
FNLP Lecture 5

Text Classification / Logistic Regression

Ivan Titov



Last time: Naive Bayes

- Given document x and set of categories Y (say, spam/not-spam), we want to assign x to the most probable category \hat{y} .

$$\begin{aligned}\hat{y} &= \operatorname{argmax}_{y \in Y} P(y|x) \\ &= \operatorname{argmax}_{y \in Y} P(x|y)P(y)\end{aligned}$$

- The **naive Bayes assumption**: features are conditionally independent given the class.

$$P(f_1, f_2, \dots, f_n|y) \approx P(f_1|y)P(f_2|y) \dots P(f_n|y)$$

- That is, the prob. of a word occurring depends **only** on the class.

Alternative feature values and feature sets

- Use only **binary** values for f_i : did this word occur in d or not?
- Use only a subset of the vocabulary for F
 - Ignore **stopwords** (function words and others with little content)
 - Choose a small task-relevant set (e.g., using a sentiment lexicon)
- Use more complex features (bigrams, syntactic features, morphological features, ...)

Task-specific features

- And for other tasks, stopwords might be very useful features
 - E.g., People with schizophrenia use more 2nd-person pronouns (?), those with depression use more 1st-person (?).
- Probably better to use too many irrelevant features than not enough relevant ones.

Advantages of Naive Bayes

- Very easy to implement
- Very fast to train and test
- Doesn't require as much training data as some other methods
- Usually works reasonably well

Use as a simple baseline for any classification task.

Problems with Naive Bayes

- Naive Bayes assumption is naive!
- Consider categories TRAVEL, FINANCE, SPORT.
- Are the following features independent given the category?

beach, sun, ski, snow, pitch, palm, football, relax, ocean

Problems with Naive Bayes

- Naive Bayes assumption is naive!
- Consider categories TRAVEL, FINANCE, SPORT.
- Are the following features independent given the category?

beach, sun, ski, snow, pitch, palm, football, relax, ocean

- No! They might be closer if we defined finer-grained categories (beach vacations vs. ski vacations), but we don't usually want to.

Non-independent features

- Features are not usually independent given the class
- Adding multiple feature types (e.g., words and morphemes) often leads to even stronger correlations between features
- Accuracy of classifier can sometimes still be ok, but it will be highly **overconfident** in its decisions.
 - Ex: NB sees 5 features that all point to class 1, treats them as five independent sources of evidence.
 - Like asking 5 friends for an opinion when some got theirs from each other.

A less naive approach

- Although Naive Bayes is a good starting point, often we have enough training data for a better model (and not so much that slower performance is a problem).
- We may be able to get better performance using loads of features and a model that doesn't assume features are conditionally independent.
- Namely, a **multinomial logistic regression** model.

Multinomial Logistic Regression

- Used widely in many different fields, under many different names
- Most commonly, **multinomial logistic regression**
 - *multinomial* if more than two possible classes
 - otherwise (or if lazy) just *logistic regression*
- Also called: max-ent classifier, log-linear model, one-layer neural network, single neuron classifier, etc ...

Naive Bayes vs Logistic Regression

- Like Naive Bayes, Logistic Regression assigns a document x to class \hat{y} , where

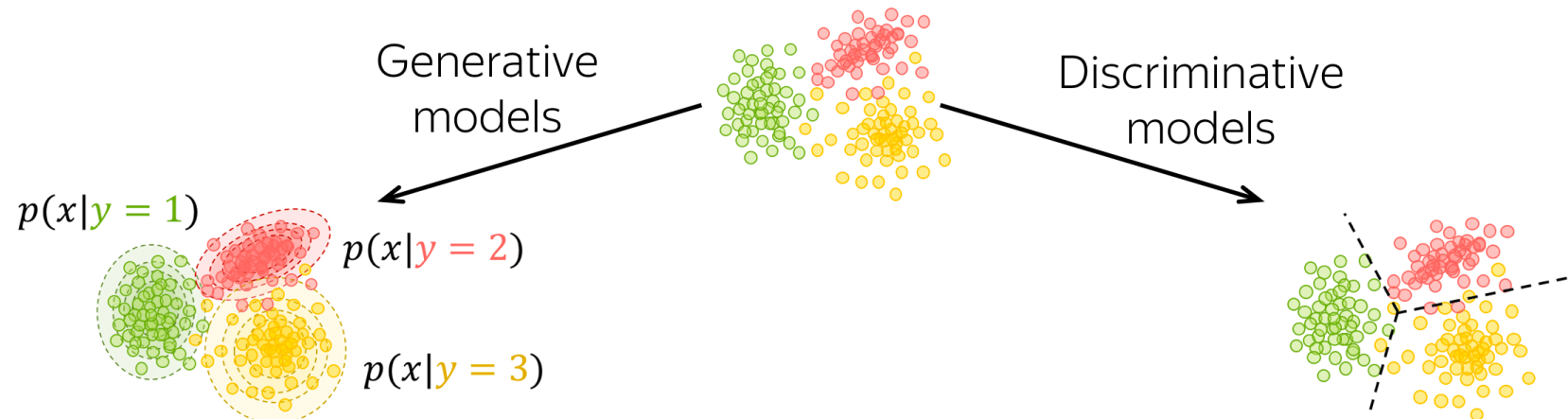
$$\hat{y} = \operatorname{argmax}_{y \in Y} P(y|x)$$

- Unlike Naive Bayes, we do not apply Bayes' Rule. Instead, we model $P(y|x)$ directly.

Logistic Regression is a *discriminative* model

- It is trained to **discriminate** correct vs. incorrect values of y , given input x . That's all it can do.
- Naive Bayes can also **generate** data: sample a class from $P(y)$, then sample words from $P(x|y)$. So, we call it a **generative** model.

Discriminative models more broadly



Learn: data distribution $p(x, y) = p(x|y) \cdot p(y)$

How predict: $y = \arg \max_k P(x, y = k) =$
 $= \arg \max_k P(x|y = k) \cdot P(y = k)$

Learn: boundary between classes $p(y|x)$

How predict: $y = \arg \max_k P(y = k|x)$

Discriminative models even more broadly

- Trained to **discriminate** correct vs. wrong values of y , given input x .
- Need not be probabilistic.
- Examples: various neural networks, decision trees, nearest neighbor methods, support vector machines
- Here, we consider only one method: logistic regression models, which *are* probabilistic.

