



TEXT CLASSIFICATION

Sandeep Soni

02/20/2024

QUESTION FOR THE DAY

“How to predict a label for given text?”

AGENDA

- Classification problem
- Naive Bayes
- Logistic regression
- Training/test setup
- Evaluation metrics

CLASSIFICATION

- Input: document (e.g., email)
- Output: label (e.g., spam/ham)



CLASSIFICATION PROBLEMS

Task	\mathcal{X}	\mathcal{Y}
Language ID	text	{english, mandarin, hindi, ...}
spam classification	email	{spam, ham}
party affiliation	speech	{republican, democrat}
sentiment analysis	text	{positive, negative, mixed, neutral}
music genre	lyric	{rock, pop, jazz, rap,...}

FORMAL TASK



FORMAL TASK

- x is input (e.g., an email)
- $x \in \mathcal{X}$ (e.g., set of emails)



FORMAL TASK

- x is input (e.g., an email)
 - $x \in \mathcal{X}$ (e.g., set of emails)
- y is output (e.g., spam)
 - $y \in \mathcal{Y}$ (e.g., {spam, ham})



FORMAL TASK

- x is input (e.g., an email)
 - $x \in \mathcal{X}$ (e.g., set of emails)
- y is output (e.g., spam)
 - $y \in \mathcal{Y}$ (e.g., {spam, ham})
- $y = h(x)$
 - h maps instances to labels



CLASSIFICATION

- The true mapping function h is not known to us so we want to find \hat{h} that's a **closest** approximation



RULE BASED CLASSIFICATION

- \hat{h} (“I wish to discuss personal investment business matters with you so as to be able to learn of the available investment opportunities in your region or country.”)

\hat{h}

if email contains phrase
“investment opportunities” then
spam



SUPERVISED LEARNING

- Learn \hat{h} from training data given in the form of $\langle x, y \rangle$ pairs



$$\hat{h}(x)$$

- We still need to resolve:
 - How to learn this mapping function? (e.g., which method to use, how to optimize, etc)
 - How to represent the input to this function?

NAIVE BAYES

- One simple yet quite effective classification method is Naive Bayes
 - Similar to LDA, it's a generative model
 - We'll represent input text as a bag of words vector

REFRESHER: CHAIN RULE

- If x and y are random variables, the joint probability can be factorized using chain rule

$$P(x, y) = P(y)P(x | y)$$

$$P(x, y) = P(x)P(y | x)$$

REFRESHER: CHAIN RULE

- If there are three variables, then chain rule gives:

$$P(x, y, z) = P(x)P(y | x)P(z | x, y)$$

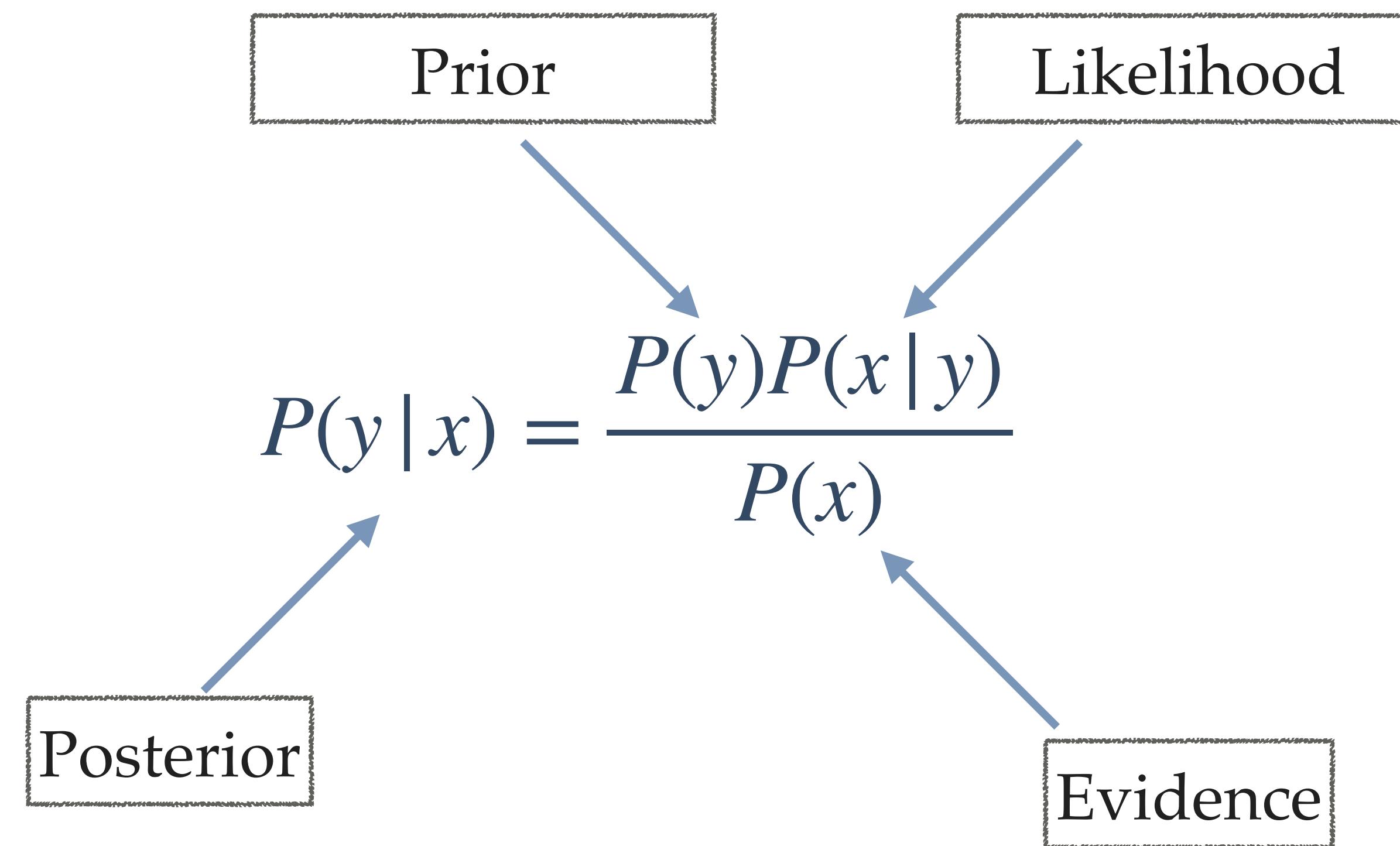
- Just like two variables, permutations are equivalent

