



HOW TO SET UP A CLASSIFIER AND HOW TO EVALUATE THEM?

Sandeep Soni

10/01/2024

QUESTION FOR THE DAY

“How to predict a label for given text?”

AGENDA

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

TEXT CLASSIFICATION

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



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



FORMAL TASK

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



FORMAL TASK

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



CLASSIFICATION

- True h is unknown so find \hat{h}
that's a **closest** approximation



SUPERVISED LEARNING

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



NAIVE BAYES

- One simple yet 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

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:

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

This is tractable!

Now we can rewrite the posterior probability as:

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

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

$$= P(y) \prod_{i=1}^n P(x_i | y)$$

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

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

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

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

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

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

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

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

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

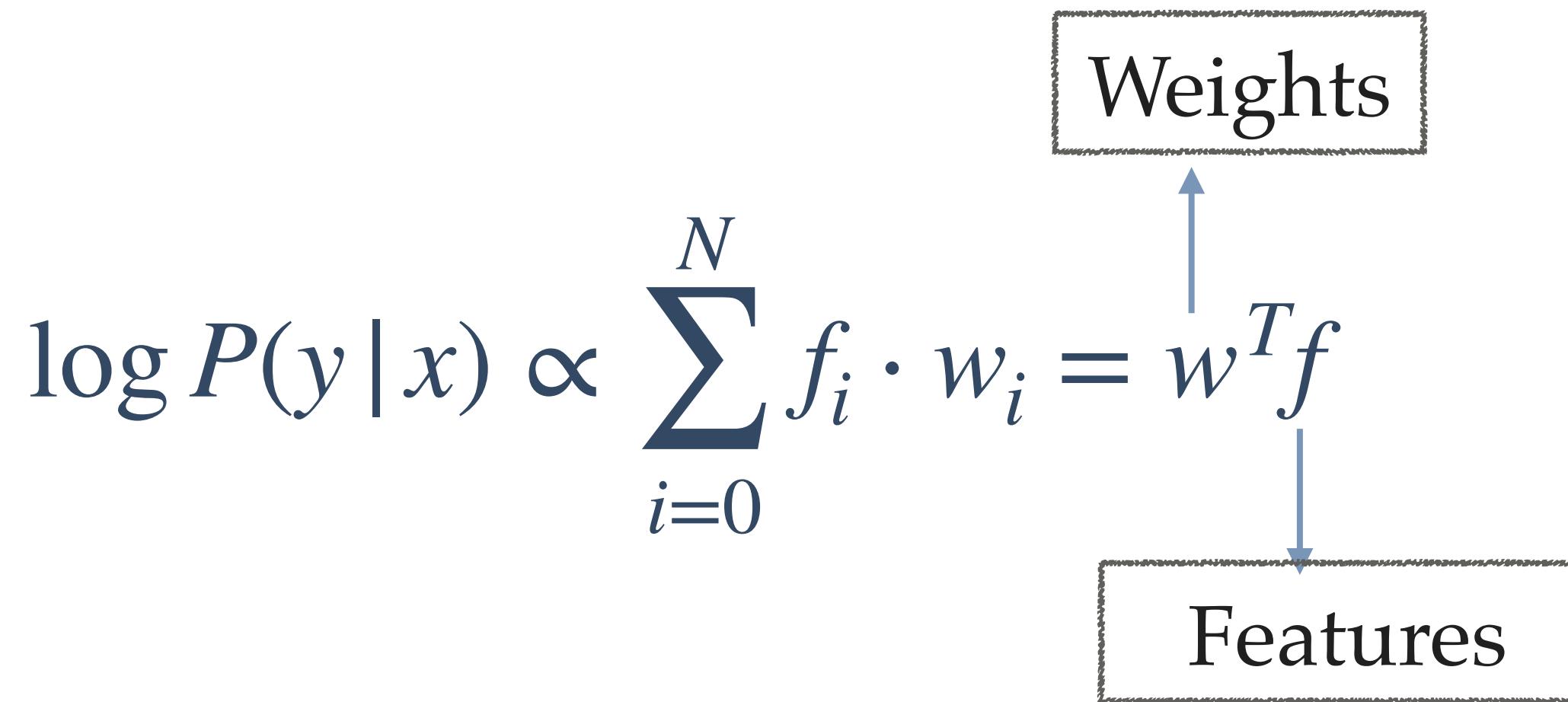
$$\log P(y|x) \propto f_0 \cdot w_0 + \sum_{i=1}^N f_i \cdot w_i$$

