Linear Classifiers

CS6120: Natural Language Processing
Northeastern University

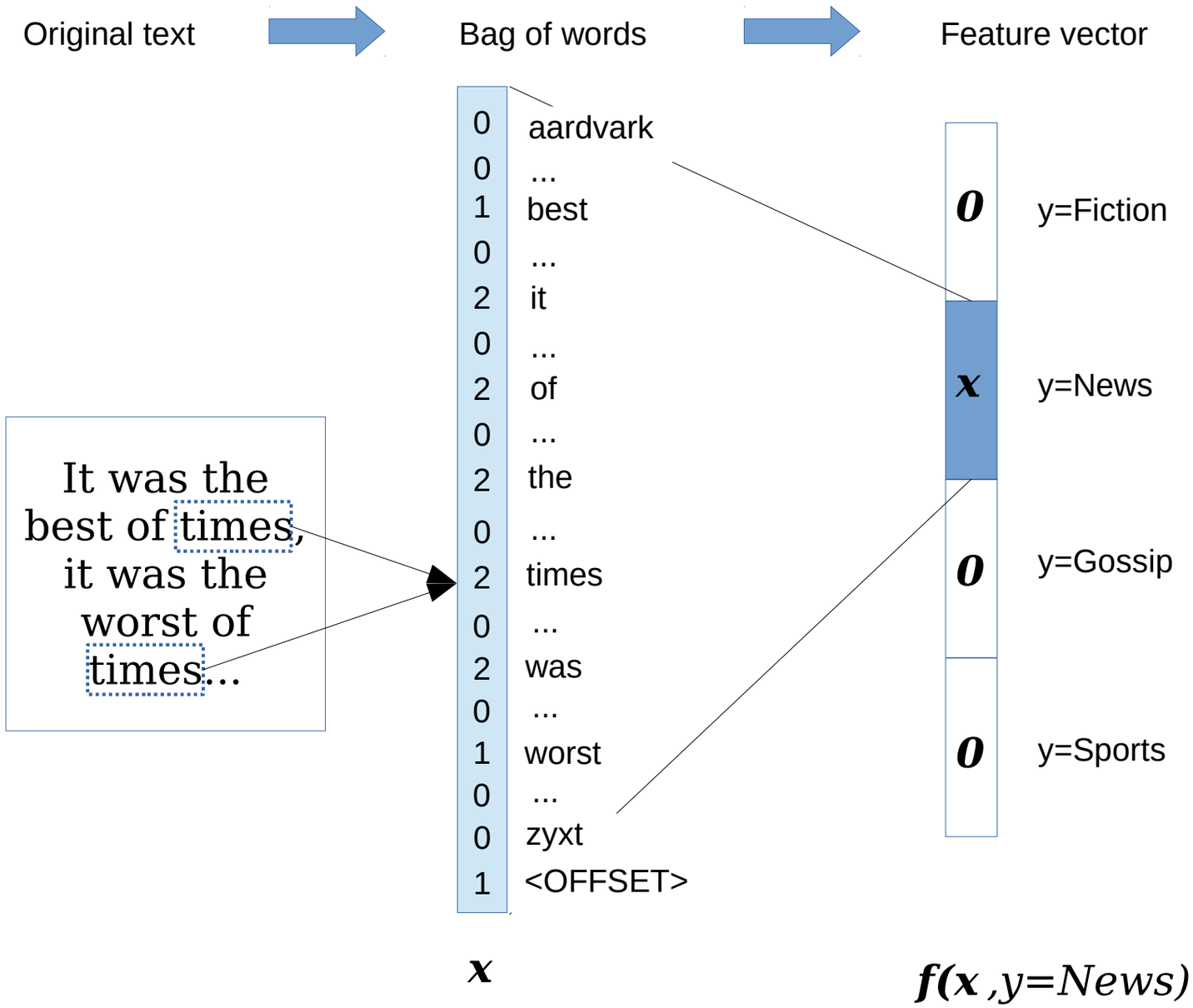
David Smith

Classification as Function Learning

Classification vs. LMs

- Language models exhibit analogies to human abilities—we'll return to LMs shortly
- Naive Bayes trains simple LMs for classification
- Now, a separate tradition of classification as **function learning** (Cf. Breiman, 2001)

Bag-of-Words Vectors



Bag-of-Words Vectors

- Common in vector-space information retrieval (Salton et al., 1960s) and good-old-fashioned expert systems
- What values to assign to elements (here, words)?
  - Introspection: "Hmm, *Moby Dick* is about whales… We'll make *whale* and *harpoon* positive and *lion* negative…" (Cf. sentiment dictionaries)
  - Try simple functions and find one that works: binary, counts, tf-idf (log or sqrt scaling?), …

MOBY DICK IS ABOUT WHALES, OR WHY SHOULD WE COUNT WORDS?

Why are we interested in counting words? The immediate payoff is not always clear. Many of us are familiar with what I like to call the Moby Dick is About Whales model of quantitative work, wherein we generate some kind of word-frequency chart and the most dominant words are terms that are so central to the overall story being presented.

In the case made by Moby Dick is About Whales we get words like WHALE, BOAT, CAPTAIN, SEA presented as hugely important terms. Great! There is no doubt that these terms are important to *Moby Dick*. However, and this is crucial: there is nothing terribly groundbreaking about discovering these words are central to the world of *Moby Dick*. In fact, it is nothing we couldn't have discovered if we sat down and read the book ourselves. (Another example of this phenomenon is 'Shakespeare's plays are about kings and queens', lest it sound like I am picking on the 19c Americanists.)

Words, Features, Weights

- Important to keep separate in our minds
  - *sufficient statistics* of a document we want to classify;
  - *input values* to the classifier;
  - *linear importance weights* of each component of the representation

$$\Psi(\mathbf{x}, y) = \theta \cdot \mathbf{f}(\mathbf{x}, y) = \sum_j \theta_j f_j(\mathbf{x}, y)$$