REFRESHER: CHAIN RULE

- In general, for multiple variables, chain rule is:

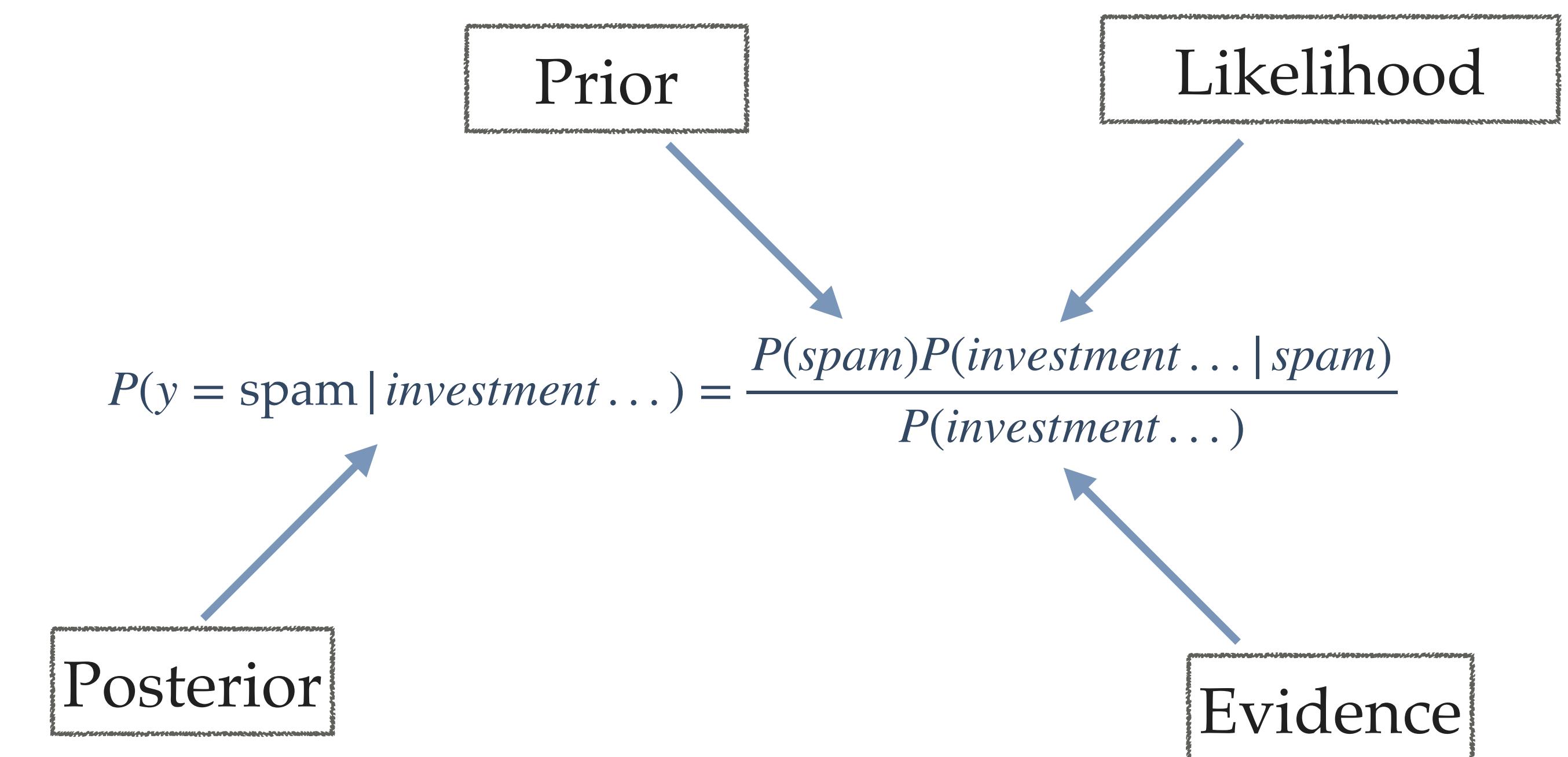
$$P(y, x_1, x_2, \dots, x_n) = P(y)P(x_1 | y)P(x_2 | x_1, y)\dots P(x_n | x_{n-1}, \dots, y)$$

REFRESHER: BAYES THEOREM



REFRESHER: BAYES THEOREM

- For spam classification:
- **Prior** how probable is to see the spam label
- **Likelihood** how likely are these words in the email for spam
- **Evidence** how probable are these words in the email
- **Posterior** how probable is spam label on this email



REFRESHER: BAYES THEOREM

- We can express the marginal probability in the denominator as:

