwise} \end{cases}$$

$$f_4(c,x) = \begin{cases} 1 & \text{if “enjoy”} \in x \ \& \ c = - \\ 0 & \text{otherwise} \end{cases}$$

Features for movie review



Features for movie review



$$P(+|x) = \frac{e^{1.9}}{e^{1.9} + e^{.9}e^{.7}e^{-.8}} = .82$$

$$P(-|x) = \frac{e^{.9}e^{.7}e^{.8}}{e^{1.9} + e^{.9}e^{.7}e^{-.8}} = .18$$

Q: what parameters should we choose?

- What is the right value for the weights?

- Maximum Likelihood Principle:
  - Pick the parameters that maximize the probability of the data

Maximum Likelihood Estimation

- Unfortunately there is no closed form solution
  – (like there was with naïve bayes)
- Solution:
  – Iteratively climb the log-likelihood surface through the derivatives for each weight
- Luckily, the derivatives turn out to be nice

Gradient ascent

Loop While not converged:

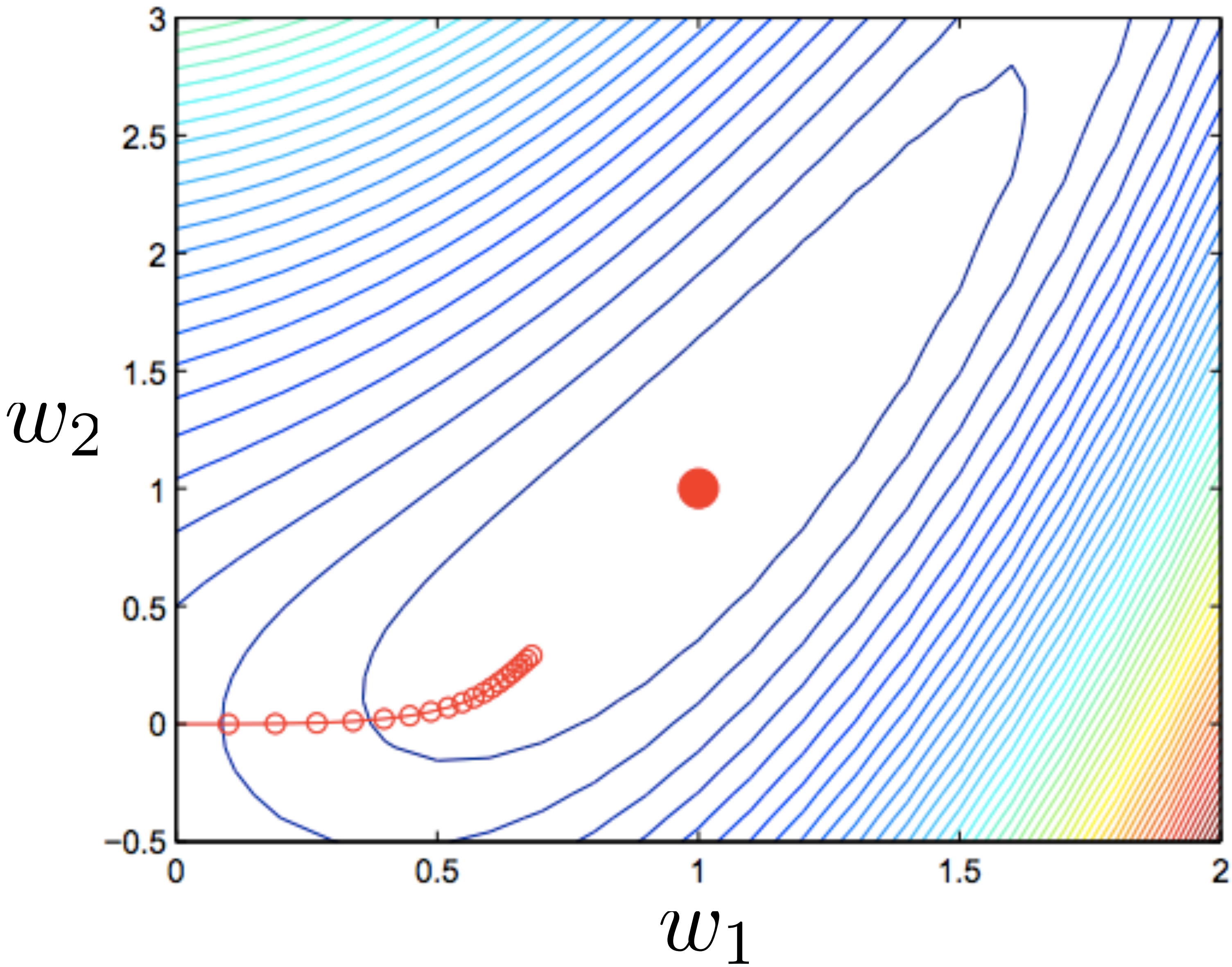
For all features **j**, compute and add derivatives

$$w_j^{\text{new}} = w_j^{\text{old}} + \eta \frac{\partial}{\partial w_j} \mathcal{L}(w)$$