Weights for a Multiclass Problem

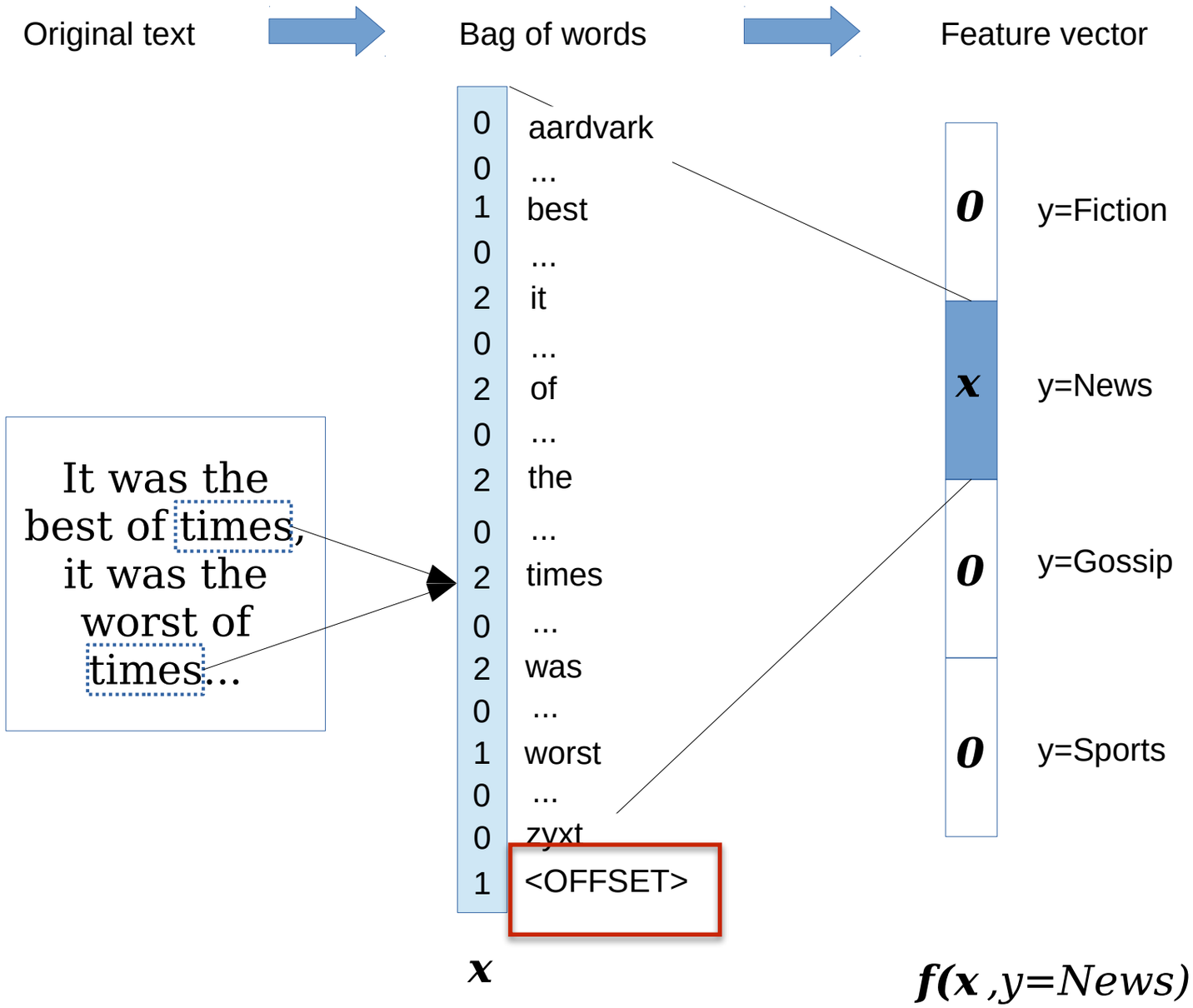
$$\theta_{(E,bicycle)} = 1 \qquad \theta_{(S,bicycle)} = 0$$
$$\theta_{(E,bicicleta)} = 0 \qquad \theta_{(S,bicicleta)} = 1$$
$$\theta_{(E,con)} = 1 \qquad \theta_{(S,con)} = 1$$
$$\theta_{(E,ordinateur)} = 0 \qquad \theta_{(S,ordinateur)} = 0.$$

Sparse Representations

- Most word features will be 0 for most documents
- Manipulating lots of $|V|$-length vectors is inefficient
- Compare inverted indices in IR

```
def compute_score(x, y, weights):
    total = 0
    for feature, count in feature_function(x,y).items():
        total += weights[feature] + count
    return total
```

Bag-of-Words Vectors



Naive Bayes is a Linear Classifier

$$\hat{y} = \arg\max_y \log p(x, y; \mu, \phi) = \arg\max_y \log p(x \mid y; \phi) + \log p(y; \mu)$$

$$\log p(x \mid y; \phi) + \log p(y; \mu) = \log \prod_{i=1}^{|D|} \phi_{y,w_i} + \log \mu_y$$

$$= \sum_{i=1}^{|D|} \log \phi_{y,w_i} + \log \mu_y$$

$$= \theta \cdot \mathbf{f}(x, y)$$

$$\text{where } \theta = [\theta^{(1)}; \theta^{(2)}; \ldots; \theta^{(K)}]$$

$$\theta^{(y)} = [\log \phi_{y,1}; \log \phi_{y,2}; \ldots; \log \phi_{y,V}; \log \mu_y]$$