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

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

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

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



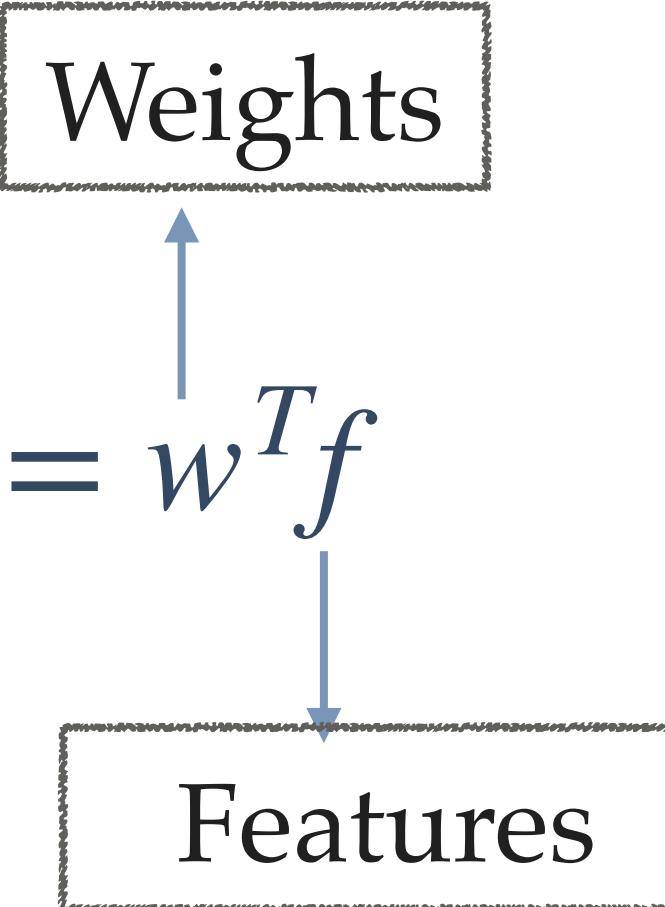
$$\log P(y|x) \propto \log P(y) + \sum_{i=1}^N \log P(x_i|y)$$

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

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

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

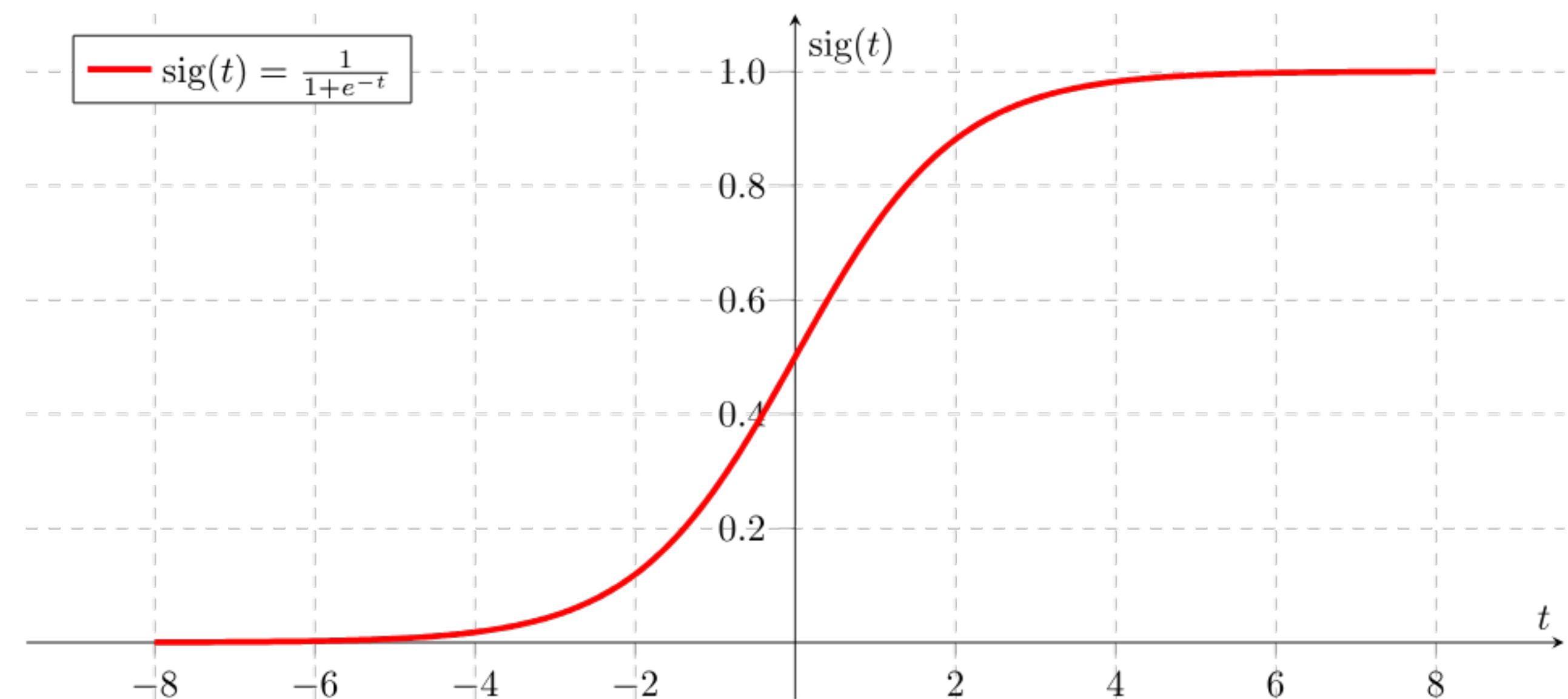
In logistic regression, we directly model the conditional probability of the label given the text

