Logistic regression

- A general classifier (features can be anything, not just word counts)
- Discriminative instead of generative
- Building block for neural networks

Generative vs. discriminative classifiers

- The goal of any classifier is to compute $P(c \mid d)$ for all $c \in C$
- Or just $\arg\max_{c \in C}(c \mid d)$
- A generative model gives us $P(d \mid c)$
- Generative classifier: $P(c \mid d) = \dfrac{P(d \mid c)P(c)}{\sum_{c' \in C} P(d \mid c')P(c')}$

Discriminative classifier

- Discriminative classifiers model $P(c \mid d)$ directly
- For each instance (example) d, we have features $\mathbf{x} = (x_1, \ldots, x_n)$
- This is all we know about the document: we are going to compute $P(c \mid \mathbf{x})$
- Easier: we don't need to know how to compute $P(x_n \mid x_1, \ldots, x_{n-1})$

Building blocks of a discriminative classifier

- Features: $d \rightarrow \mathbf{x}$
- Classification function $P(c \mid \mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$
- Loss function: cross-entropy
- Optimization method: stochastic gradient descent

Classification function

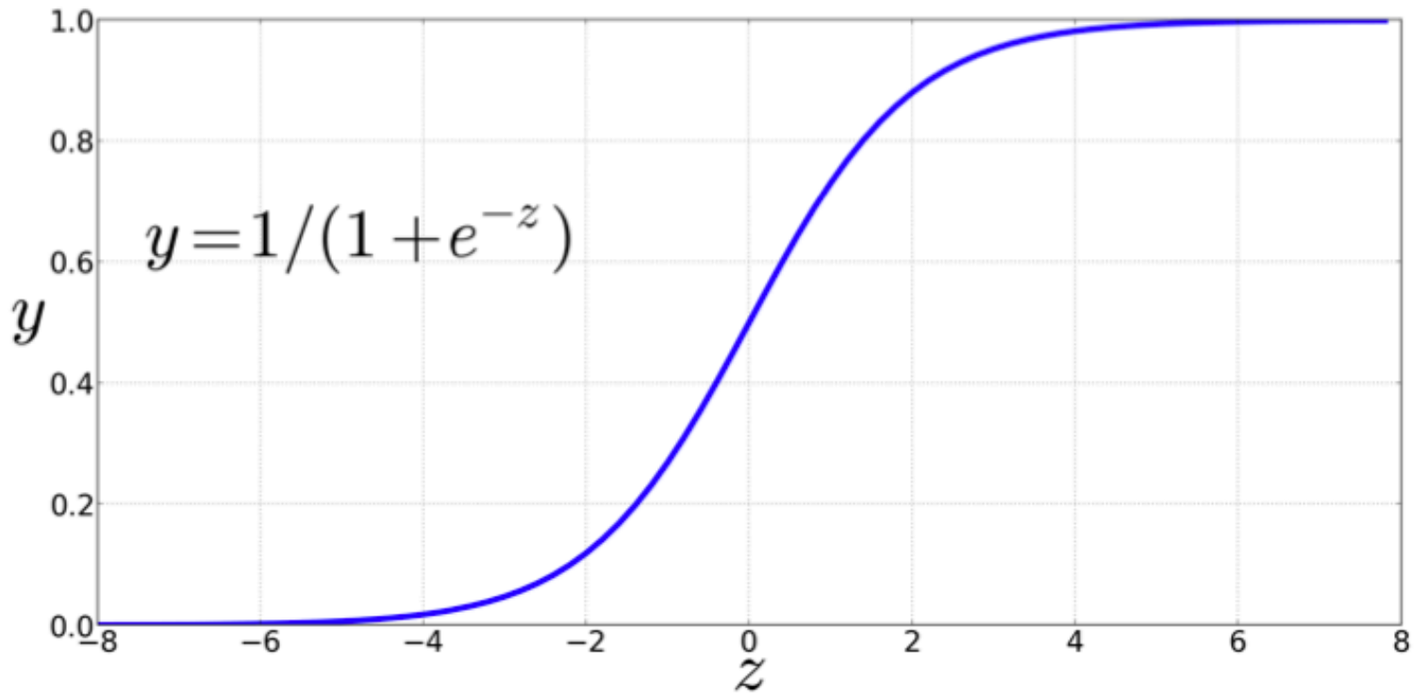
$$z = \mathbf{w} \cdot \mathbf{x} + b = \sum_{j=1}^{n} w_j x_j + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$P(y = 1) = \hat{y}$$