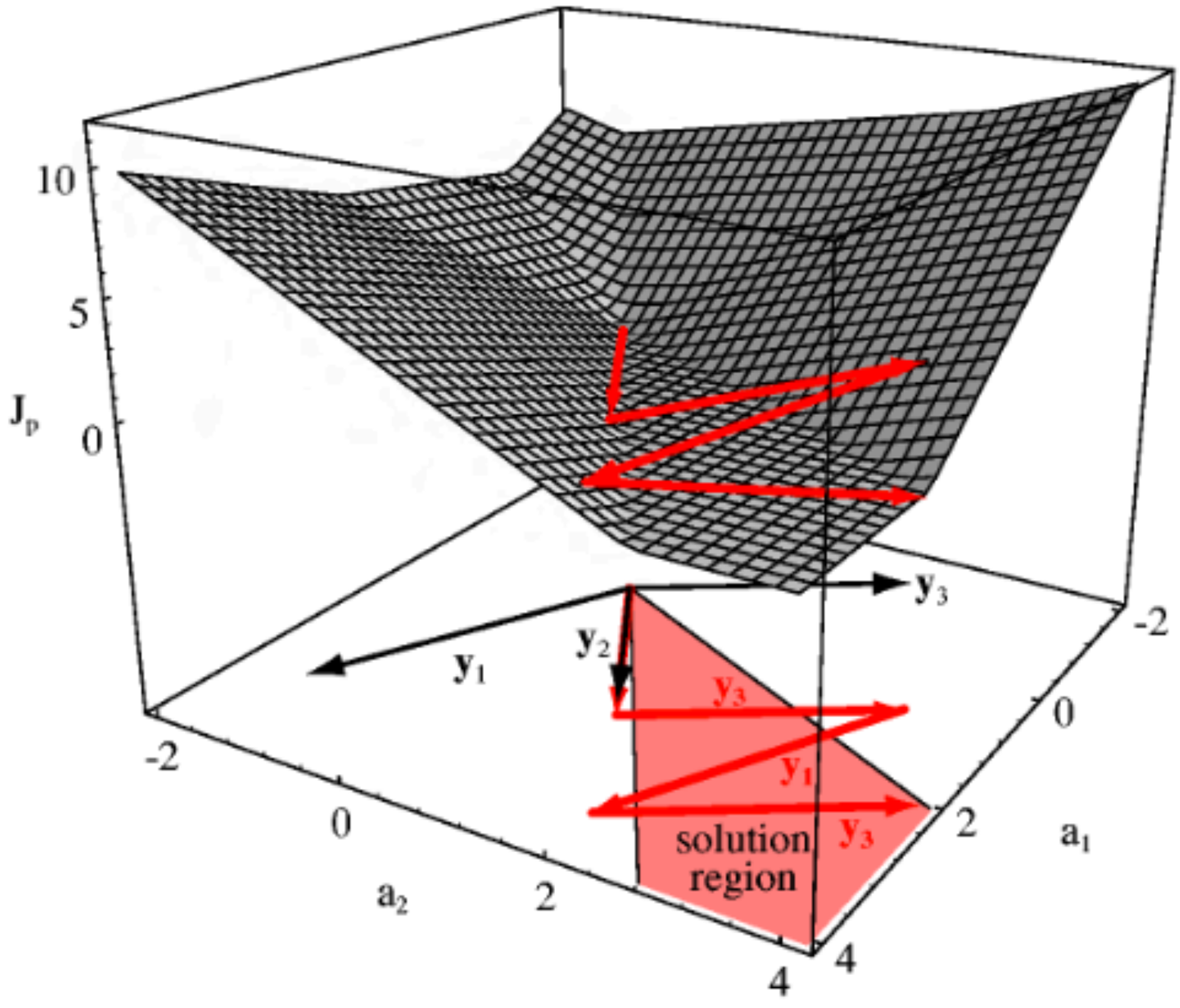
Perceptrons

Generative vs. Discriminative

- Naive Bayes learning maximizes the likelihood of the data
- But data are constant during classification!
- Instead, consider a simple model of concept learning by an idealized neuron that updates weights only when it makes a mistake: the **perceptron** (McCullough & Pitts, 1943)

Algorithm 3 Perceptron learning algorithm

procedure PERCEPTRON($\boldsymbol{x}^{(1:N)}, y^{(1:N)}$)
    $t \leftarrow 0$
    $\boldsymbol{\theta}^{(0)} \leftarrow \mathbf{0}$
    repeat
        $t \leftarrow t + 1$
        Select an instance i
        $\hat{y} \leftarrow \text{argmax}_y \, \boldsymbol{\theta}^{(t-1)} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y)$
        if $\hat{y} \neq y^{(i)}$ **then**
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} + \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \boldsymbol{f}(\boldsymbol{x}^{(i)}, \hat{y})$
        else
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}$
    until tired
    return $\boldsymbol{\theta}^{(t)}$


10
5
0
J_p
y_3
y_2
y_1
-2
0
2
4
a_2
-2
0
2
4
a_1
y_3
y_1
y_3
solution
region

Algorithm 4 Averaged perceptron learning algorithm

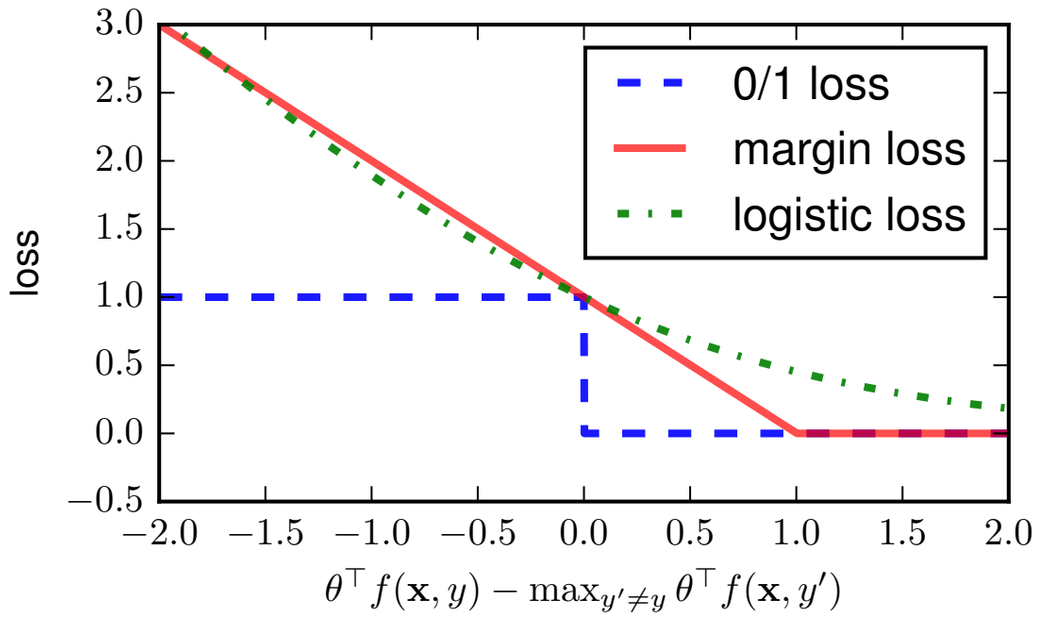
procedure AVG-PERCEPTRON($\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}$)
    $t \leftarrow 0$
    $\boldsymbol{\theta}^{(0)} \leftarrow 0$
    repeat
        $t \leftarrow t + 1$
        Select an instance i
        $\hat{y} \leftarrow \text{argmax}_y \, \boldsymbol{\theta}^{(t-1)} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y)$
        if $\hat{y} \neq y^{(i)}$ **then**
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} + \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - \boldsymbol{f}(\boldsymbol{x}^{(i)}, \hat{y})$
        else
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)}$
        $\boldsymbol{m} \leftarrow \boldsymbol{m} + \boldsymbol{\theta}^{(t)}$
    until tired
    $\overline{\boldsymbol{\theta}} \leftarrow \frac{1}{t}\boldsymbol{m}$
    return $\overline{\boldsymbol{\theta}}$