Example: classify by topic

- Given a web page document, which topic does it belong to?
 - x is the words in the document, plus info about headers and links.
 - y is the unknown class. Assume three possibilities:

$y =$	class
1	TRAVEL
2	SPORT
3	FINANCE

Feature functions

- Like Naive Bayes, Logistic Regression models use **features** we think will be useful for classification.
- For example, we could have a binary corresponding to each token (word) in the vocabulary:

f_1 : contains('ski')

f_2 : contains('sun')

... ..

f_n : contains('antidisestablishmentarianism')

Classification with LR

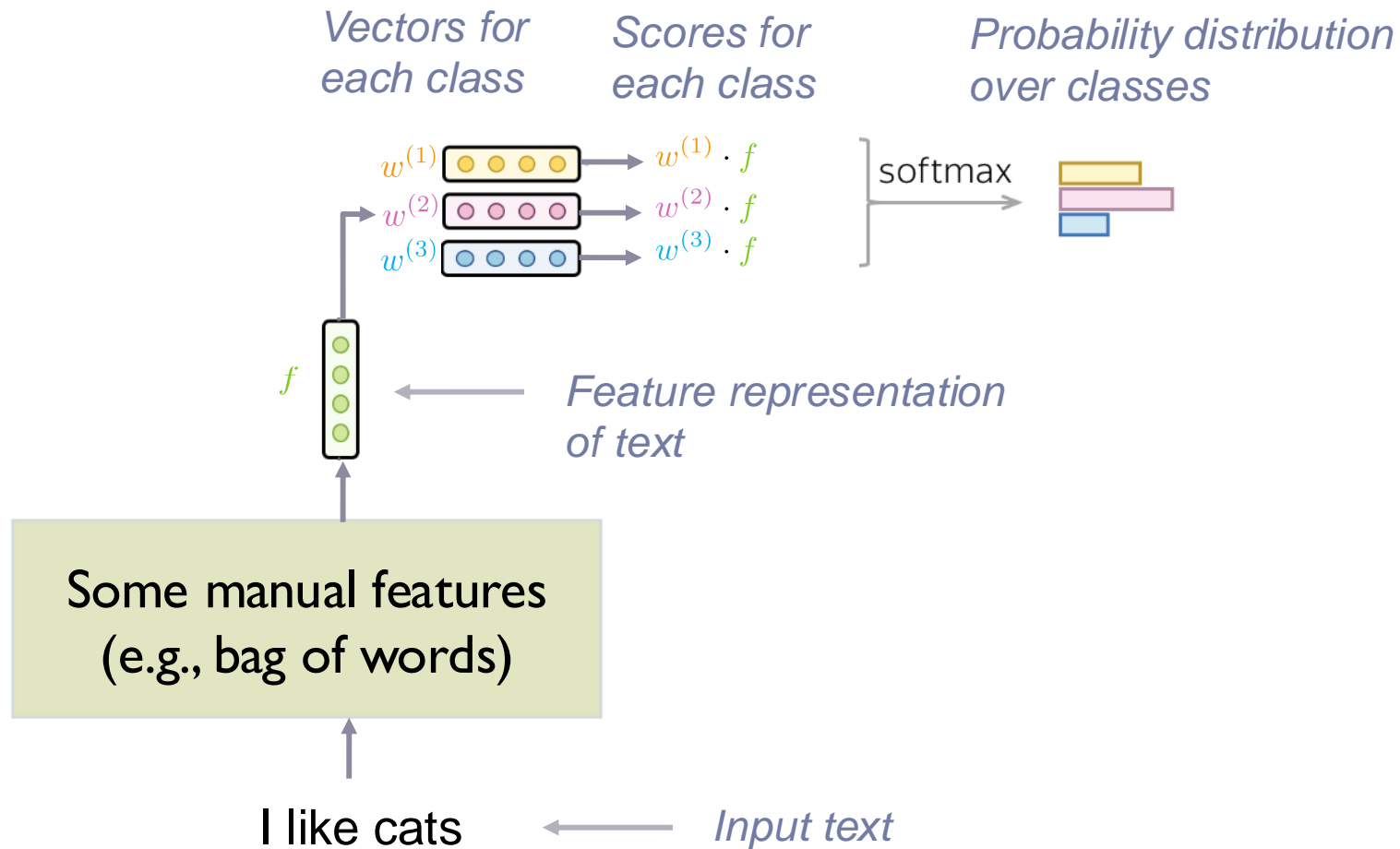
Choose the class that has highest probability according to

$$P(y = k|x) = \frac{1}{Z} \exp \left(\sum_i w_i^{(k)} f_i(x) \right)$$

where the normalization constant $Z = \sum_{k'} \exp(\sum_i w_i^{(k')} f_i(x))$

- Inside brackets is just a dot product: $s^{(k)} = w^{(k)} \cdot f(x)$.
- Z does not depend on k
- So, we will end up choosing class k for which $s^{(k)}$ is highest.
- **Softmax** function: exponentiation of scores s , followed by normalization to turn into a distribution

Schematic view of logistic regression



Classification example

f_1 : `contains('ski')`

$$w_1^{(1)} = 1.2$$

$$w_1^{(2)} = 2.3$$

$$w_1^{(3)} = -0.5$$

f_2 : `link_to('expedia.com')`

$$w_2^{(1)} = 4.6$$

$$w_2^{(2)} = -0.2$$

$$w_2^{(3)} = 0.5$$

f_3 : `num_links`

$$w_3^{(1)} = 0.0$$

$$w_3^{(2)} = 0.2$$

$$w_3^{(3)} = -0.1$$

- f_3 is a **numeric** feature that counts outgoing links.

Classification example

- Suppose our test document contains **ski** and 6 outgoing links.
- We don't know class **y** for this doc, so we try out each possible value.
 - * Travel: $\sum_i w_i^{(1)} f_i(x) = 1.2 + (0.0)(6) = 1.2$.
 - * Sport: $\sum_i w_i^{(2)} f_i(x) = 2.3 + (0.2)(6) = 3.5$.
 - * Finance: $\sum_i w_i^{(3)} f_i(x) = -0.5 + (-0.1)(6) = -1.1$.
- We'd need to do further work to compute the probability of each class, but we know already that **SPORT** will be the most probable.