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)}$$

$$P(y|x) = \frac{P(y)P(x|y)}{\sum_{y'} P(x, y')}$$

$$P(y|x) = \frac{P(y)P(x|y)}{\sum_{y'} P(y')P(x|y')}$$

REFRESHER: PROBABILITY DISTRIBUTIONS

Y is binary RV

Y	P(Y)
TRUE	0.25
FALSE	0.75

Marginal distribution

X and Y both
binary RVs

X	Y	P(X,Y)
TRUE	TRUE	0.1
TRUE	FALSE	0.3
FALSE	TRUE	0.25
FALSE	FALSE	0.35

Joint distribution

X and Y both
binary RVs

P(Y X)	X=TRUE	X=FALSE
Y=TRUE	0.2	0.8
Y=FALSE	0.7	0.3

Conditional distribution

REFRESHER: PROBABILITY DISTRIBUTIONS

Y is binary RV

Y	P(Y)
TRUE	0.25
FALSE	0.75

Marginal distribution

To specify this distribution, we need 1 parameter

If X can take n values then n-1 parameters

X and Y both binary RVs

X	Y	P(X,Y)
TRUE	TRUE	0.1
TRUE	FALSE	0.3
FALSE	TRUE	0.25
FALSE	FALSE	0.35

Joint distribution

X and Y both binary RVs

P(Y X)	X=TRUE	X=FALSE
Y=TRUE	0.2	0.8
Y=FALSE	0.7	0.3

Conditional distribution

REFRESHER: PROBABILITY DISTRIBUTIONS

Y is binary RV

Y	P(Y)
TRUE	0.25
FALSE	0.75

Marginal distribution

To specify this distribution, we need 1 parameter

If X can take n values then n-1 parameters

X and Y both binary RVs

X	Y	P(X,Y)
TRUE	TRUE	0.1
TRUE	FALSE	0.3
FALSE	TRUE	0.25
FALSE	FALSE	0.35

Joint distribution

To specify this distribution we need 3 parameters

If there are K binary random variables, then $2^k - 1$

X and Y both binary RVs

P(Y X)	X=TRUE	X=FALSE
Y=TRUE	0.2	0.8
Y=FALSE	0.7	0.3

Conditional distribution

REFRESHER: PROBABILITY DISTRIBUTIONS

Y is binary RV

Y	P(Y)
TRUE	0.25
FALSE	0.75

Marginal distribution

To specify this distribution, we need 1 parameter

If X can take n values then n-1 parameters

X and Y both binary RVs

X	Y	P(X,Y)
TRUE	TRUE	0.1
TRUE	FALSE	0.3
FALSE	TRUE	0.25
FALSE	FALSE	0.35

Joint distribution

To specify this distribution we need 3 parameters

If there are K binary random variables, then $2^k - 1$

X and Y both binary RVs

P(Y X)	X=TRUE	X=FALSE
Y=TRUE	0.2	0.8
Y=FALSE	0.7	0.3

Conditional distribution

To specify this distribution we need 2 parameters

If X takes u different values and Y takes v different values, we need $v(u-1)$ parameters

TOWARDS CLASSIFICATION

- One way to learn the mapping $\hat{h}(x)$ is to learn the posterior distribution $P(y|x)$
- x is text, so not just one random variable but a bunch of variables, so we have to estimate $P(y|x_1, x_2, x_3, \dots, x_n)$

TOWARDS CLASSIFICATION

$$P(y | x_1, x_2, \dots, x_n) = \frac{P(y, x_1, x_2, \dots, x_n)}{P(x_1, x_2, \dots, x_n)}$$

TOWARDS CLASSIFICATION

- Joint probability in the numerator can be factorized using chain rule

$$P(y | x_1, x_2, \dots, x_n) = \frac{P(y)P(x_1, x_2, \dots, x_n | y)}{P(x_1, x_2, \dots, x_n)}$$

Continue to apply chain rule to the numerator (denominator is ignored for now)

$$P(y | x_1, x_2, \dots, x_n) \propto P(y)P(x_1, x_2, \dots, x_n | y)$$

$$\propto P(y)P(x_1 | y)P(x_2, \dots, x_n | y, x_1)$$

...

$$\propto P(y)P(x_1 | y)P(x_2 | x_1, y)P(x_3 | x_2, x_1, y) \dots P(x_n | x_{n-1}, \dots, x_1, y)$$

This is intractable!

Continue to apply chain rule to the numerator (denominator is ignored for now)

$$P(y | x_1, x_2, \dots, x_n) \propto P(y)P(x_1, x_2, \dots, x_n | y)$$

$$\propto P(y)P(x_1 | y)P(x_2, \dots, x_n | y, x_1)$$

...

$$\propto P(y)P(x_1 | y)P(x_2 | x_1, y)P(x_3 | x_2, x_1, y) \dots P(x_n | x_{n-1}, \dots, x_1, y)$$

This is intractable!

NAIVE BAYES

- Naive conditional independence assumption

$$P(x_i | x_{i-1}, x_{i-2}, \dots, y) = P(x_i | y)$$

- Given the category, the words (features) are independent of each other
- Under naive Bayes, $P(\text{"prince"} | \text{spam}) = P(\text{"prince"} | \text{spam, "kenyan"})$

Now we can rewrite the posterior probability as:

$$\begin{aligned} P(y|x_1, x_2, \dots, x_n) &\propto P(y)P(x_1|y)P(x_2|y)P(x_3|y)\dots P(x_n|y) \\ &\propto P(y) \prod_{i=1}^n P(x_i|y) \end{aligned}$$

This is tractable!

We can estimate these probabilities by simply counting the instances in training data

$$P(y = \text{spam}) = \frac{\#\text{samples labeled spam}}{\#\text{ samples}}$$

$$P(\text{"kenyan"} | \text{spam}) = \frac{\#\text{samples labeled spam and contain "kenyan"}}{\#\text{samples labeled spam}}$$

PICKING THE LABEL

- Once you estimate the probabilities from training data, you can pick the label that maximizes the posterior

$$P(y = \text{spam} \mid \text{text}) > P(y = \text{ham} \mid \text{text}) \quad \text{Spam}$$

otherwise Ham

NAIVE BAYES

NAIVE BAYES

- Let's rewrite the posterior probability under naive Bayes

NAIVE BAYES

- Let's rewrite the posterior probability under naive Bayes

$$P(y|x) \propto P(y)P(x_1|y)P(x_2|y)P(x_3|y)\dots P(x_n|y)$$

NAIVE BAYES

- Let's rewrite the posterior probability under naive Bayes

$$\begin{aligned} P(y|x) &\propto P(y)P(x_1|y)P(x_2|y)P(x_3|y)\dots P(x_n|y) \\ &\propto P(y) \prod_{i=1}^n P(x_i|y) \end{aligned}$$

NAIVE BAYES

- Let's rewrite the posterior probability under naive Bayes

$$\begin{aligned} P(y|x) &\propto P(y)P(x_1|y)P(x_2|y)P(x_3|y)\dots P(x_n|y) \\ &\propto P(y) \prod_{i=1}^n P(x_i|y) \end{aligned}$$

- If we take a log of both sides, we get

NAIVE BAYES

- Let's rewrite the posterior probability under naive Bayes

$$\begin{aligned} P(y|x) &\propto P(y)P(x_1|y)P(x_2|y)P(x_3|y)\dots P(x_n|y) \\ &\propto P(y) \prod_{i=1}^n P(x_i|y) \end{aligned}$$