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

LOGISTIC REGRESSION

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

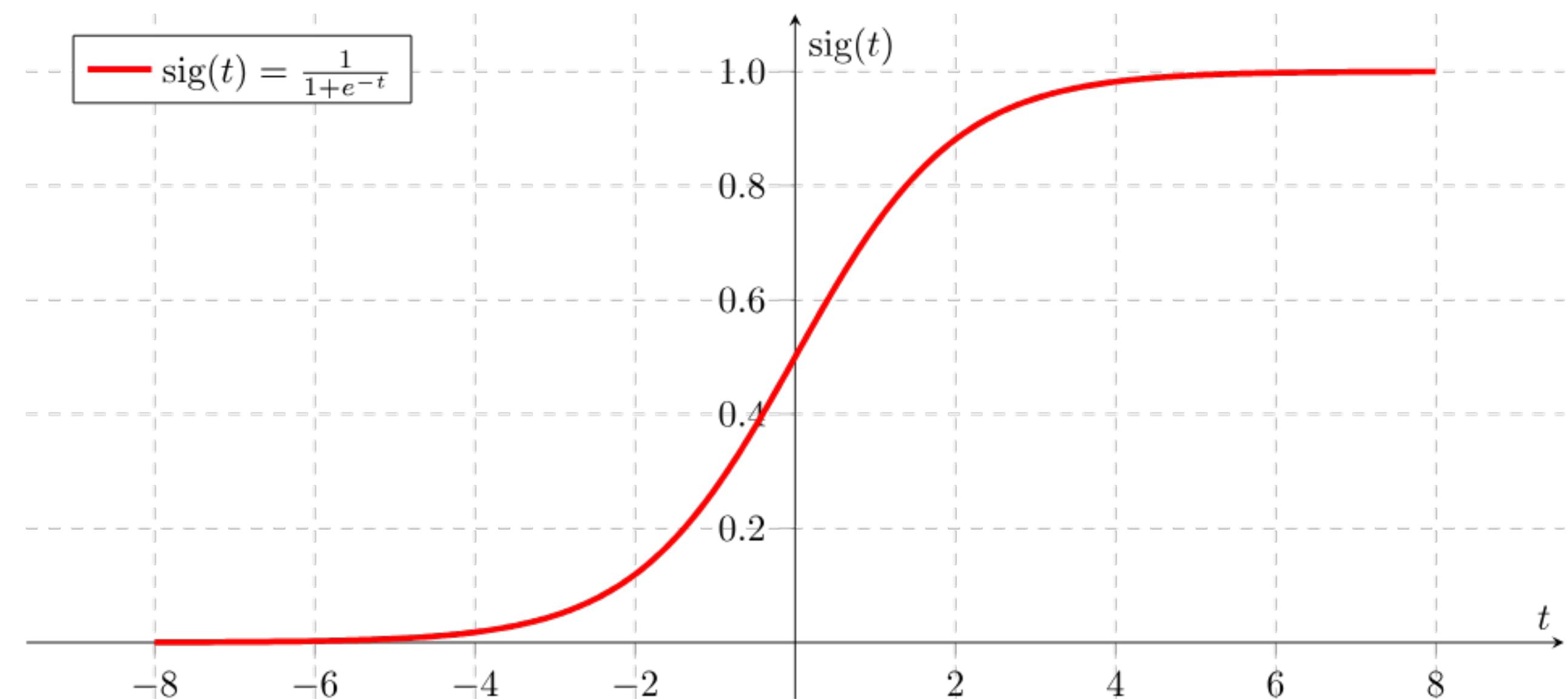
$$P(y|x) = \text{sig}(w^T f(x))$$



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 optimizing some criterion on training data

GENERALIZED CLASSIFICATION

GENERALIZED CLASSIFICATION

- Alternatives to linear models also exist.