Feature templates

- In practice, features are usually defined using **templates**
 - `contains(w)`
 - `header_contains(w)`
 - `header_contains(w) & link_in_header`
 - * instantiate with all possible words w
 - * usually filter out features occurring very few times
- NLP tasks often have a few templates, but 1000s or 10000s of features

Training: conditional likelihood

- Examples $x^{(1)} \dots x^{(N)}$ are annotated with labels $y^{(1)} \dots y^{(N)}$
- The **conditional likelihood** (CL) of a model is given by the probability of labels under the model

$$CL = \prod_j P(y^{(j)} | x^{(j)})$$

Training: conditional likelihood

- Examples $x^{(1)} \dots x^{(N)}$ are annotated with labels $y^{(1)} \dots y^{(N)}$
- The **conditional likelihood** (CL) of a model is given by the probability of labels under the model

$$CL = \prod_j P(y^{(j)} | x^{(j)})$$

- In practice, we work with conditional **log**-likelihood:

$$CLL = \sum_j \log P(y^{(j)} | x^{(j)})$$

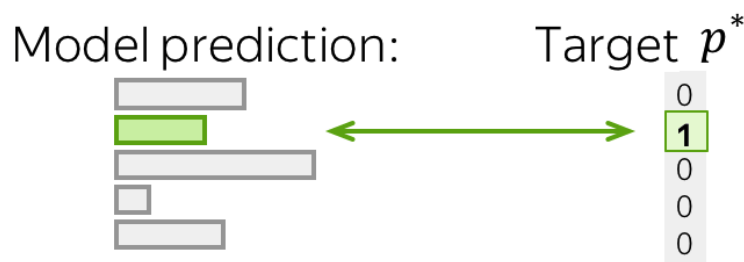
- Choose weights that maximize conditional log-likelihood CLL:

$$\hat{w} = \operatorname{argmax}_w \sum_j \log P(y^{(j)} | x^{(j)})$$

Training the model: cross-entropy

- Think of the target label $y^{(j)}$ as a distribution over classes, $P_*^{(j)} = (0, \dots, 1, \dots, 0)$, where $P_*^{(j)}(y^{(j)}) = 1$ and the rest are 0s
- **cross-entropy loss** for a training set is defined as

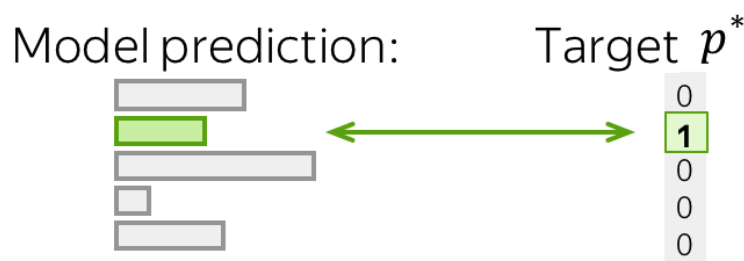
$$loss_{CE} = - \sum_j \sum_{y \in Y} P_*^{(j)}(y) \cdot \log P(y|x^{(j)})$$



Training the model: cross-entropy

- Think of the target label $y^{(j)}$ as a distribution over classes, $P_*^{(j)} = (0, \dots, 1, \dots, 0)$, where $P_*^{(j)}(y^{(j)}) = 1$ and the rest are 0s
- **cross-entropy loss** for a training set is defined as

$$loss_{CE} = - \sum_j \sum_{y \in Y} P_*^{(j)}(y) \cdot \log P(y|x^{(j)})$$



- Since $P_*^{(j)}$ is a **one-hot** vector

$$loss_{CE} = - \sum_j \log P(y^{(j)}|x^{(j)}) = -CLL$$

- So maximizing CLL and minimizing CE loss is equivalent

Regularisation: Weight Decay

Simply minimising the conditional (log-)likelihood may lead to overfitting (decreasing the training loss while increasing the **generalisation loss**).

An inductive bias for our parameters is Occam's razor: our solution should be 'simple'.

Weight decay (or L_2 regularisation) adds the L_2 norm of the parameters to the loss, such that:

$$\hat{w} = \arg \min - \sum_{\mathbf{x} \in \mathcal{D}} \log f_w(\mathbf{x}) + \lambda ||w||_2^2$$

where λ is an importance hyper-parameter. This corresponds to a Bayesian prior $\mathcal{N}(0, \lambda^{-1}I)$.

Training: gradient descent

$w \leftarrow \text{Random}()$

Random initialization (e.g.,
from a Gaussian
distribution)

repeat

$$w \leftarrow w + \eta \cdot \nabla_w \sum_{j=1}^N \log P(y^{(j)} | x^{(j)})$$

until Converged()

Common strategy: finish
when the performance on
the development set stops
improving (or after a fixed
number of iterations)

Learning rate: a scalar
regulating how much you
update on every example

Training: mini-batch gradient descent

$w \leftarrow \text{Random}()$

Choosing a “**batch**”: Indexes of a random subset of examples (e.g., choose 10 random examples)

repeat

$B \leftarrow \text{RandomSubset}([1, \dots, N])$

$$w \leftarrow w + \eta \cdot \nabla_w \sum_{j \in B} \log P(y^{(j)} | x^{(j)})$$

until Converged()

Sum only over examples in the current batch

How does the gradient look like?

$$P(y = k|x) = \frac{1}{Z} \exp \left(\sum_i w_i^{(k)} f_i(x) \right)$$
$$Z = \sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x) \right)$$

Let us consider a single component of the gradient, corresponding to a specific feature l and specific class k

How does the gradient look like?

$$P(y = k|x) = \frac{1}{Z} \exp \left(\sum_i w_i^{(k)} f_i(x) \right)$$
$$Z = \sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x) \right)$$

Let us consider a single component of the gradient, corresponding to a specific feature l and specific class k

$$\frac{d}{dw_l^{(k)}} \log P(y^{(j)}|x^{(j)}) =$$

For example, corresponding to $f_l : \text{contains('ski')}$

Contribution of the annotated document $(x^{(j)}, y^{(j)})$ to the loss

How does the gradient look like?

$$P(y = k|x) = \frac{1}{Z} \exp \left(\sum_i w_i^{(k)} f_i(x) \right)$$
$$Z = \sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x) \right)$$