- If we take a log of both sides, we get

$$\log P(y|x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i|y)$$

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i \mid y)$$

$$\log P(y \mid x) \propto 1 \cdot \log P(y) + \sum_{i=1}^N 1 \cdot \log P(x_i \mid y)$$

$$\log P(y|x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i|y)$$

$$\log P(y|x) \propto 1 \cdot \log P(y) + \sum_{i=1}^N 1 \cdot \log P(x_i|y)$$

$$\log P(y|x) \propto f_0 \cdot w_0 + \sum_{i=1}^N f_i \cdot w_i$$

$$\log P(y|x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i|y)$$

$$\log P(y|x) \propto 1 \cdot \log P(y) + \sum_{i=1}^N 1 \cdot \log P(x_i|y)$$

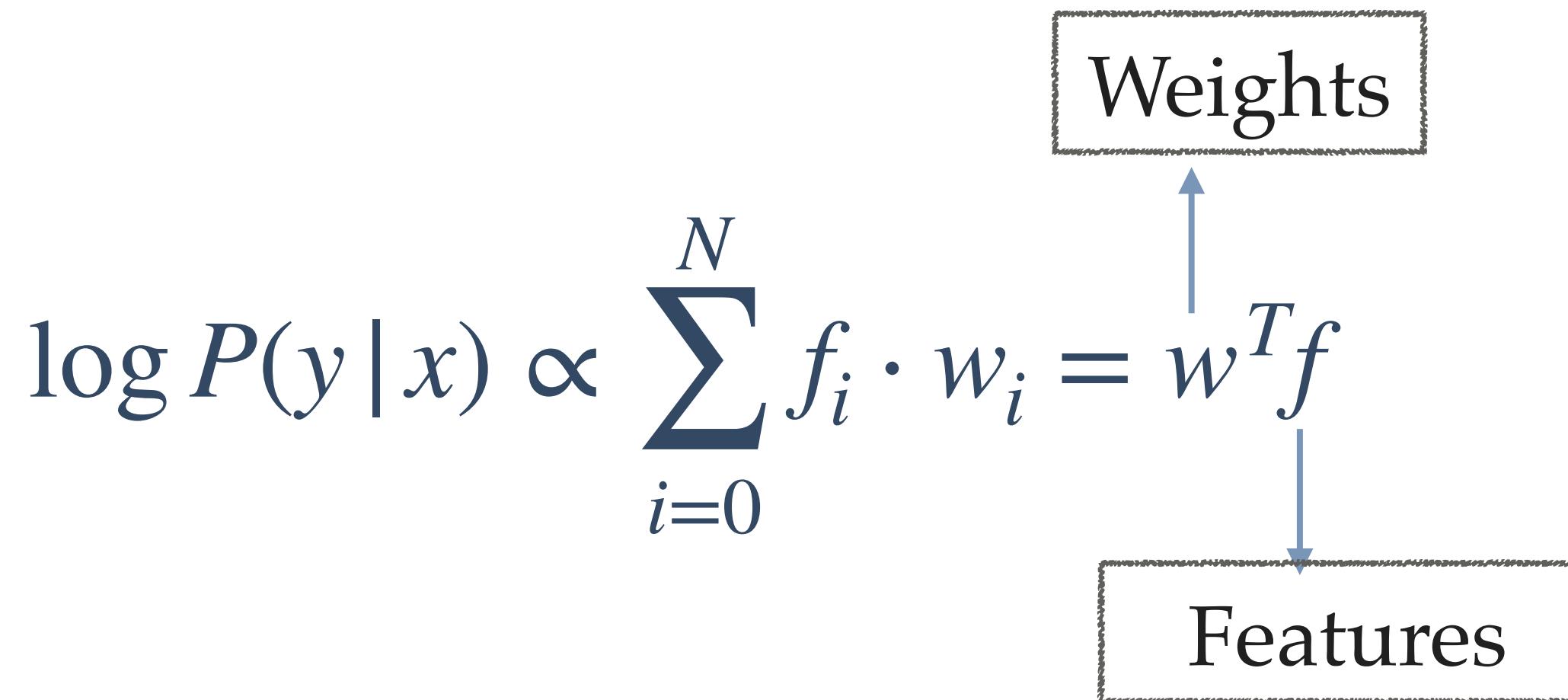
$$\log P(y|x) \propto f_0 \cdot w_0 + \sum_{i=1}^N f_i \cdot w_i$$

$$\log P(y|x) \propto \sum_{i=0}^N f_i \cdot w_i = w^T f$$

$$\log P(y|x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i|y)$$

$$\log P(y|x) \propto 1 \cdot \log P(y) + \sum_{i=1}^N 1 \cdot \log P(x_i|y)$$

$$\log P(y|x) \propto f_0 \cdot w_0 + \sum_{i=1}^N f_i \cdot w_i$$



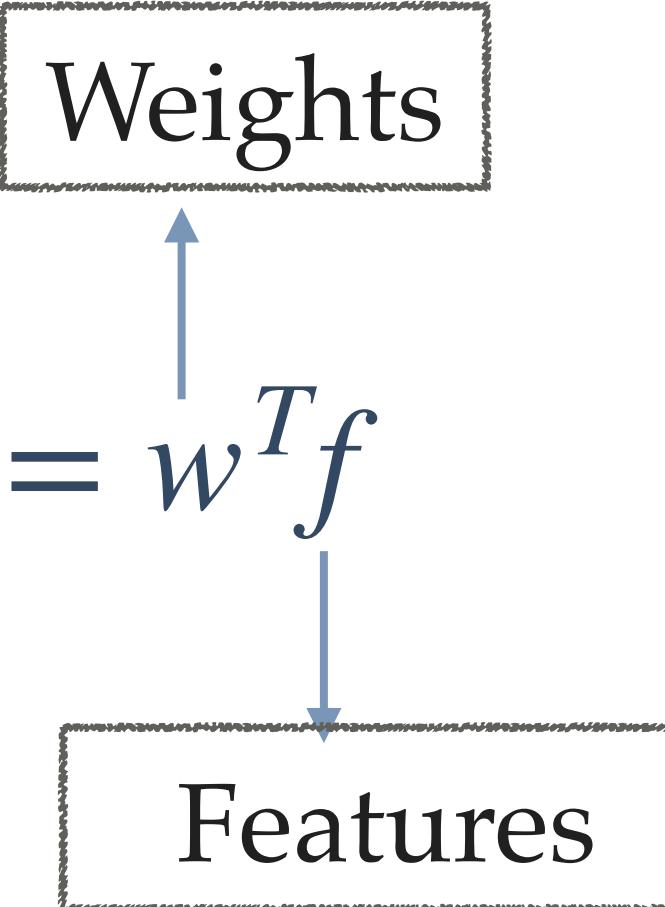
$$\log P(y|x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i|y)$$

$$\log P(y|x) \propto 1 \cdot \log P(y) + \sum_{i=1}^N 1 \cdot \log P(x_i|y)$$

$$\log P(y|x) \propto f_0 \cdot w_0 + \sum_{i=1}^N f_i \cdot w_i$$

Naive Bayes is a
linear model

$$\log P(y|x) \propto \sum_{i=0}^N f_i \cdot w_i = w^T f$$



LOGISTIC REGRESSION

LOGISTIC REGRESSION

In logistic regression, we directly model the conditional probability of the label given the text