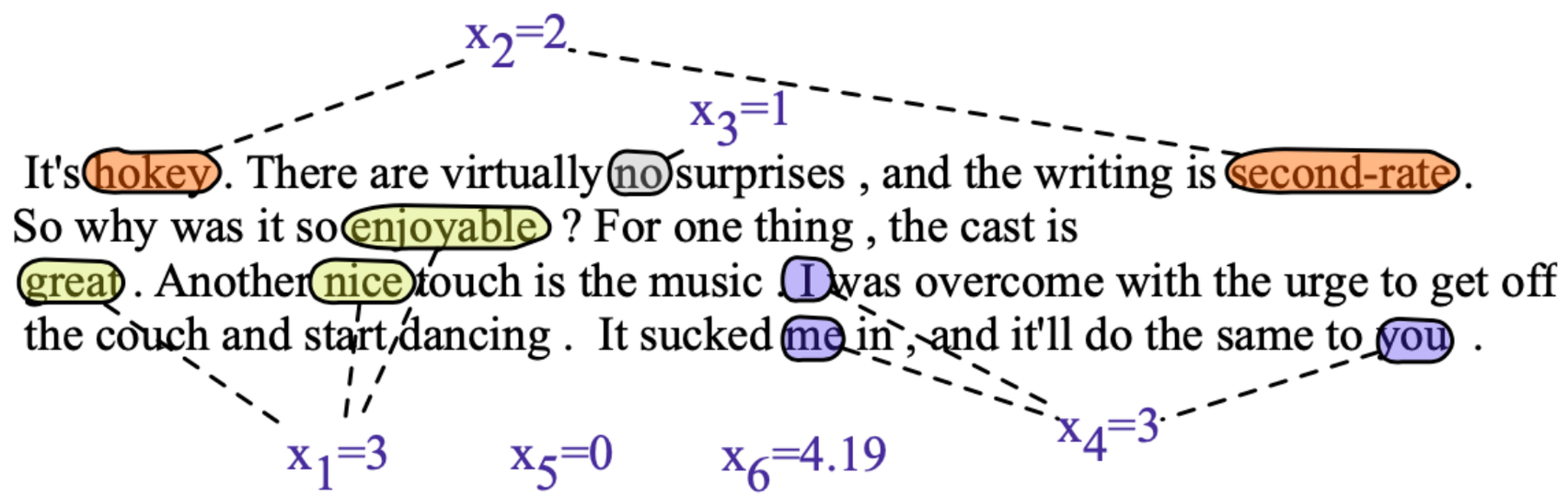
$$P(y = 0) = 1 - \hat{y}$$

$$P(y = 1) > 0.5?$$



Example

Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if ``no'' } \in \text{ doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if ``!'' } \in \text{ doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$