Let us consider a single component of the gradient, corresponding to a specific feature l and specific class k

$$\frac{d}{dw_l^{(k)}} \log P(y^{(j)}|x^{(j)}) =$$
$$= \frac{d}{dw_l^{(k)}} \log \left(\exp \left(\sum_i w_i^{(y^{(j)})} f_i(x^{(j)}) \right) \right) - \frac{d \log Z}{dw_l^{(k)}}$$

How does the gradient look like?

$$P(y = k|x) = \frac{1}{Z} \exp \left(\sum_i w_i^{(k)} f_i(x) \right)$$
$$Z = \sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x) \right)$$

Let us consider a single component of the gradient, corresponding to a specific feature l and specific class k

$$\frac{d}{dw_l^{(k)}} \log P(y^{(j)}|x^{(j)}) =$$
$$= \frac{d}{dw_l^{(k)}} \log \left(\cancel{\exp \left(\sum_i w_i^{(y^{(j)})} f_i(x^{(j)}) \right)} \right) - \frac{d \log Z}{dw_l^{(k)}} = \text{(I)} - \text{(II)}$$

First term (I)

$$(I) = \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(y^{(j)})} f_i(x^{(j)}) \right)$$

First term (I)

$$(I) = \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(y^{(j)})} f_i(x^{(j)}) \right)$$

It is guaranteed to be 0 if $y^{(j)} \neq k$, otherwise - $f_l(x^{(j)})$

Let's re-write it as

First term (I)

$$(I) = \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(y^{(j)})} f_i(x^{(j)}) \right)$$

It is guaranteed to be 0 if $y^{(j)} \neq k$, otherwise - $f_l(x^{(j)})$

Let's re-write it as

$$= [y^{(j)} = k] \cdot f_l(x^{(j)})$$

[.] is the Iverson bracket:

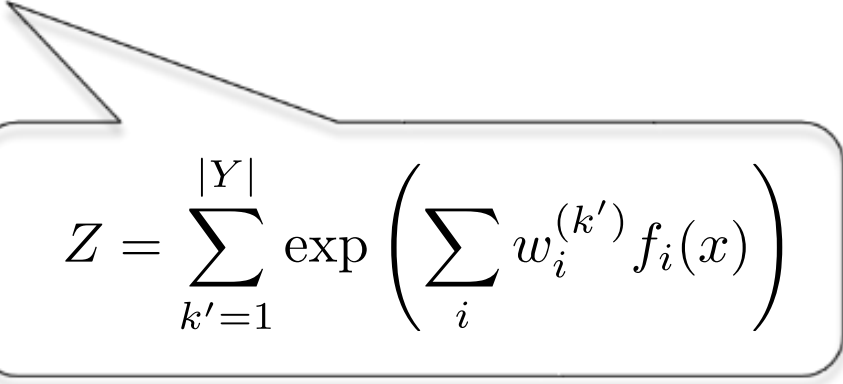
$$[S] \equiv \begin{cases} 0 & \text{if } S \text{ is false} \\ 1 & \text{if } S \text{ is true,} \end{cases}$$

Second term (II)

$$(II) = \frac{d \log Z}{dw_l^{(k)}}$$

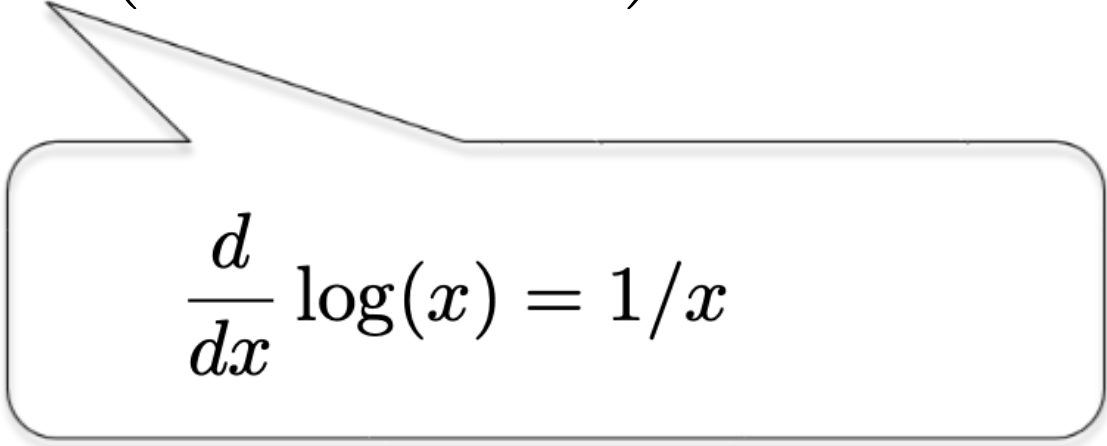
Second term (II)

$$(II) = \frac{d \log Z}{dw_l^{(k)}} = \frac{d}{dw_l^{(k)}} \log \left(\sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right) \right)$$


$$Z = \sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x) \right)$$

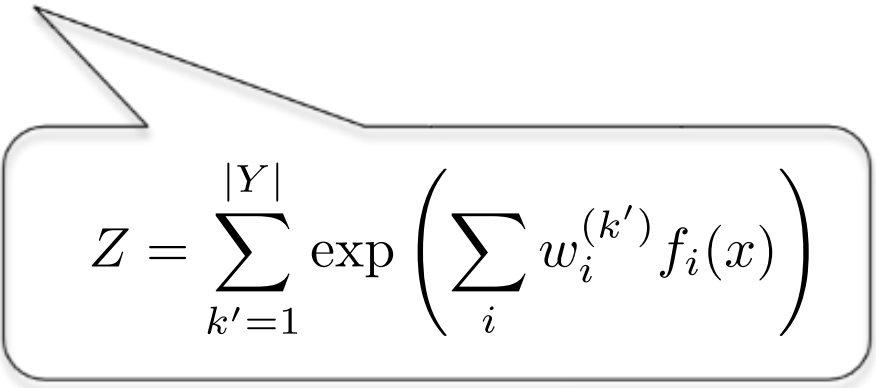
Second term (II)