LOGISTIC REGRESSION

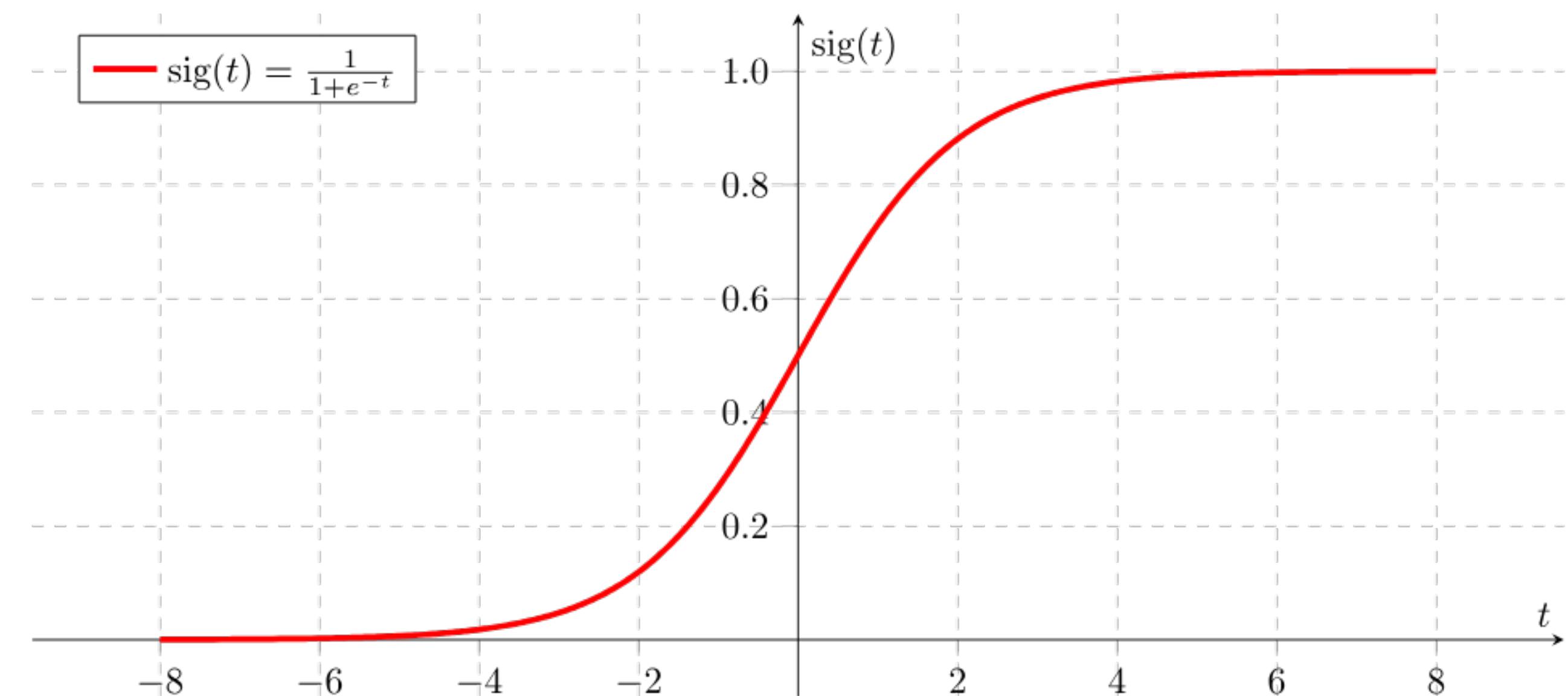
In logistic regression, we directly model the conditional probability of the label given the text

$$P(y|x) = \text{sig}(w^T f(x))$$

LOGISTIC REGRESSION

In logistic regression, we directly model the conditional probability of the label given the text