Loss Functions

$$\ell_{\text{NB}}(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}) = -\log \text{p}(\boldsymbol{x}^{(i)}, y^{(i)}; \boldsymbol{\theta})$$

$$\ell_{\text{0-1}}(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}) = \begin{cases} 0, & y^{(i)} = \text{argmax}_y \, \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y) \\ 1, & \text{otherwise} \end{cases}$$

$$\ell_{\text{PERCEPTRON}}(\boldsymbol{\theta}; \boldsymbol{x}^{(i)}, y^{(i)}) = \max_{y \in \mathcal{Y}} \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y) - \boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)})$$



Perceptron vs. NB

- Both are convex, but only NB solvable in closed form
- ℓ_{NB} can suffer **infinite** loss on a single example (Why?)
- $\ell_{\text{PERCEPTRON}}$ treats all correct answers equally (compare large margin methods)

Logistic Regression

From Classifier to Probability

From Classifier to Probability

Scoring function $\Psi(x, y) = \theta \cdot f(x, y)$

From Classifier to Probability

Scoring function $\Psi(x, y) = \theta \cdot f(x, y)$

Nonnegative $u(x, y) = \exp(\theta \cdot f(x, y))$

From Classifier to Probability

Scoring function $\Psi(x, y) = \theta \cdot f(x, y)$

Nonnegative $u(x, y) = \exp(\theta \cdot f(x, y))$

Joint probability $p(x, y; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{x' \in \mathcal{X}, y' \in \mathcal{Y}} \exp(\theta \cdot f(x', y'))}$

From Classifier to Probability

Scoring function $\Psi(x, y) = \theta \cdot f(x, y)$

Nonnegative $u(x, y) = \exp(\theta \cdot f(x, y))$

Joint probability $p(x, y; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{x' \in \mathcal{X}, y' \in \mathcal{Y}} \exp(\theta \cdot f(x', y'))}$

Conditional prob. $p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x, y'))}$

Logistic Regression

$$p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x, y'))}$$

Logistic Regression

$$p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x, y'))}$$

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x^{(i)}, y'))$$

Logistic Regression

$$p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x, y'))}$$

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x^{(i)}, y'))$$

$$\ell_{\text{Perceptron}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \max_{y' \in \mathcal{Y}} \theta \cdot f(x^{(i)}, y')$$

Priors/Regularization

In the original LogReg loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x^{(i)}, y'))$$

Priors/Regularization

In the original LogReg loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x^{(i)}, y'))$$

What if $\theta_{(j,y^{(i)})} \to \infty$?

Priors/Regularization

In the original LogReg loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x^{(i)}, y'))$$

What if $\theta_{(j,y^{(i)})} \to \infty$?

Do we think, a priori, that weights should be infinite? No!

Priors/Regularization

In the original LogReg loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) = -\theta \cdot f(x^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x^{(i)}, y'))$$

What if $\theta_{(j,y^{(i)})} \to \infty$?

Do we think, a priori, that weights should be infinite? No!

$$p(y \mid x) = p(y \mid x; \theta) \cdot p(\theta)$$

Priors/Regularization

Maximum a Posteriori (MAP) loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) + ???$$

Priors/Regularization

Maximum a Posteriori (MAP) loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) + ???$$

Gaussian/L2 $\log N(\theta; 0, \sigma^2) = -\frac{1}{2\sigma^2}\|\theta\|_2^2 = -\frac{1}{2\sigma^2}\sum_j \theta_j^2$

Priors/Regularization

Maximum a Posteriori (MAP) loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) + ???$$

Gaussian/L2 $$\log N(\theta; 0, \sigma^2) = -\frac{1}{2\sigma^2}\|\theta\|_2^2 = -\frac{1}{2\sigma^2}\sum_j \theta_j^2$$

Laplace/L1 $$\log L(\theta; 0, b) = -\frac{1}{b}\|\theta\|_1 = -\frac{1}{b}\sum_j |\theta_j|$$

Priors/Regularization

Maximum a Posteriori (MAP) loss function

$$\ell_{\text{LogReg}}(\theta; x^{(i)}, y^{(i)}) + ???$$

Gaussian/L2 $$\log N(\theta; 0, \sigma^2) = -\frac{1}{2\sigma^2}\|\theta\|_2^2 = -\frac{1}{2\sigma^2}\sum_j \theta_j^2$$

Laplace/L1 $$\log L(\theta; 0, b) = -\frac{1}{b}\|\theta\|_1 = -\frac{1}{b}\sum_j |\theta_j|$$

Generic variance term/Lagrange multiplier λ

LogReg Gradients

$$\ell_{\textsc{LogReg}} = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp\left(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\right)$$

LogReg Gradients

$$\ell_{\textsc{LogReg}} = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big)$$

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \frac{1}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

LogReg Gradients

$$\ell_{\textsc{LogReg}} = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big)$$

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \frac{1}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \frac{\exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\big)}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

LogReg Gradients

$$\ell_{\textsc{LogReg}} = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big)$$

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \frac{1}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \frac{\exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\big)}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \mathrm{p}(y' \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

LogReg Gradients

$$\ell_{\textsc{LogReg}} = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big)$$

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \frac{1}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \frac{\exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\big)}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \mathrm{p}(y' \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + E_{Y|X}[\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)]$$

LogReg Gradients

$$\ell_{\textsc{LogReg}} = -\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \log \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big)$$

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \frac{1}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \sum_{y' \in \mathcal{Y}} \exp\Big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\Big) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \frac{\exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')\big)}{\sum_{y'' \in \mathcal{Y}} \exp\big(\boldsymbol{\theta} \cdot \boldsymbol{f}(\boldsymbol{x}^{(i)}, y'')\big)} \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + \sum_{y' \in \mathcal{Y}} \mathrm{p}(y' \mid \boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \times \boldsymbol{f}(\boldsymbol{x}^{(i)}, y')$$

$$= -\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) + E_{Y|X}[\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)]$$

all data + L2 prior

$$\nabla_{\boldsymbol{\theta}} L_{\textsc{LogReg}} = \lambda\boldsymbol{\theta} - \sum_{i=1}^{N} \Big(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}) - E_{y|\boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}^{(i)}, y)]\Big)$$

Gradient-Based Optimization

Batch

Gradient descent $\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta^{(t)} \nabla_\theta L$

for **learning rate** η

See also conjugate gradient, quasi-Newton methods

Online

Stochastic gradient descent $\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta^{(t)} E[\nabla_\theta L]$