Loss function

$$\mathbf{x}^{(\mathbf{i})}, y^{(i)}, \hat{y}^{(i)}$$

$$L_\theta(\hat{y}^{(i)}, y^{(i)}) \qquad \theta = (\mathbf{w}, b)$$

$$\arg\min_\theta \sum_{i=1}^{k} L_\theta(\hat{y}^{(i)}, y^{(i)})$$

Logistic loss function

$$\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

$$\hat{P}(y = 1 \,|\, x) = \hat{y}$$

$$\hat{P}(y = 0 \,|\, x) = 1 - \hat{y}$$

$$\hat{P}(y \,|\, x) = \hat{y}^{y}(1 - \hat{y})^{1-y}$$

$$\log \hat{P}(y \,|\, x) = y \log(\hat{y}) + (1 - y)\log(1 - \hat{y})$$

$$L(\hat{y}, y) = -\, y \log(\hat{y}) - (1 - y)\log(1 - \hat{y})$$

Gradient descent

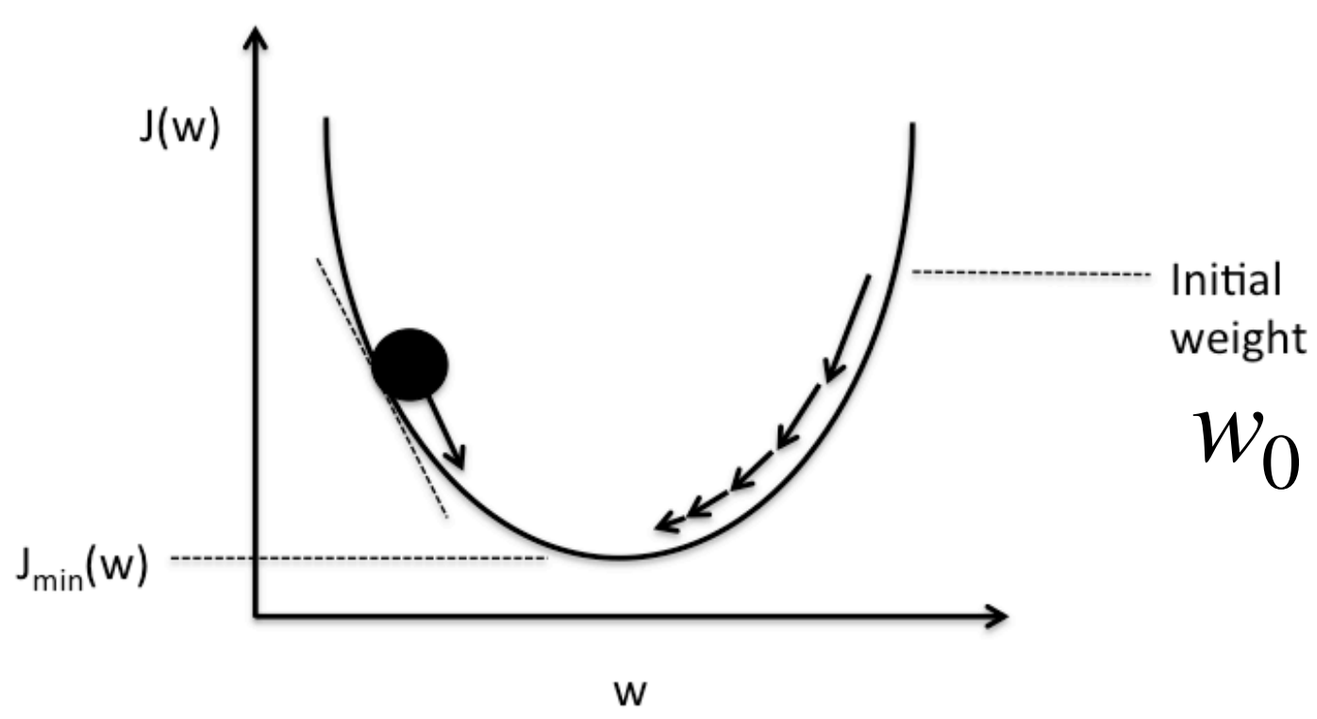
- Goal: find the weights that minimize the loss
  $L(\hat{y}, y) = L(w)$

- The derivative of the loss indicates whether the loss is increasing or decreasing in *w*, and at what rate