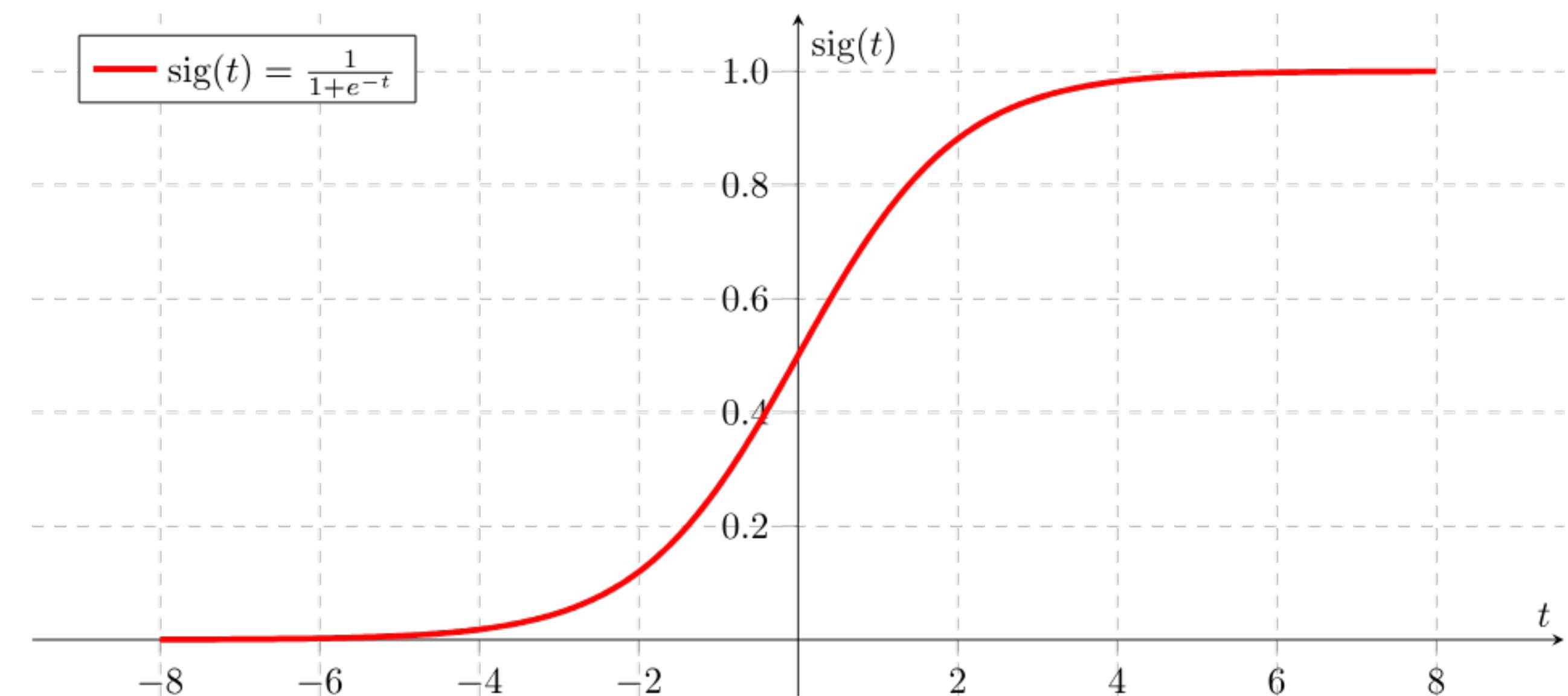
$$P(y|x) = \text{sig}(w^T f(x))$$



LOGISTIC REGRESSION

In logistic regression, we directly model the conditional probability of the label given the text

$$P(y|x) = \text{sig}(w^T f(x))$$



This is highly flexible because we can encode any type of features that we think could be useful

LEARNING

LEARNING

- The objective is to learn the vector of weights w

LEARNING

- The objective is to learn the vector of weights w
- Unlike Naive Bayes, no closed form solution

LEARNING

- The objective is to learn the vector of weights w
- Unlike Naive Bayes, no closed form solution
- Weights are learned by casting classification as an optimization problem

LEARNING

- The objective is to learn the vector of weights w
- Unlike Naive Bayes, no closed form solution
- Weights are learned by casting classification as an optimization problem
- Weights typically learned by iterative algorithms on training data

GENERALIZED CLASSIFICATION

GENERALIZED CLASSIFICATION

- Linear models are not the only models we can use

GENERALIZED CLASSIFICATION

- Linear models are not the only models we can use
- Features can be transformed by one or more non-linear functions

GENERALIZED CLASSIFICATION

- Linear models are not the only models we can use
- Features can be transformed by one or more non-linear functions
- $P(y|x) = \text{sig}(g(f(x)))$

GENERALIZED CLASSIFICATION

- Linear models are not the only models we can use
- Features can be transformed by one or more non-linear functions
- $P(y|x) = \text{sig}(g(f(x)))$
- For multi-class classification, $P(y|x) = \text{softmax}(g(f(x)))$

GENERALIZED CLASSIFICATION

- Linear models are not the only models we can use
- Features can be transformed by one or more non-linear functions
- $P(y|x) = \text{sig}(g(f(x)))$ g could be learned using a neural network
- For multi-class classification, $P(y|x) = \text{softmax}(g(f(x)))$

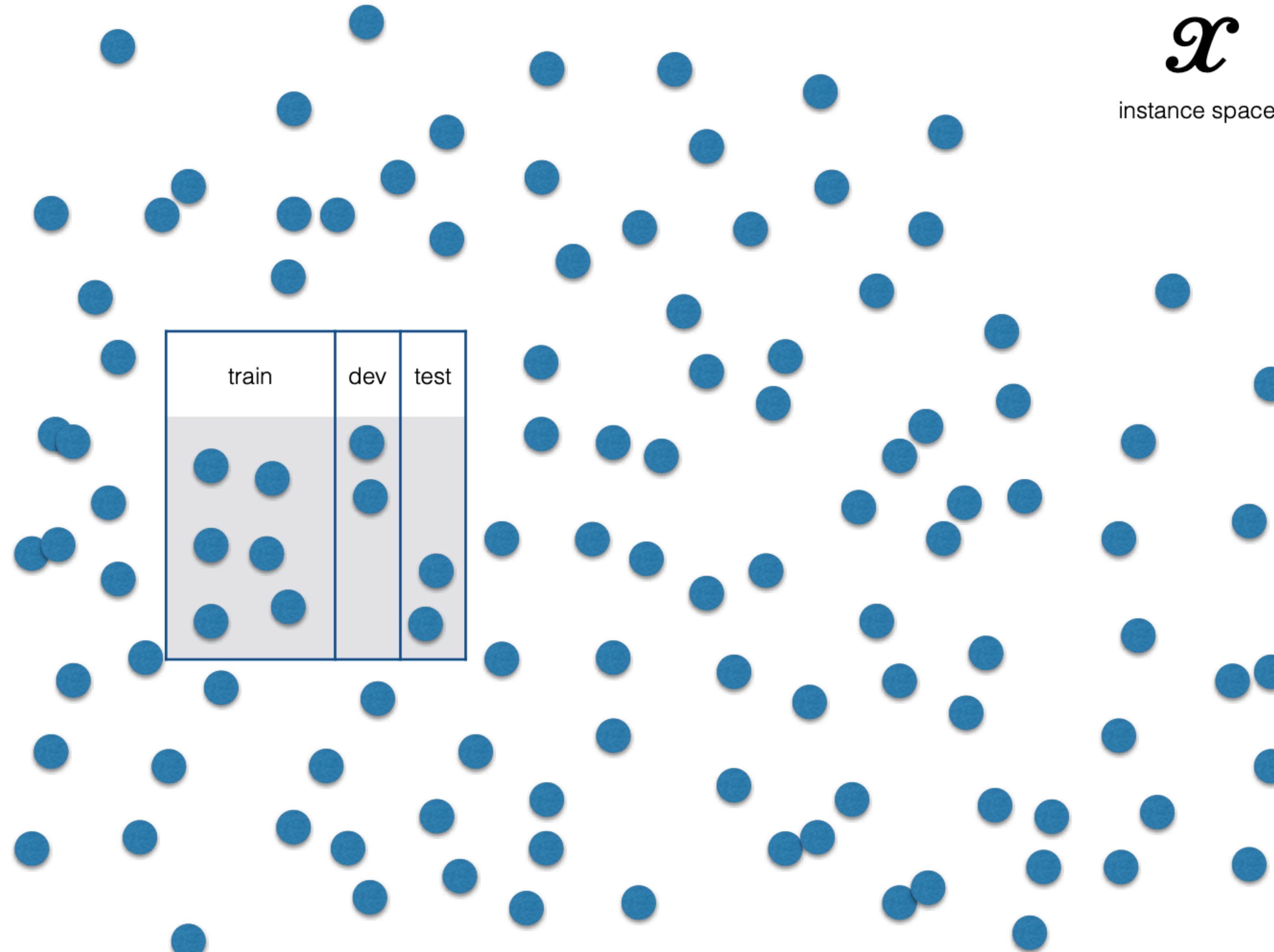
FORECASTING

FORECASTING

- Instead of converting the scores into probabilities, the same models can be used to predict a continuous value from text
- $y = g(f(x))$
- If g is a linear function, this is linear regression