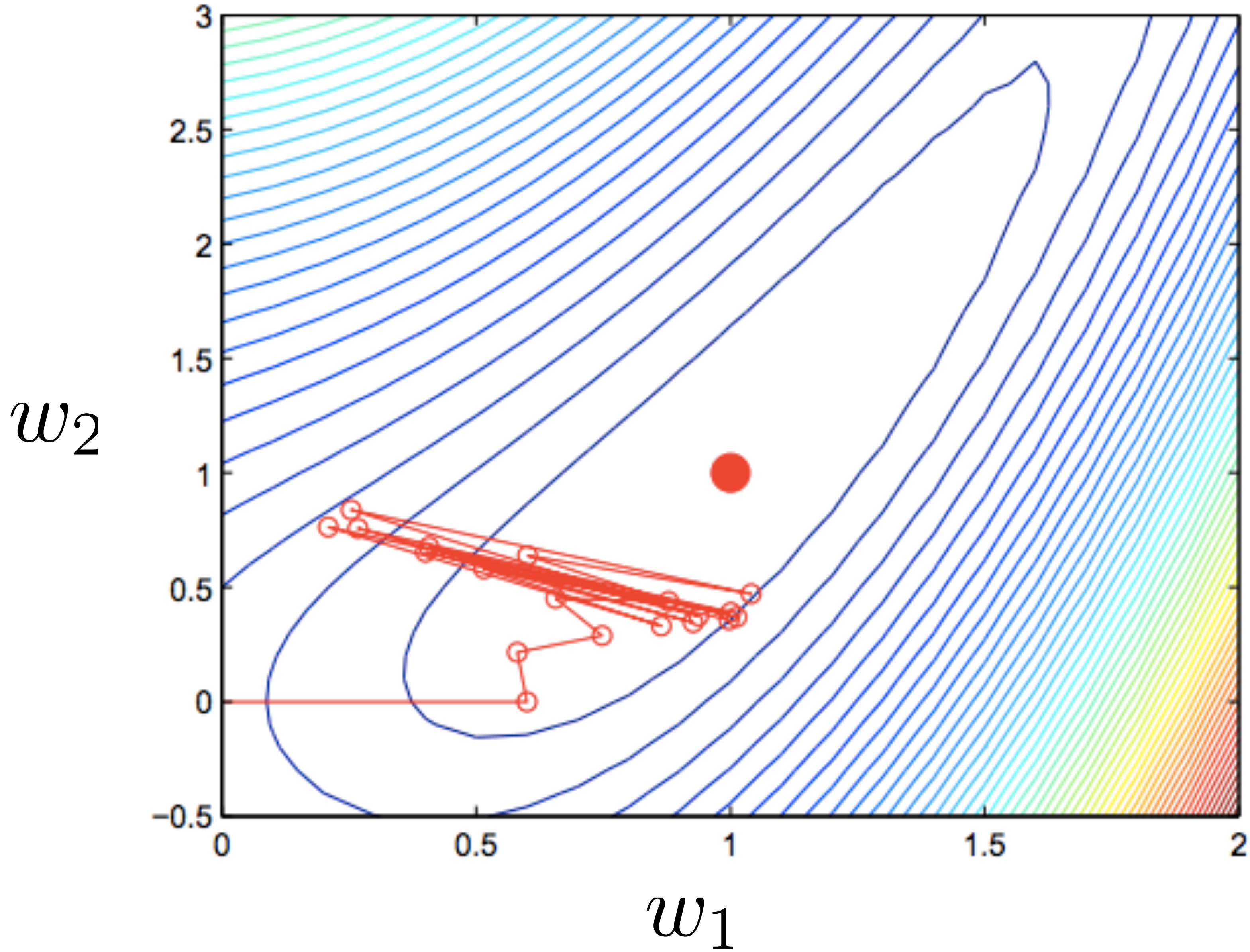
$\mathcal{L}(w)$: Training set log-likelihood

$\left(\frac{\partial \mathcal{L}}{\partial w_1}, \frac{\partial \mathcal{L}}{\partial w_2}, \ldots, \frac{\partial \mathcal{L}}{\partial w_n} \right)$: Gradient vector

Gradient ascent



Gradient ascent



Derivative of Sigmoid

$$\frac{ds(x)}{dx} = \frac{1}{1+e^{-x}}$$

$$= \left(\frac{1}{1+e^{-x}}\right)^2 \frac{d}{dx}(1+e^{-x})$$

$$= \left(\frac{1}{1+e^{-x}}\right)^2 e^{-x}(-1)$$

$$= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right)(-e^{-x})$$

$$= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{-e^{-x}}{1+e^{-x}}\right)$$

$$= s(x)(1-s(x))$$

LR Gradient

$$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_i (y_i - p_i) x_j$$

j -> iterating over features

i -> iterating over training examples

Perceptron Algorithm

- Algorithm is Very similar to logistic regression
- Not exactly computing gradients

Perceptron Algorithm

- Algorithm is Very similar to logistic regression
- Not exactly computing gradients

```
Initialize weight vector w = 0
Loop for K iterations
    Loop For all training examples x_i
        y_i’ = sign(w * x_i)
        if y_i’!= y_i
            w += (y_i - y_i’) * x_i
```

Perceptron Notes

- Guaranteed to converge if the data is linearly separable
- Only hyperparameter is maximum number of iterations
- Parameter averaging

Differences: LR vs. Perceptron

- Batch Learning vs. Online learning

- Perceptron doesn’t always make updates

Online Learning (perceptron)

- Rather than making a full pass through the data, compute gradient and update parameters after each training example.