Single-example Update

perceptron

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \eta^{(t)}[-f(x^{(i)}, y^{(i)}) + \max_{\hat{y}} f(x^{(i)}, \hat{y})]$$

$$E_{Y|X} f(x^{(i)}, y)]$$

logistic regression

Gradient-Based Optimization

Algorithm 5 Generalized gradient descent. The function BATCHER partitions the training set into B batches such that each instance appears in exactly one batch. In gradient descent, $B = 1$; in stochastic gradient descent, $B = N$; in minibatch stochastic gradient descent, $1 < B < N$.

procedure GRADIENT-DESCENT($\boldsymbol{x}^{(1:N)}, \boldsymbol{y}^{(1:N)}, L, \eta^{(1...\infty)}$, BATCHER, T_{max})
    $\boldsymbol{\theta} \leftarrow \mathbf{0}$
    $t \leftarrow 0$
    repeat
        $(\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(B)}) \leftarrow$ BATCHER(N)
        for $n \in \{1, 2, \ldots, B\}$ **do**
            $t \leftarrow t + 1$
            $\boldsymbol{\theta}^{(t)} \leftarrow \boldsymbol{\theta}^{(t-1)} - \eta^{(t)} \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}^{(t-1)}; \boldsymbol{x}^{(b_1^{(n)}, b_2^{(n)}, \ldots)}, \boldsymbol{y}^{(b_1^{(n)}, b_2^{(n)}, \ldots)})$
            if Converged$(\boldsymbol{\theta}^{(1,2,\ldots,t)})$ **then**
                return $\boldsymbol{\theta}^{(t)}$
    until $t \geq T_{\text{max}}$
    return $\boldsymbol{\theta}^{(t)}$

Reparameterizing LogReg

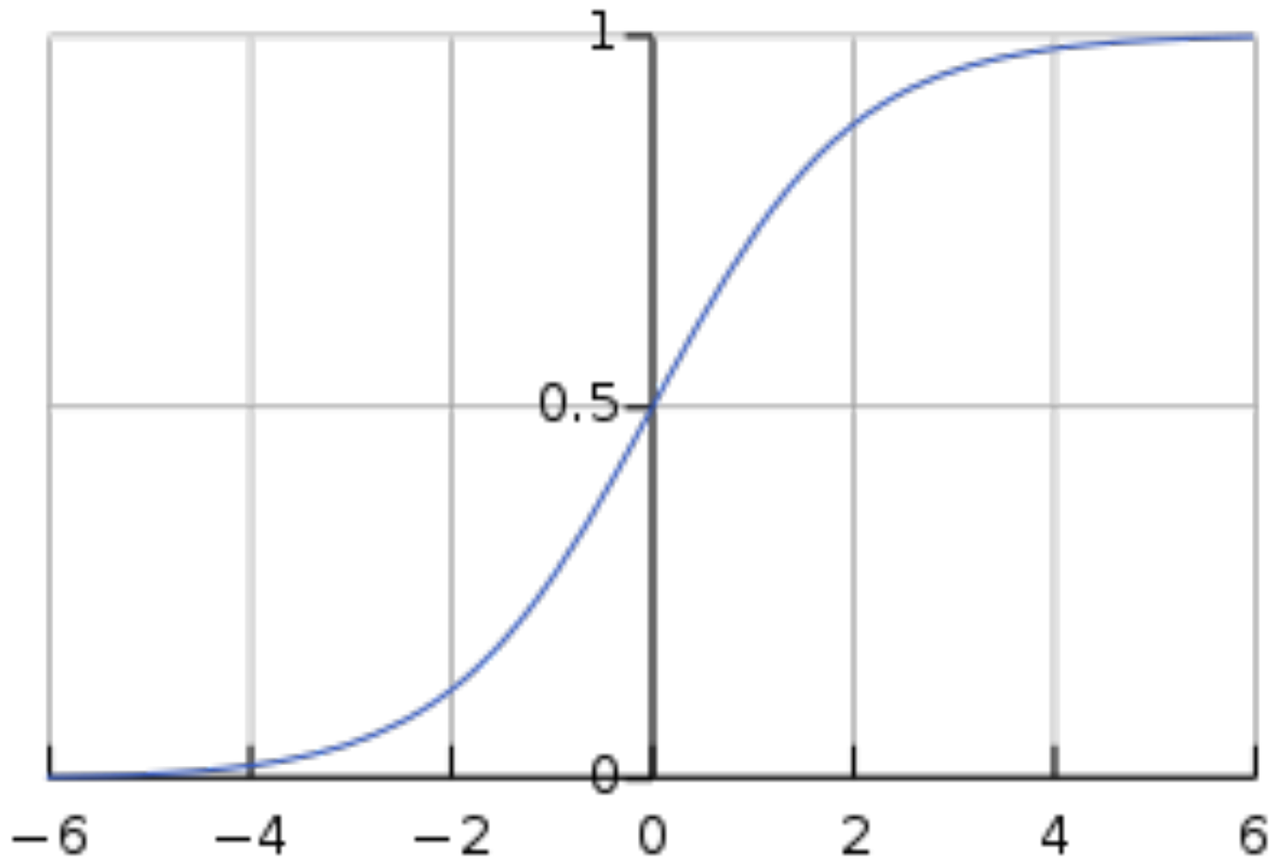
$$p(y \mid x; \theta) = \frac{\exp(\theta \cdot f(x, y))}{\sum_{y' \in \mathcal{Y}} \exp(\theta \cdot f(x, y'))}$$

$$p(\texttt{pos} \mid x; \theta) = \frac{\exp(\theta \cdot f(x, \texttt{pos}))}{\exp(\theta \cdot f(x, \texttt{pos})) + \exp(\theta \cdot f(x, \texttt{neg}))}$$

$$p(\texttt{pos} \mid x; \theta) = \frac{1}{1 + \frac{\exp(\theta \cdot f(x, \texttt{neg}))}{\exp(\theta \cdot f(x, \texttt{pos}))}}$$

$$p(\texttt{pos} \mid x; \theta) = \frac{1}{1 + \exp(-\theta \cdot [f(x, \texttt{pos}) - f(x, \texttt{neg})])}$$