GENERALIZED CLASSIFICATION

- Alternatives to linear models also exist.
- $P(y|x) = \text{sig}(g(f(x)))$

GENERALIZED CLASSIFICATION

- Alternatives to linear models also exist.
- $P(y|x) = \text{sig}(g(f(x)))$
- For multi-class classification, $P(y|x) = \text{softmax}(g(f(x)))$

GENERALIZED CLASSIFICATION

- Alternatives to linear models also exist.
- $P(y|x) = \text{sig}(g(f(x)))$ g could be a neural network
- For multi-class classification, $P(y|x) = \text{softmax}(g(f(x)))$

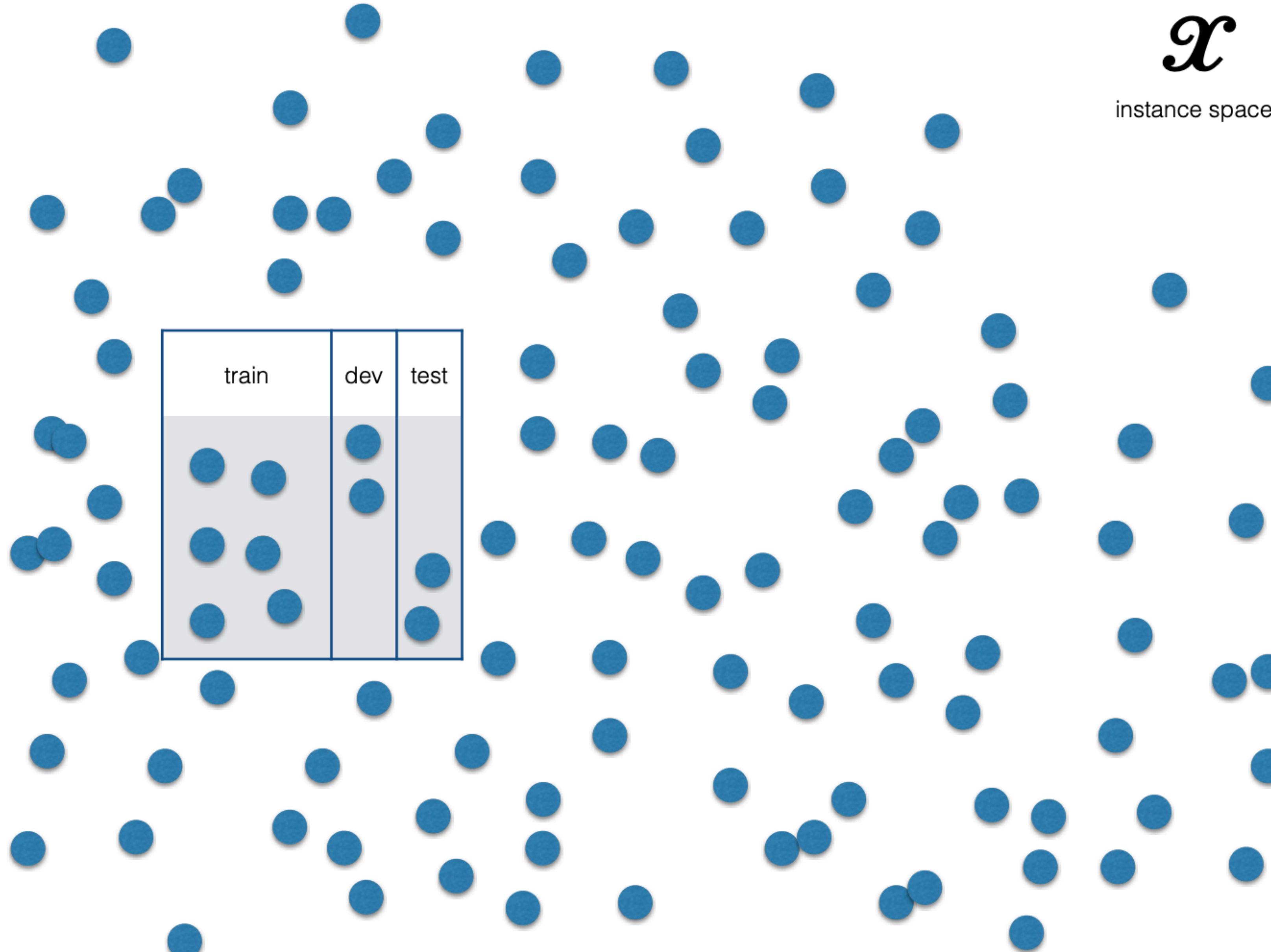
GENERALIZED FORECASTING

GENERALIZED FORECASTING

- Same model can be used to predict continuous values
- $y = g(f(x))$
- 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=spam	y=ham
yhat = spam	True positives	True positives	False positives
	False negatives	False negatives	True negatives
yhat=ham			