\mathcal{X}

instance space



EXPERIMENT DESIGN

- Training set is to estimate parameters of the model
- Development set is to perform model selection
- Test set for evaluation

EXPERIMENT DESIGN

- Typically, we use 80% data for training, 10% for model selection and 10% for evaluation
- One should be careful never to use development or test data to do estimation

FEATURE FUNCTION

FEATURE FUNCTION

- Feature function transforms text into a feature vector
 - e.g. text → counts vector
 - e.g. text → counts vector | topics proportion

REGULARIZATION

REGULARIZATION

- To enforce some structure or bake in some domain expertise, we use regularization by adding penalty terms to our optimization objective (e.g., if we don't want very high weights, we'll add a penalty to our loss that we're minimizing)

REGULARIZATION

- To enforce some structure or bake in some domain expertise, we use regularization by adding penalty terms to our optimization objective (e.g., if we don't want very high weights, we'll add a penalty to our loss that we're minimizing)
- The value of the penalty is controlled by a hyper parameter which should be tuned on the development set

How do we know if our learned classifier is good?

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

	1	2	3	4	5	6	7	8	9	10
y	spam	spam	spam	spam	spam	ham	ham	ham	ham	ham
yhat	spam	spam	ham	spam	spam	spam	ham	spam	ham	spam

CONFUSION MATRIX

CONFUSION MATRIX

Predicted \ Observed		y=spam	y=ham
yhat = spam	4	3	
yhat=ham	1	2	

CONFUSION MATRIX

CONFUSION MATRIX

Predicted \ Observed		y=spam	y=ham
yhat = spam	True positives	False positives	
yhat=ham	False negatives	True negatives	

CONFUSION MATRIX

Predicted \ Observed		y=spam	y=ham
yhat = spam	True positives	False positives	
yhat=ham	False negatives	True negatives	

- $N = \text{True positives} + \text{True negatives} + \text{False positives} + \text{False negatives}$

ACCURACY

ACCURACY

Predicted \ Observed		$y = \text{spam}$	$y = \text{ham}$
$\hat{y} = \text{spam}$	True positives	False positives	
$\hat{y} = \text{ham}$	False negatives	True negatives	

ACCURACY

$$\frac{\text{True positives} + \text{True negatives}}{N}$$

Predicted \ Observed		y=spam	y=ham
yhat = spam	True positives	False positives	
yhat=ham	False negatives	True negatives	

ACCURACY

$$\frac{\text{True positives} + \text{True negatives}}{N}$$

Predicted \ Observed		y=spam	y=ham
yhat = spam	True positives	False positives	
yhat=ham	False negatives	True negatives	

- We want the accuracy of the classifier to be high

PRECISION

PRECISION

Predicted \ Observed		$y = \text{spam}$	$y = \text{ham}$
$\hat{y} = \text{spam}$	True positives	False positives	
$\hat{y} = \text{ham}$	False negatives	True negatives	

PRECISION

$$\frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

Predicted \ Observed		y=spam	y=ham
yhat = spam	True positives	False positives	
yhat=ham	False negatives	True negatives	

PRECISION

$$\frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

Predicted \ Observed		y=spam	y=ham
yhat = spam	True positives		False positives
	False negatives	True negatives	
yhat=ham			

- We want number of false positives to be low and precision to be high

RECALL

RECALL

Predicted \ Observed		$y = \text{spam}$	$y = \text{ham}$
$\hat{y} = \text{spam}$	True positives	False positives	
$\hat{y} = \text{ham}$	False negatives	True negatives	

RECALL

$$\frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

Predicted \ Observed		$y = \text{spam}$	$y = \text{ham}$
$y_{\hat{}} = \text{spam}$	True positives	False positives	
$y_{\hat{}} = \text{ham}$	False negatives	True negatives	