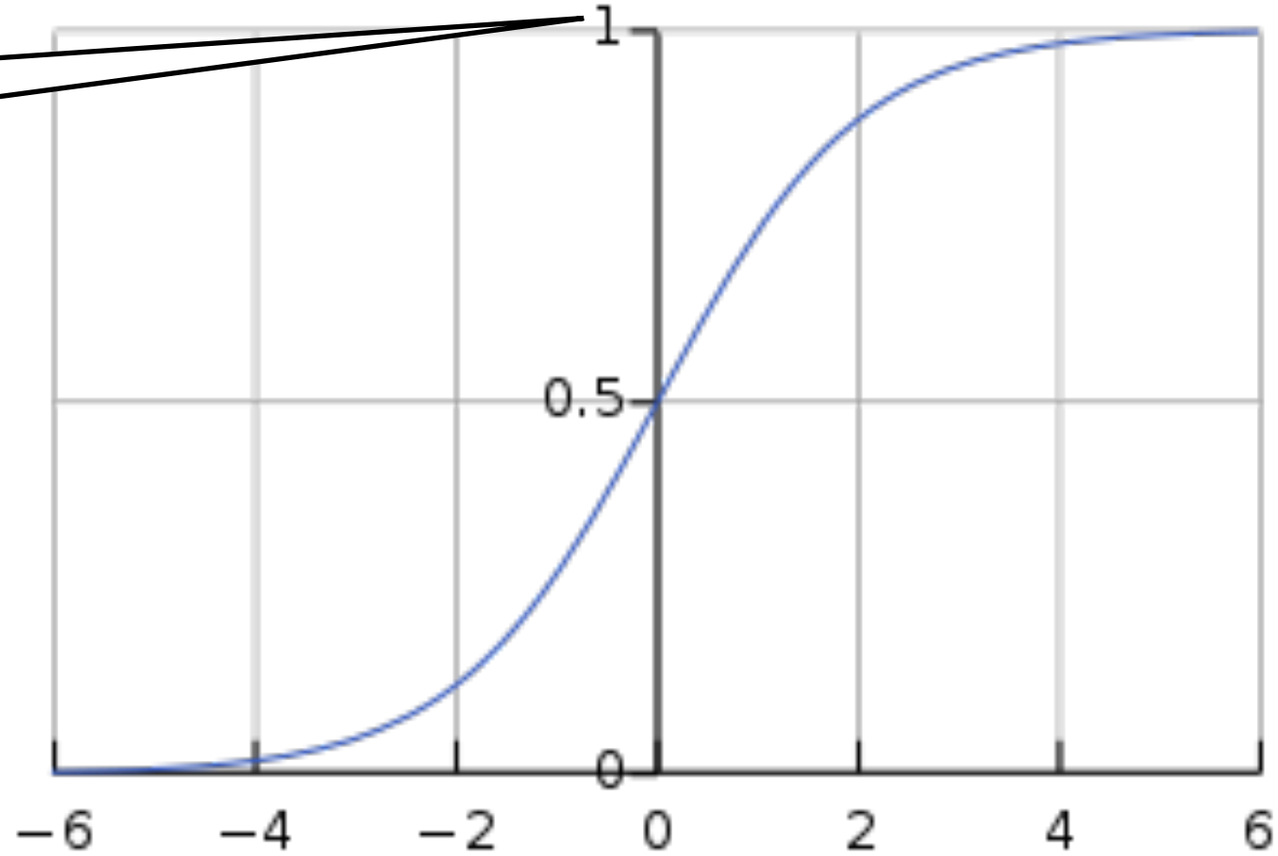
Logistic Sigmoid



$$\sigma(x) = \frac{1}{1 + e^{-\theta x + b}}$$

Logistic Sigmoid

Squashing function:
$[-\infty,\infty] \rightarrow [0,1]$

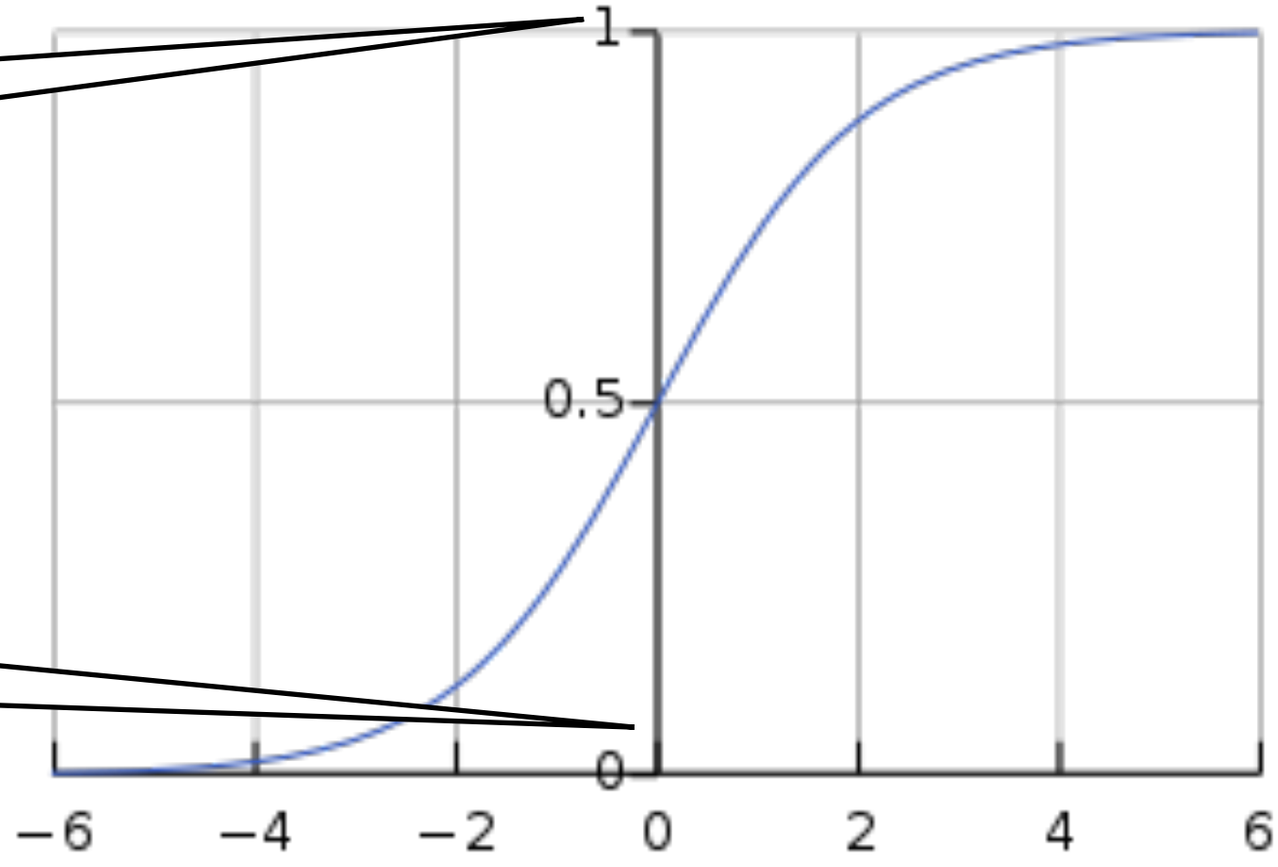


$$\sigma(x) = \frac{1}{1 + e^{-\theta x + b}}$$

Logistic Sigmoid

Squashing function:
$[-\infty,\infty] \rightarrow [0,1]$

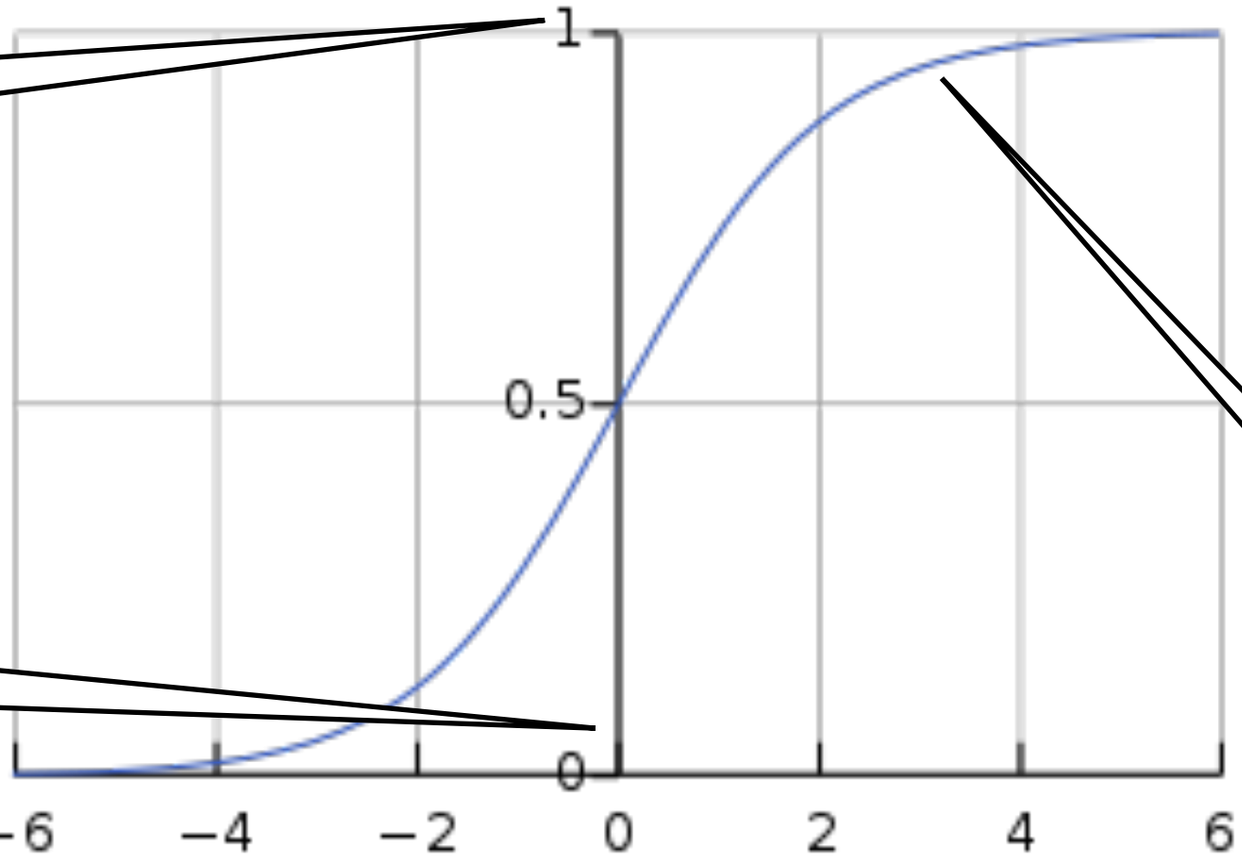
Bias $b = 0$ here



$$\sigma(x) = \frac{1}{1 + e^{-\theta x + b}}$$

Logistic Sigmoid

Squashing function:
$[-\infty,\infty] \rightarrow [0,1]$