$$\begin{aligned} \text{(II)} &= \frac{d \log Z}{dw_l^{(k)}} = \frac{d}{dw_l^{(k)}} \log \left(\sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right) \right) \\ &= \frac{\frac{d}{dw_l^{(k)}} \sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)}{\sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)} \end{aligned}$$


$$\frac{d}{dx} \log(x) = 1/x$$

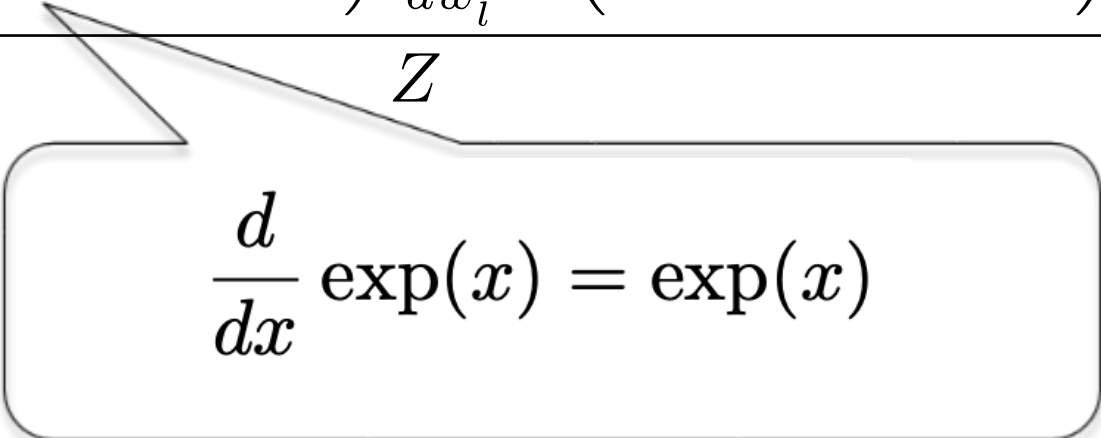
Second term (II)

$$\begin{aligned} \text{(II)} &= \frac{d \log Z}{dw_l^{(k)}} = \frac{d}{dw_l^{(k)}} \log \left(\sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right) \right) \\ &= \frac{\frac{d}{dw_l^{(k)}} \sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)}{\sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)} \\ &= \frac{\frac{d}{dw_l^{(k)}} \exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)}{Z} \end{aligned}$$


$$Z = \sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x) \right)$$

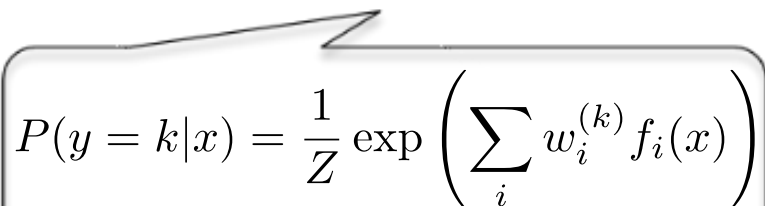
Second term (II)

$$\begin{aligned} \text{(II)} &= \frac{d \log Z}{dw_l^{(k)}} = \frac{d}{dw_l^{(k)}} \log \left(\sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right) \right) \\ &= \frac{\frac{d}{dw_l^{(k)}} \sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)}{\sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)} \\ &= \frac{\frac{d}{dw_l^{(k)}} \exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)}{Z} \\ &= \frac{\exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right) \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)}{Z} \end{aligned}$$


$$\frac{d}{dx} \exp(x) = \exp(x)$$

Second term (II)

$$\begin{aligned} \text{(II)} &= \frac{d \log Z}{dw_l^{(k)}} = \frac{d}{dw_l^{(k)}} \log \left(\sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right) \right) \\ &= \frac{\frac{d}{dw_l^{(k)}} \sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)}{\sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)} \\ &= \frac{\frac{d}{dw_l^{(k)}} \exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)}{Z} \\ &= \frac{\exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right) \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)}{Z} \\ &= P(y = k|x^{(j)}) \cdot \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right) \end{aligned}$$


$$P(y = k|x) = \frac{1}{Z} \exp \left(\sum_i w_i^{(k)} f_i(x) \right)$$

Second term (II)

$$\begin{aligned} \text{(II)} &= \frac{d \log Z}{dw_l^{(k)}} = \frac{d}{dw_l^{(k)}} \log \left(\sum_{k'=1}^{|Y|} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right) \right) \\ &= \frac{\frac{d}{dw_l^{(k)}} \sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)}{\sum_{k'} \exp \left(\sum_i w_i^{(k')} f_i(x^{(j)}) \right)} \\ &= \frac{\frac{d}{dw_l^{(k)}} \exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)}{Z} \\ &= \frac{\exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right) \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)}{Z} \\ &= P(y = k | x^{(j)}) \cdot \frac{d}{dw_l^{(k)}} \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right) \\ &= P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \end{aligned}$$

The feature value weighted by the probability assigned to the considered class k

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(\text{I})} - \textcolor{red}{(\text{II})} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= ([y^{(j)} = k] - P(y = k | x^{(j)})) f_l(x^{(j)})\end{aligned}$$

Bringing everything together

$$\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) = \textcolor{red}{(I)} - \textcolor{red}{(II)}$$

$$= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)})$$

$$= ([y^{(j)} = k] - P(y = k | x^{(j)})) f_l(x^{(j)})$$

What is the sign of this expression α ?

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

What is the sign of this expression α ?

Negative if the k-class is incorrect. (i.e. $k \neq y^{(j)}$)

Positive if the k-th class is correct

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(\text{I})} - \textcolor{red}{(\text{II})} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(\sum_i w_i^{(k)} f_i(x^{(j)}) \right)$$

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Recall that the form of update is

$$w_l^{(k)} \leftarrow w_l^{(k)} + \eta \cdot \frac{d \log P(y^j | x^{(j)})}{dw_l^{(k)}}$$

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Recall that the form of update is

$$w_l^{(k)} \leftarrow w_l^{(k)} + \eta \cdot \alpha \cdot f_l(x^{(j)})$$

Bringing everything together


$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Recall that the form of update is

$$w_l^{(k)} \leftarrow w_l^{(k)} + \eta \cdot \alpha \cdot f_l(x^{(j)})$$


$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left((w_l^{(k)} + \textcolor{red}{\eta \cdot \alpha \cdot f_l(x^{(j)})}) f_l(x^{(j)}) + \dots \right)$$