The derivative of the loss function

$$w^{t+1} = w^t - \eta L'(w)$$

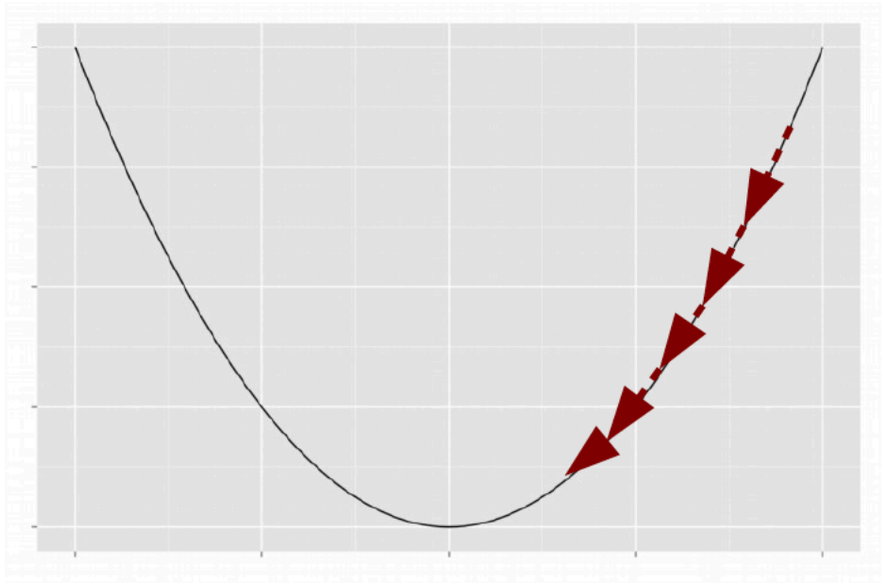
Learning rate



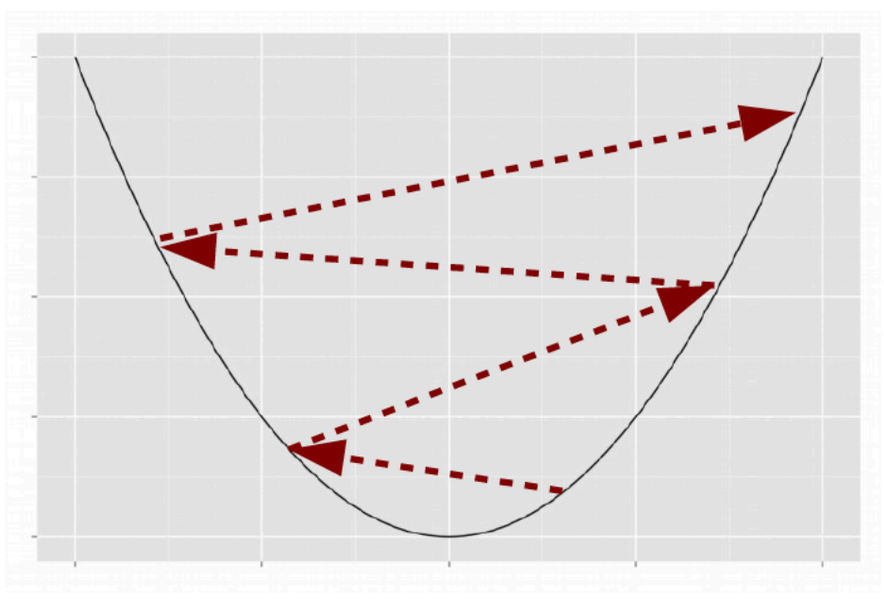
Schematic of gradient descent.