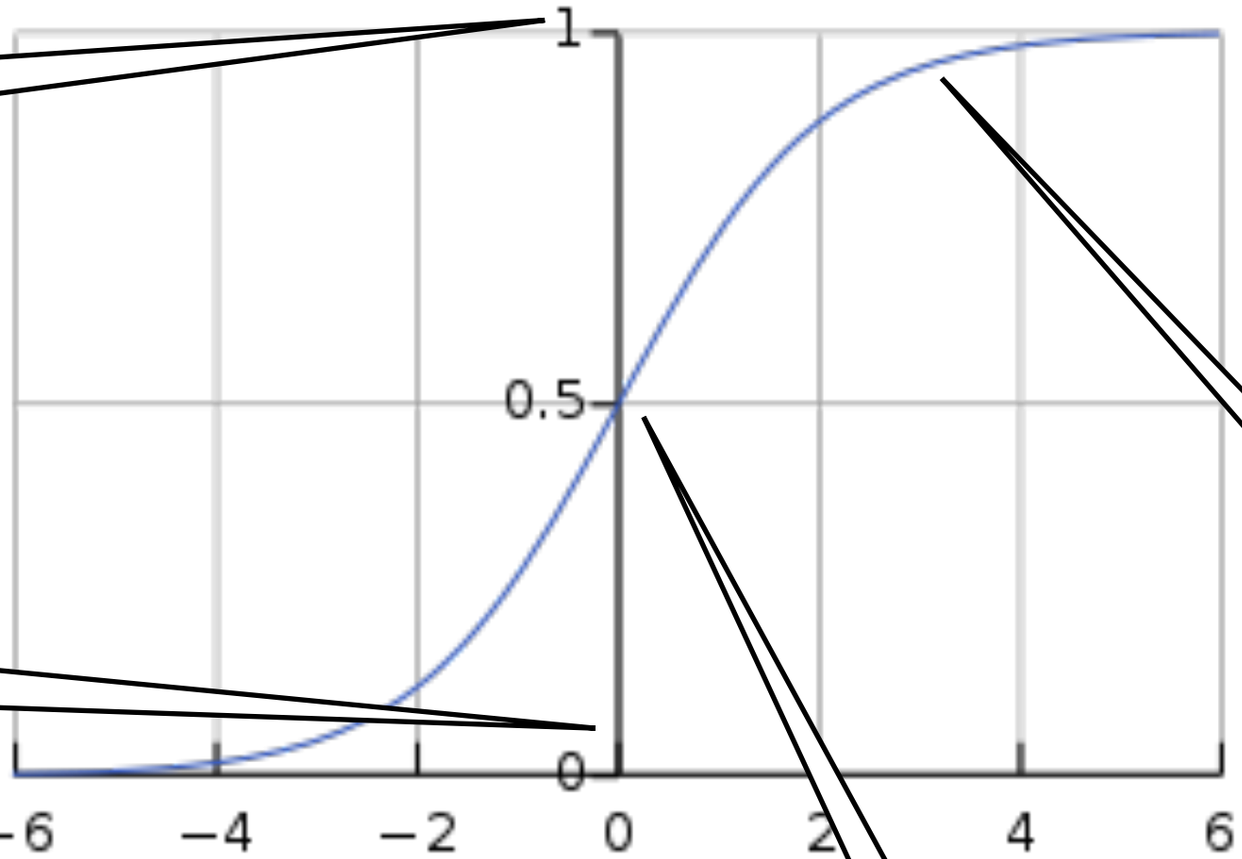
Bias b = 0 here

Offset, scaled tanh

$$\sigma(x) = \frac{1}{1 + e^{-\theta x + b}}$$

Logistic Sigmoid

Squashing function:
$[-\infty,\infty] \rightarrow [0,1]$



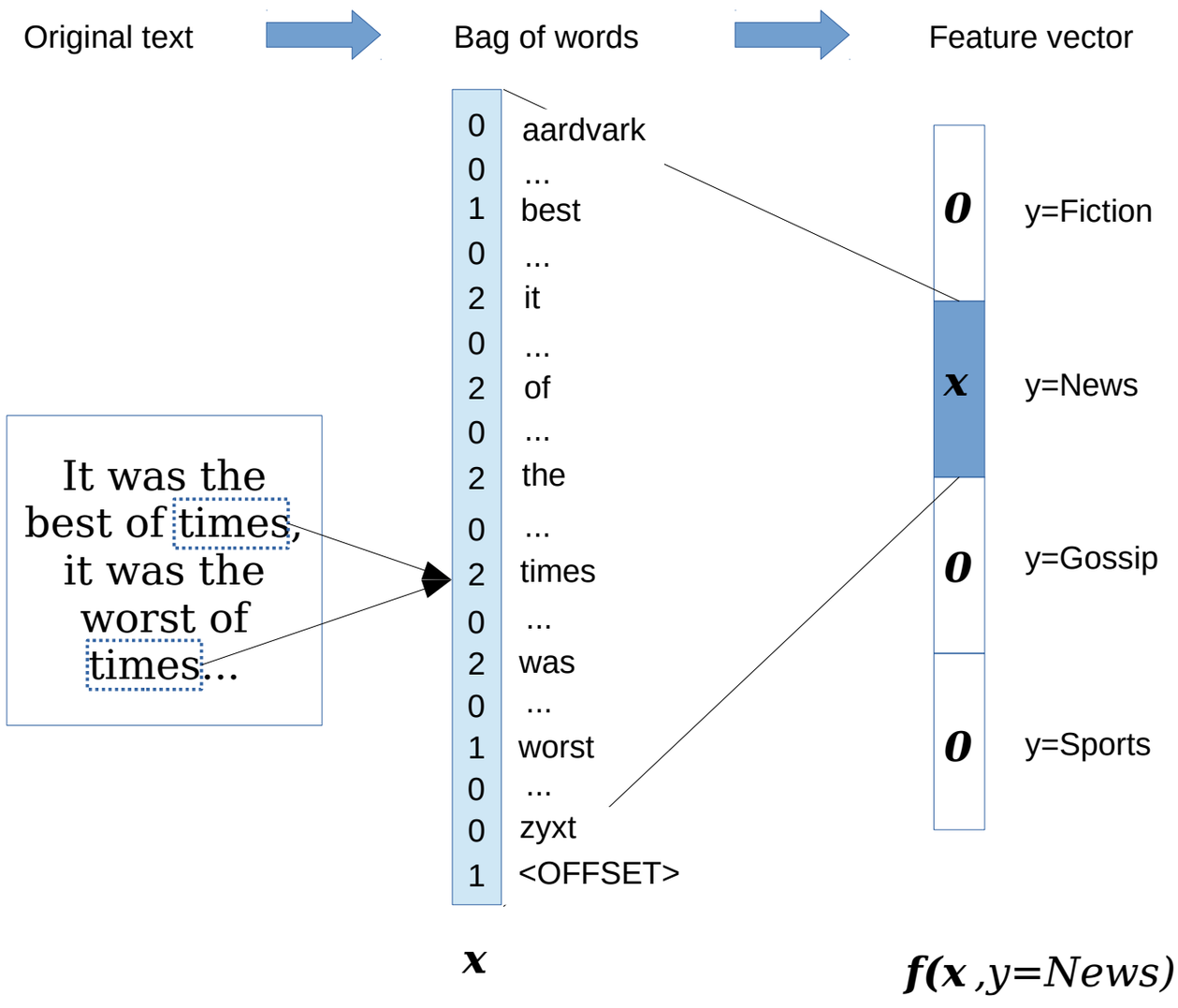
Bias b = 0 here

Offset, scaled tanh

θ is a slope parameter

$$\sigma(x) = \frac{1}{1 + e^{-\theta x + b}}$$

What Should My Inputs Look Like?



One-Hot Encoding

	time	fruit	flies	like	a	an	arrow	banana
1_{time}	1	0	0	0	0	0	0	0
1_{fruit}	0	1	0	0	0	0	0	0
1_{flies}	0	0	1	0	0	0	0	0
1_{like}	0	0	0	1	0	0	0	0
1_{a}	0	0	0	0	1	0	0	0
1_{an}	0	0	0	0	0	1	0	0
1_{arrow}	0	0	0	0	0	0	1	0
1_{banana}	0	0	0	0	0	0	0	1

Summary of Linear Classifiers

- **Naive Bayes**
  - Pro: single-pass, closed-form estimation; probabilistic predictions
  - Con: poor accuracy w/correlated features
- **Perceptron**
  - Pro: online, error-driven learning; typically high accuracy w/ averaging
  - Con: not probabilistic; stopping not well motivated
- **Logistic regression**
  - Pro: error-driven + probabilistic; regularization well-motivated
  - Con: logistic loss saturates/overtrains on correct labels