Bringing everything together


$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Recall that the form of update is

$$w_l^{(k)} \leftarrow w_l^{(k)} + \eta \cdot \alpha \cdot f_l(x^{(j)})$$


$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \underbrace{\eta \cdot \alpha \cdot f_l^2(x^{(j)})}_{\substack{> 0 \\ \geq 0}} + \dots \right)$$

Bringing everything together

$$\begin{aligned}
 \frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \text{(I)} - \text{(II)} \\
 &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\
 &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\alpha} f_l(x^{(j)})
 \end{aligned}$$

Let's see what happens to the probability

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Recall that the form of update is

$$w_l^{(k)} \leftarrow w_l^{(k)} + \eta \cdot \alpha \cdot f_l(x^{(j)})$$

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \underbrace{\eta \cdot \alpha \cdot f_l^2(x^{(j)})}_{\substack{> 0 \\ \geq 0 \text{ if } k \text{ is correct} \\ \leq 0, \text{ otherwise}}} + \dots \right)$$

> 0 ≥ 0 if k is correct
 ≤ 0 , otherwise ≥ 0

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \text{(I)} - \text{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\alpha} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

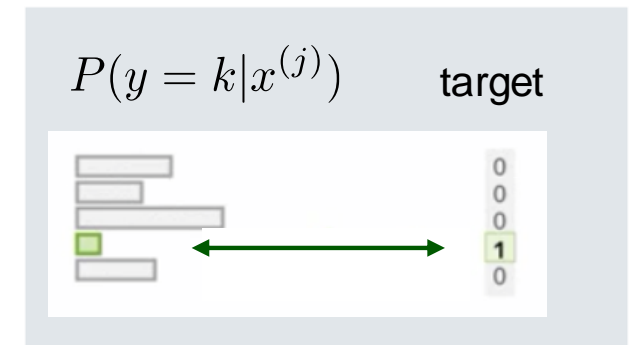
$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Recall that the form of update is

$$w_l^{(k)} \leftarrow w_l^{(k)} + \eta \cdot \alpha \cdot f_l(x^{(j)})$$

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \underbrace{\eta \cdot \alpha \cdot f_l^2(x^{(j)})}_{\text{update}} + \dots \right)$$

> 0 $\begin{matrix} \geq 0 & \text{if } k \text{ is correct} \\ \leq 0 & \text{otherwise} \end{matrix}$ ≥ 0



Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \text{(I)} - \text{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\alpha} f_l(x^{(j)})\end{aligned}$$

Let's see what happens to the probability

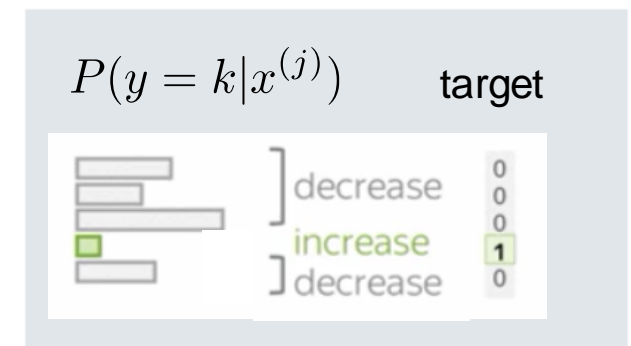
$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \dots \right)$$

Recall that the form of update is

$$w_l^{(k)} \leftarrow w_l^{(k)} + \eta \cdot \alpha \cdot f_l(x^{(j)})$$

$$P(y = k | x^{(j)}) = \frac{1}{Z} \exp \left(w_l^{(k)} f_l(x^{(j)}) + \underbrace{\eta \cdot \alpha \cdot f_l^2(x^{(j)})}_{\text{update}} + \dots \right)$$

> 0 $\begin{matrix} \geq 0 & \text{if } k \text{ is correct} \\ \leq 0 & \text{otherwise} \end{matrix}$ ≥ 0



Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

When is α close to zero?

Bringing everything together

$$\begin{aligned}\frac{d}{dw_l^{(k)}} \log P(y^{(j)} | x^{(j)}) &= \textcolor{red}{(I)} - \textcolor{red}{(II)} \\ &= [y^{(j)} = k] \cdot f_l(x^{(j)}) - P(y = k | x^{(j)}) \cdot f_l(x^{(j)}) \\ &= \underbrace{([y^{(j)} = k] - P(y = k | x^{(j)}))}_{\textcolor{red}{\alpha}} f_l(x^{(j)})\end{aligned}$$

Close to zero if the classifier confidently predicts the correct class

$$P(y = k | x^{(j)}) \approx \begin{cases} 1 & \text{if } y^{(j)} = k \\ 0 & \text{otherwise} \end{cases}$$

If the classifier is already confident, gradient is close to 0 and no learning is happening

Relation to Naive Bayes

f_1 : contains('ski')

$$w_1^{(1)} = \log \hat{P}(\text{'ski'}|c = 1)$$

$$w_1^{(2)} = \log \hat{P}(\text{'ski'}|c = 2)$$

$$w_1^{(3)} = \log \hat{P}(\text{'ski'}|c = 3)$$

f_2 : contains('beach')