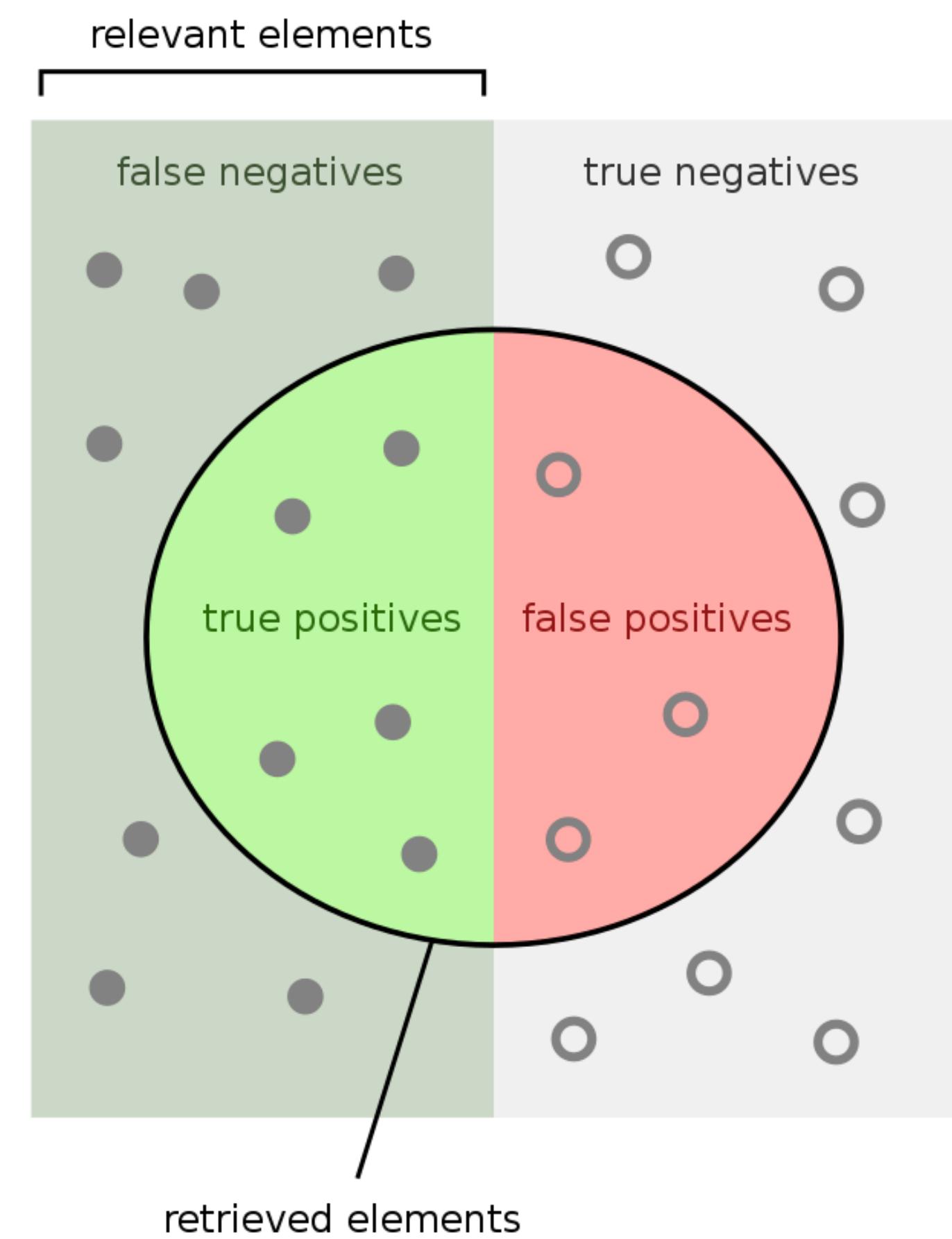
RECALL

$$\frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

Predicted \ Observed		y=spam	y=ham
yhat = spam	True positives		False positives
	False negatives	True negatives	
yhat=ham			

- We want number of false negatives to be low and recall to be high

PRECISION AND RECALL



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

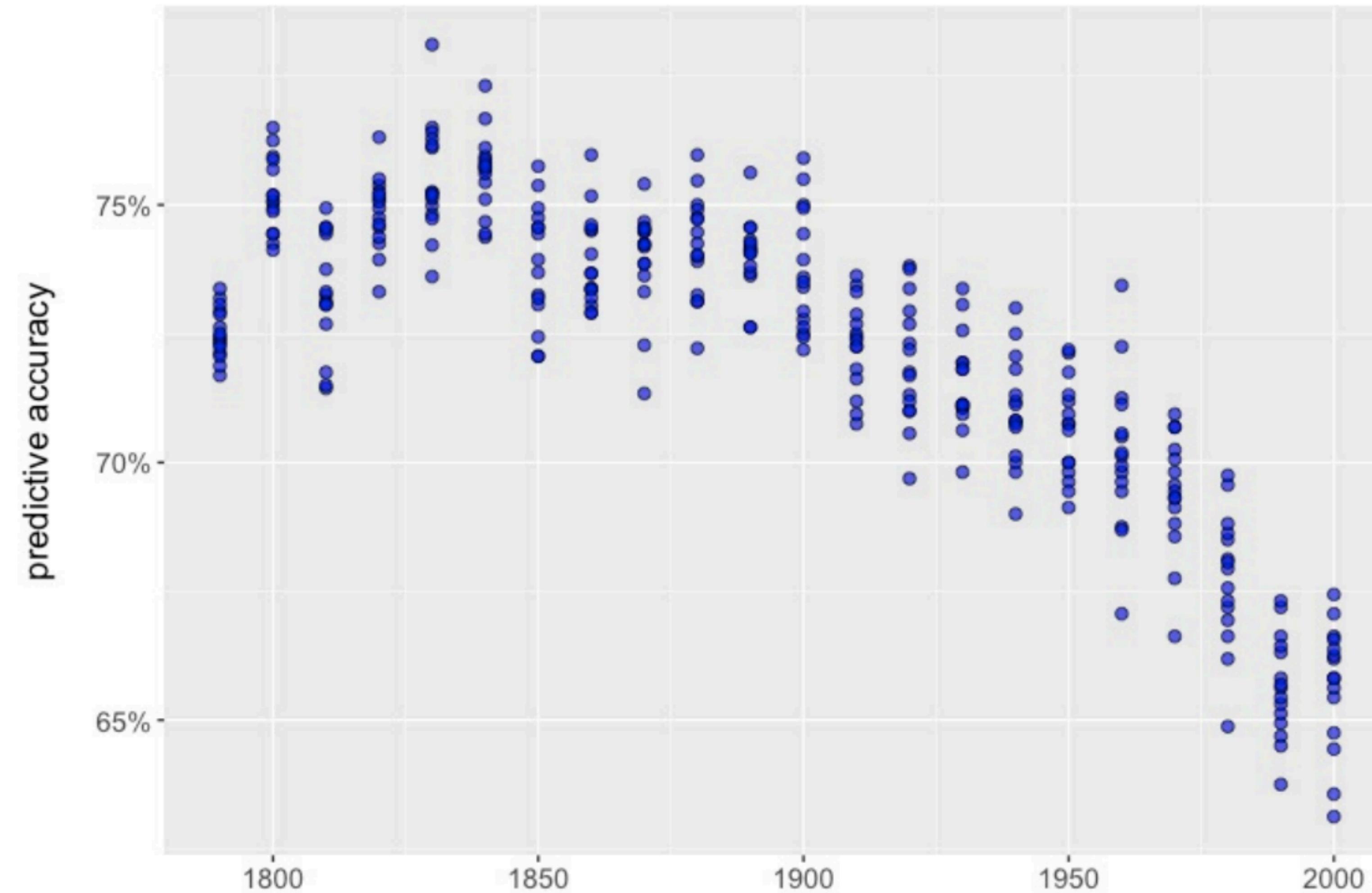
Source: Wikipedia

F1

- We can combine precision and recall into a single metric by taking the harmonic mean of the two quantities.
- $$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

How is classifier used on real world problems?

Accuracy of gender prediction, 1600-character samples



IN CLASS

- Text classification demo