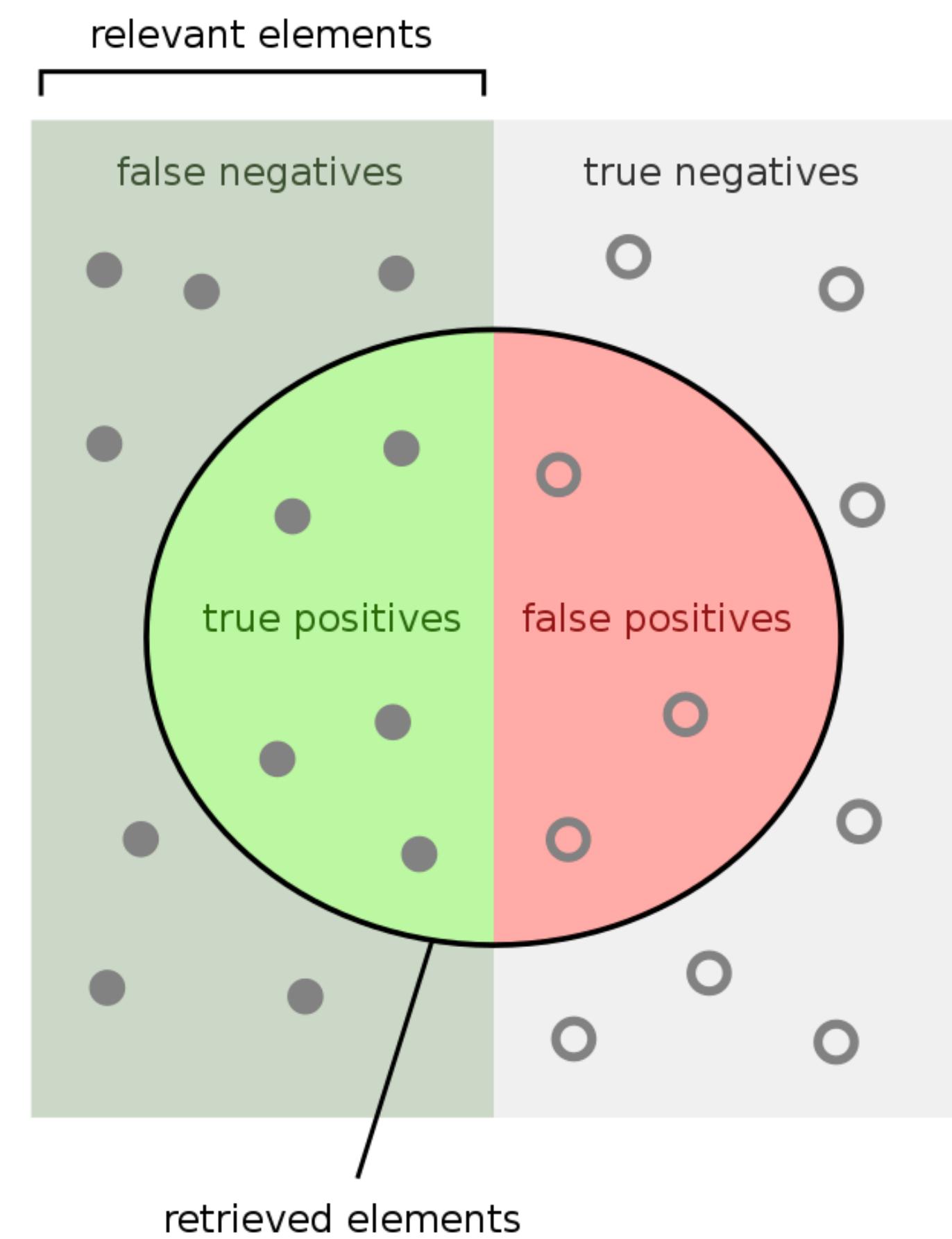
RECALL

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

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.
- $$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

IN CLASS

- Text classification demo