Gradient descent: learning rate

Lower learning rate: slow convergence



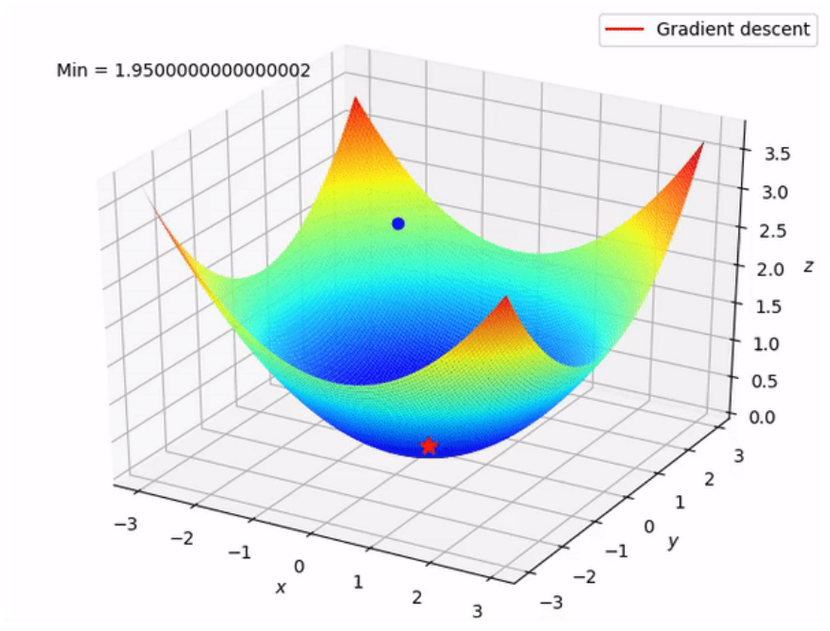
Higher learning rate: may not always converge



Gradient descent

- With multiple variables: move in the direction of the gradient of the loss function

$$\mathbf{w}^{(\mathbf{t}+\mathbf{1})} = \mathbf{w}^{(\mathbf{t})} - \eta \nabla L(\mathbf{w})$$



$$\nabla L(\mathbf{w}) = \left(\frac{\partial L}{\partial w_1}, \ldots, \frac{\partial J}{\partial w_n} \right)$$

Gradient descent for logistic regression

Loss for the instance $(\mathbf{x}^{(i)}, y^{(i)})$:

$$L^{(i)}(\mathbf{w}, b) = -\, y^{(i)} \log(\sigma(\mathbf{w} \cdot \mathbf{x}^{(\mathbf{i})} + b)) - (1 - y^{(i)}) \log(1 - \sigma(\mathbf{w} \cdot \mathbf{x}^{(\mathbf{i})} + b))$$

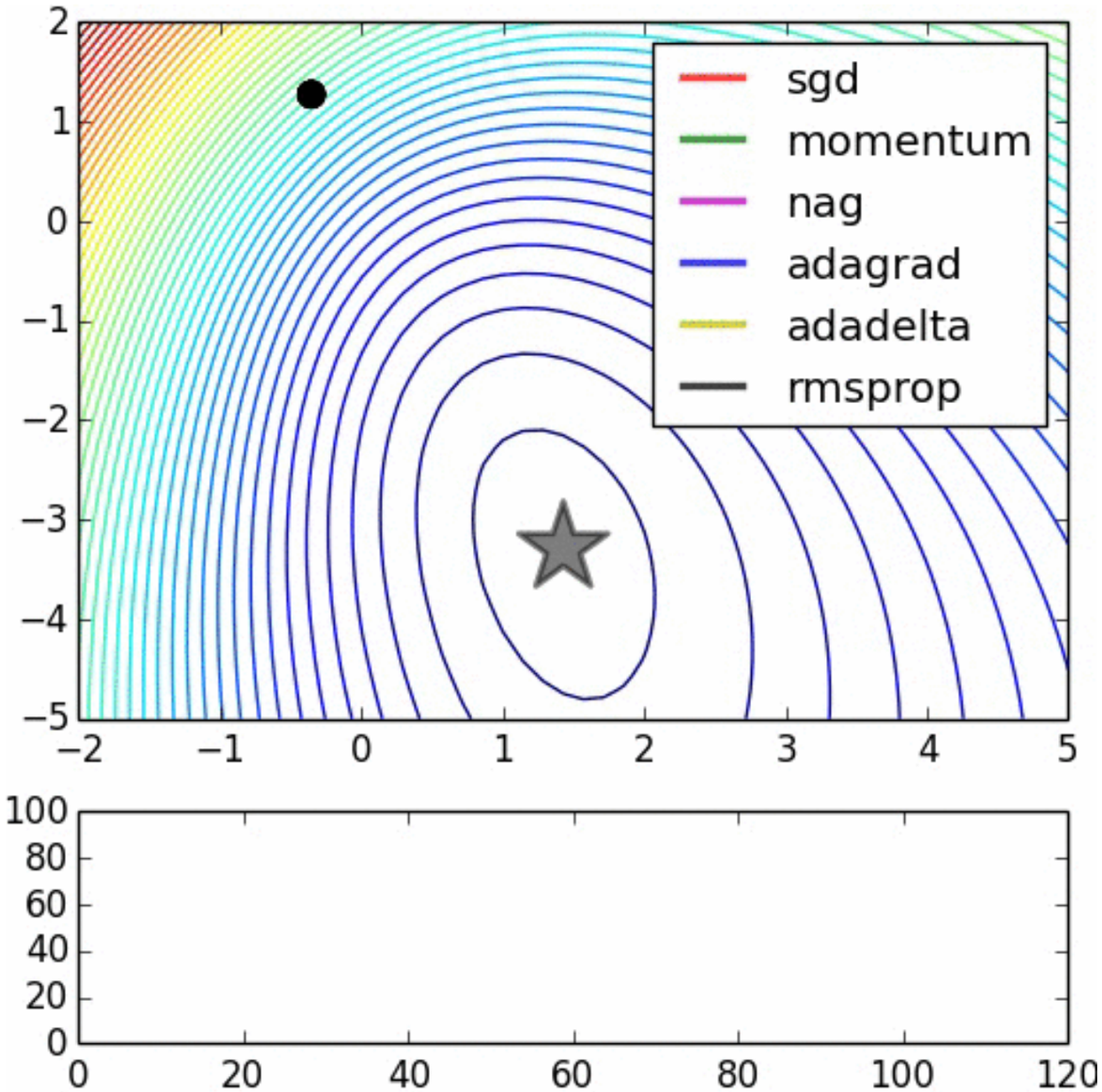
$$\frac{\partial L^{(i)}(\mathbf{w}, b)}{\partial w_j} = (\sigma(\mathbf{w} \cdot \mathbf{x}^{(\mathbf{i})} + b) - y^{(i)}) x_j^{(i)}$$