$$w_2^{(1)} = \log \hat{P}(\text{'beach'}|c = 1)$$

$$w_2^{(2)} = \log \hat{P}(\text{'beach'}|c = 2)$$

$$w_2^{(3)} = \log \hat{P}(\text{'beach'}|c = 3)$$

f_3 : **1**

$$w_3^{(1)} = \log \hat{P}(c = 1)$$

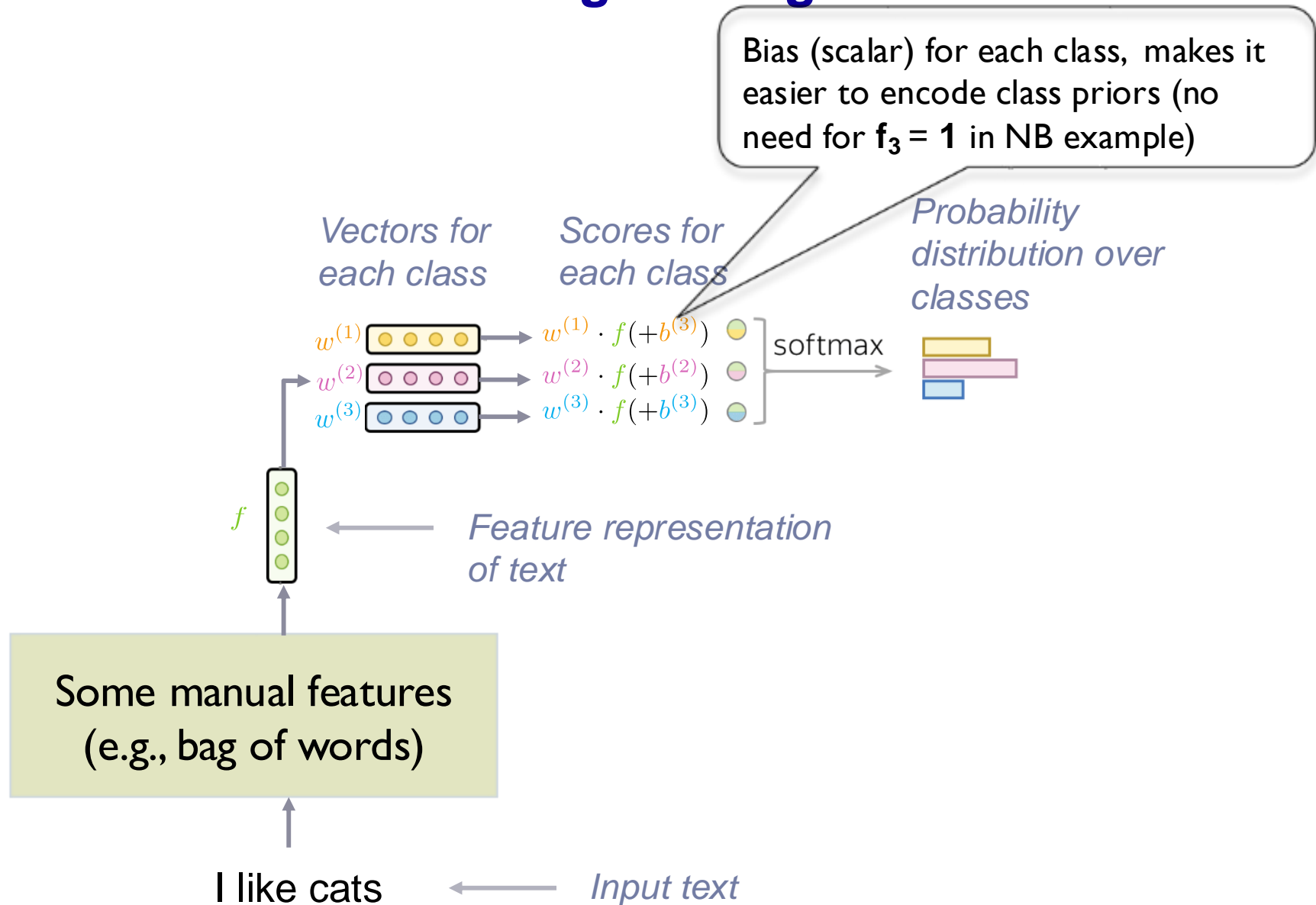
$$w_3^{(2)} = \log \hat{P}(c = 2)$$

$$w_3^{(3)} = \log \hat{P}(c = 3)$$

Relation to Naive Bayes (continued)

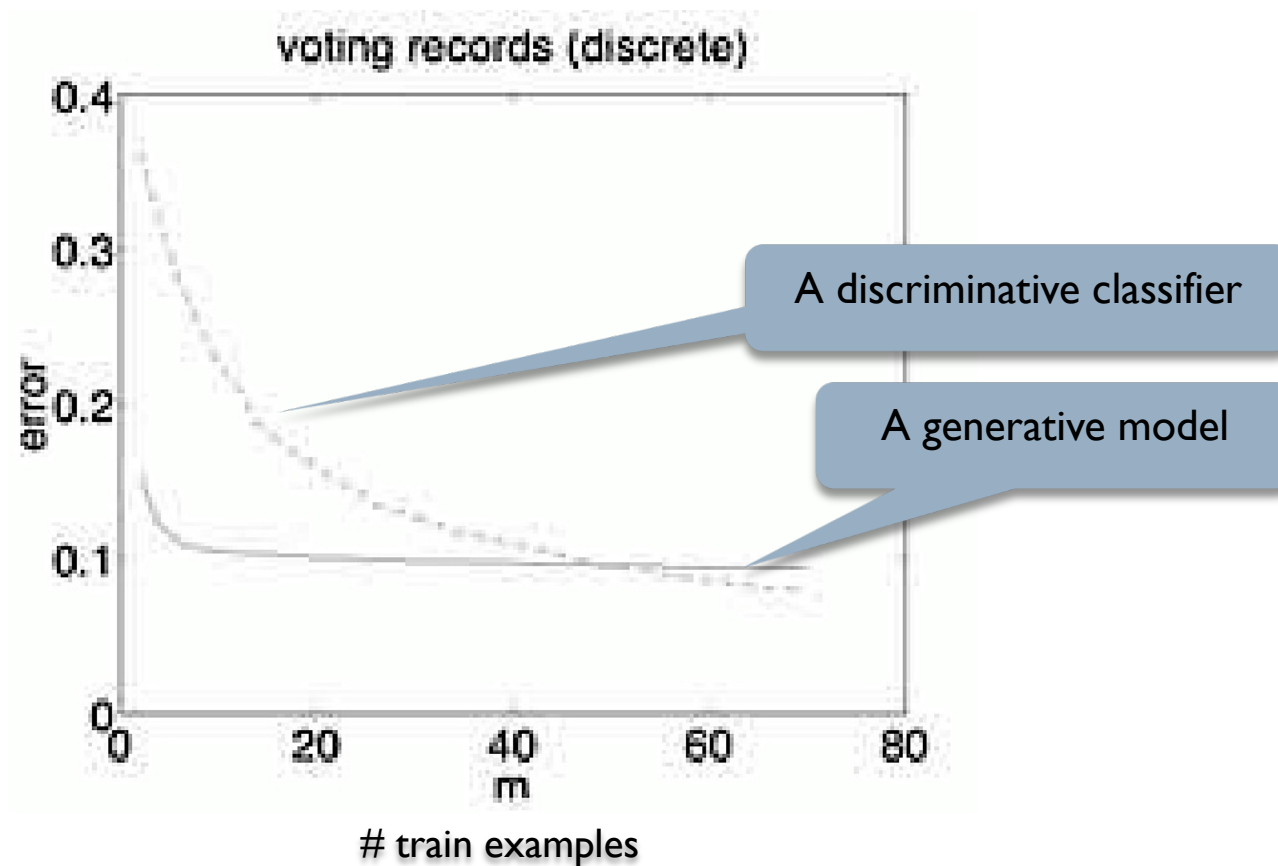
- Naive Bayes is also a linear classifier, and can be expressed in the same form
- Should the features be actually independent (will never happen), they would converge to the same solution as the amount of training data increases