- Gradients will be less accurate, but the overall effect is to move in the right direction

- Often works well and converges faster than batch learning

MultiClass Classification

- Q: what if we have more than 2 categories?
  - Sentiment: Positive, Negative, Neutral
  - Document topics: Sports, Politics, Business, Entertainment, …

MultiClass Classification

- Q: what if we have more than 2 categories?
  - Sentiment: Positive, Negative, Neutral
  - Document topics: Sports, Politics, Business, Entertainment, …
- Could train a separate logistic regression model for each category…

Log-Linear Models

$$P(y|x) \propto e^{w \cdot f(d,y)}$$

$$P(y|x) = \frac{1}{Z(w)} e^{w \cdot f(d,y)}$$

MultiClass Logistic Regression

$$P(y|x) \propto e^{w \cdot f(d,y)}$$

$$P(y|x) = \frac{1}{Z(w)} e^{w \cdot f(d,y)}$$

$$\boxed{P(y|x) = \frac{e^{w \cdot f(d,y)}}{\sum_{y' \in Y} e^{w \cdot f(d,y')}}}$$

MultiClass Logistic Regression

- Binary logistic regression:
  - We have one feature vector that matches the size of the vocabulary
- Multiclass in practice:
  - one weight vector for each category

w_{pos} $\quad\quad$ w_{neg} $\quad\quad$ w_{neut}

MultiClass Logistic Regression

- Binary logistic regression:
  - We have one feature vector that matches the size of th
- Mul
  - o

Can represent this in practice with one giant weight vector and repeated features for each category.

w_{pos} w_{neg} w_{neut}

Maximum Likelihood Estimation

$$w_{\text{MLE}} = \text{argmax}_w \log P(y_1, \ldots, y_n | x_1, \ldots, x_n; w)$$

$$= \text{argmax}_w \sum_i \log P(y_i | x_i; w)$$

$$= \text{argmax}_w \sum_i \log \frac{e^{w \cdot f(x_i, y_i)}}{\sum_{y' \in Y} e^{w \cdot f(x_i, y_i)}}$$

Multiclass Leaning

LR : $$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^{D} f_j(y_i, d_i) - \sum_{i=1}^{D} \sum_{y \in Y} f_j(y, d_i) P(y|d_i)$$

Multiclass Leaning

LR : $$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^{D} f_j(y_i, d_i) - \sum_{i=1}^{D} \sum_{y \in Y} f_j(y, d_i) P(y|d_i)$$

Perceptron : $$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^{D} f_j(y_i, d_i) - \sum_{i=1}^{D} f_j(\arg\max_{y \in Y} P(y|d_i), d_i)$$

Multiclass Leaning

LR : $$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^{D} f_j(y_i, d_i) - \sum_{i=1}^{D} \boxed{\sum_{y \in Y}} f_j(y, d_i) P(y|d_i)$$

Perceptron : $$\frac{\partial \mathcal{L}}{\partial w_j} = \sum_{i=1}^{D} f_j(y_i, d_i) - \sum_{i=1}^{D} f_j(\boxed{\arg\max_{y \in Y}} P(y|d_i), d_i)$$

MultiClass Perceptron Algorithm

$$Initialize\ weight\ vector\ w = 0$$

$$Loop\ for\ K\ iterations$$

$$Loop\ For\ all\ training\ examples\ x_i$$

$$y_{pred} = argmax_y(w_y * x_i)$$

$$if\ y_{pred}\ != \ y_i$$
$$w_{y_{gold}} += x_i$$
$$w_{y_{pred}} -= x_i$$

Q: what if there are only 2 categories?

$$P(y = j|x_i) = \frac{e^{w_j \cdot x_i}}{\sum_k e^{w_k \cdot x_i}}$$

$$P(y = 1|x) = \frac{e^{w_1 \cdot x}}{e^{w_0 \cdot x + w_1 \cdot x - w_1 \cdot x} + e^{w_1 \cdot x}}$$

$$P(y = 1|x) = \frac{e^{w_1 \cdot x}}{e^{w_0 \cdot x - w_1 \cdot x} e^{w_1 \cdot x} + e^{w_1 \cdot x}}$$

$$P(y = 1|x) = \frac{e^{w_1 \cdot x}}{e^{w_1 \cdot x}(e^{w_0 \cdot x - w_1 \cdot x} + 1)}$$

$$P(y = 1|x) = \frac{1}{e^{w_0 \cdot x - w_1 \cdot x} + 1}$$

Q: what if there are only 2 categories?

$$P(y = 1|x) = \frac{1}{e^{-w' \cdot x} + 1}$$

Sigmoid (logistic) function

$$w' = w_1 - w_0$$

Next

- Regularization
- Markov Model