$$L(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} L^{(i)}(\mathbf{w}, b)$$

Stochastic gradient descent

- The loss is a function of all the training instances
- Stochastic gradient descent (SGD): compute the loss gradient after each example, and move in that direction
- Noisy estimate of the gradient over the full training data: mini-batching can reduce the variance of this estimate

Variations on stochastic gradient descent



(Credit: Alec Radford)

Skip-gram with negative sampling (SGNS)

```
... lemon,  a [tablespoon of apricot jam,      a] pinch ...
               c1         c2    t     c3        c4
```

$$P(+|t,c) = \sigma(\mathbf{t}\cdot\mathbf{c}) \qquad P(-|t,c) = 1-\sigma(\mathbf{t}\cdot\mathbf{c})$$

$$P(+|t,c_1,\ldots,c_k) = \prod_{i=1}^{k} P(+|t,c_i)$$

Skip-gram with negative sampling (SGNS)

```
... lemon,  a [tablespoon of apricot jam,      a] pinch ...
               c1         c2    t     c3       c4
```

$$P(+|t,c) = \sigma(\mathbf{t}\cdot\mathbf{c}) \qquad P(-|t,c) = 1-\sigma(\mathbf{t}\cdot\mathbf{c})$$

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

negative examples -

t	c	t	c
apricot	aardvark	apricot	seven
apricot	my	apricot	forever
apricot	where	apricot	dear
apricot	coaxial	apricot	if

Skip-gram with negative sampling (SGNS)

$$\theta = (\mathbf{t}_{\text{dog}}, \mathbf{c}_{\text{dog}}, \mathbf{t}_{\text{penguin}}, \mathbf{c}_{\text{penguin}}, \ldots)$$

$$F(\theta) = \sum_{(t,c)\in +} \log(P(+|t,c)) + \sum_{(t,c)\in -} \log(P(-|t,c))$$

$$P(+|t,c) = \sigma(\mathbf{t}\cdot\mathbf{c}) \qquad P(-|t,c) = 1 - \sigma(\mathbf{t}\cdot\mathbf{c})$$