Schematic view of logistic regression



NB vs MaxEnt dependence on dataset size

- Theoretical results: generative classifiers converge faster with training set size to their optimal error [Ng & Jordan, NeurIPS 2001]
- Empirical:



Predicting Democrat
vs Republican, based
on voting records

The downside to Logistic Regressions

- Supervised MLE in generative models is easy: compute counts and normalize.
- Supervised CMLE in LR model not so easy
 - * requires multiple iterations over the data to gradually improve weights (using gradient ascent).
 - * each iteration computes $P(y^{(j)}|x^{(j)})$ for all j .
 - * this can be time-consuming, especially if there are a large number of classes and/or thousands of features to extract from each training example.

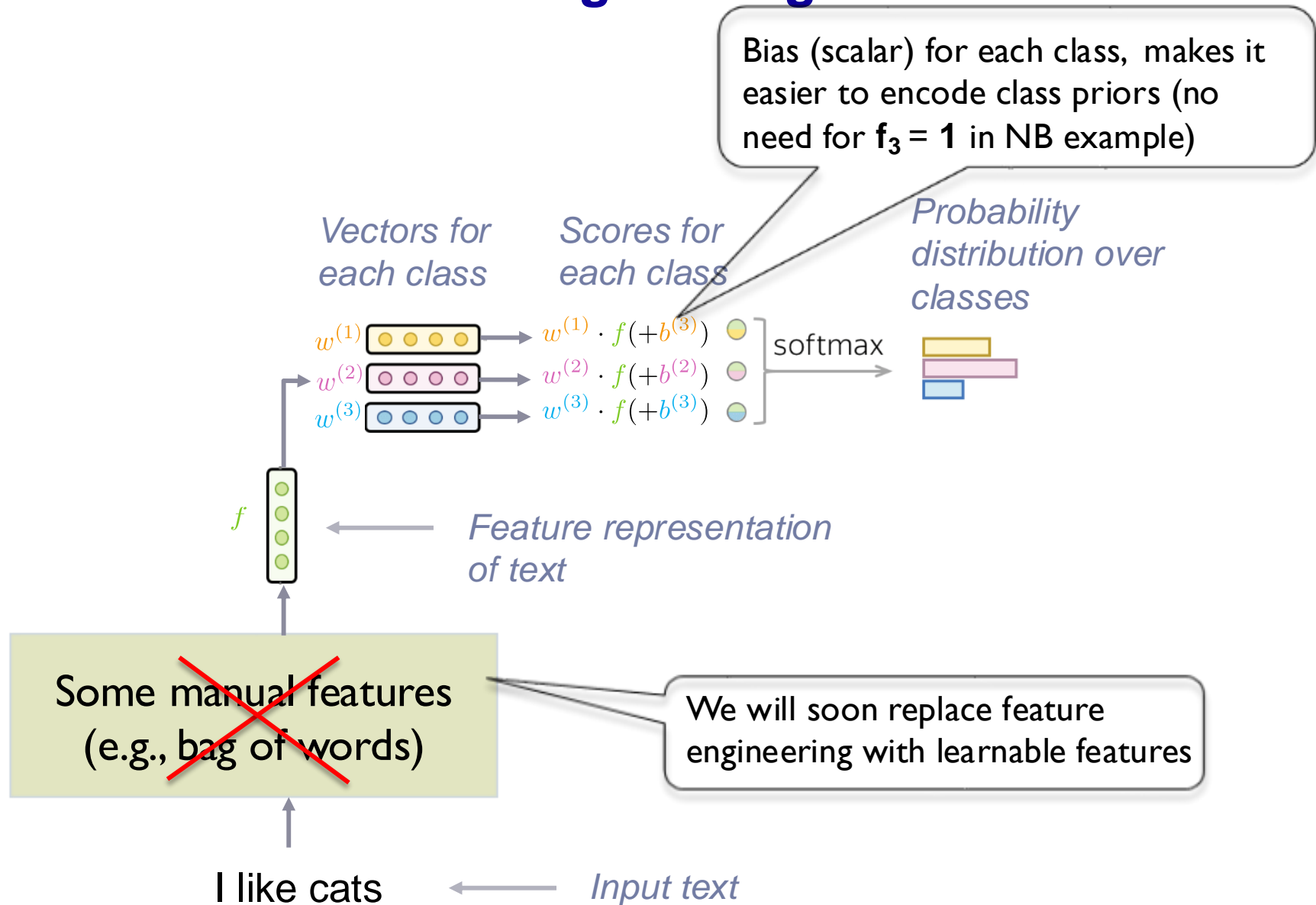
Robustness: LR and Naive Bayes

- Imagine that in training there is one very frequent predictive feature
 - * E.g., in training sentiment data contained emoticons but not at test time
- The model can quickly learn to rely on this feature
 - * model is confident on examples with emoticons
 - * the gradient on these examples gets close to zero
 - * the model does not learn other features

Robustness (continued)

- In LR, a feature weight will depend on the presence of other **predictive features**
- Naive Bayes will rely on all features
 - * The weight of a feature is not affected by how predictive other features are
- This makes NB more robust than (**basic**) Logistic Regression when test data is (distributionally) different from train data

Schematic view of logistic regression



Summary

- Two methods for text classification: Naive Bayes and Logistic Regression
- Make different independence assumptions, have different training requirements.
- Both are easily available in standard ML toolkits.
 - * But you now also know how to implement them!
- Both require some work to figure out what features are good to use.
 - * Soon, we will see how to alleviate the need for feature engineering