6.034 Introduction to Artificial Intelligence

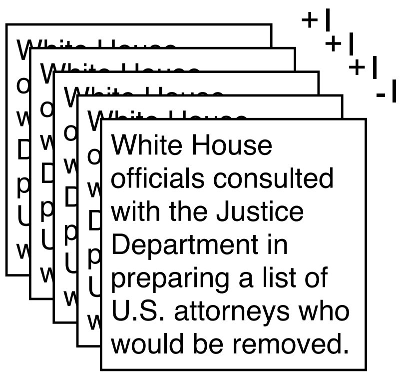
Machine learning and applications

Problems we will cover

- Computational biology
  - cancer classification
  - functional classification of genes
- Information retrieval
  - document classification/ranking
- Recommender systems
  - predicting user preferences (e.g., movies)

Learning with documents

- Binary classification
  - e.g., filtering spam, retrieving similar documents from literature, web,...
- Multi-class classification
  - e.g., topic annotation
- Preference learning
  - e.g., rating reviews
- Learning relations
  - e.g., whether two documents are similar



Documents as feature vectors

- Does the word order matter?

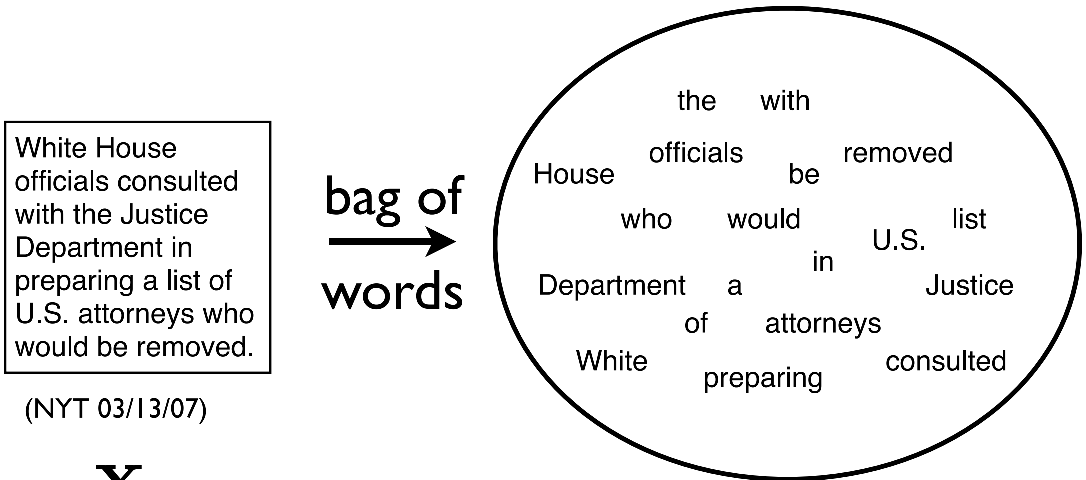
White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.

(NYT 03/13/07)

$\mathbf{x}$

Documents as feature vectors

- Does the word order matter?

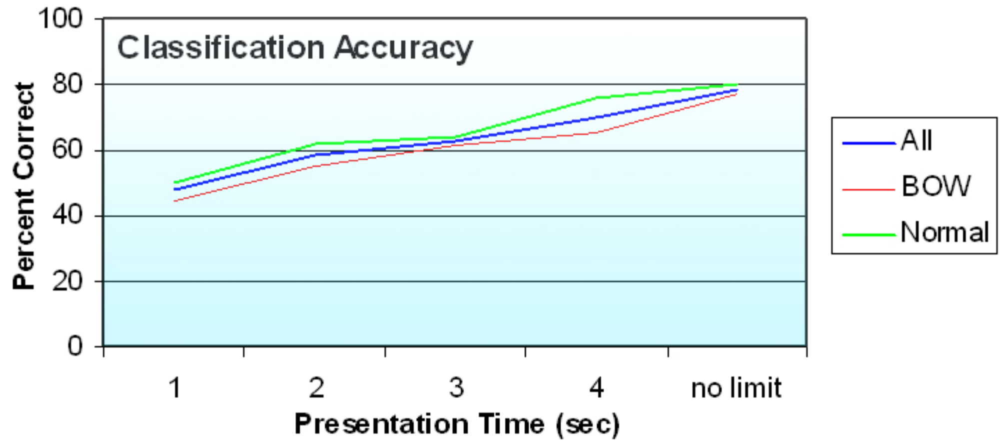


(NYT 03/13/07)

$\mathbf{x}$

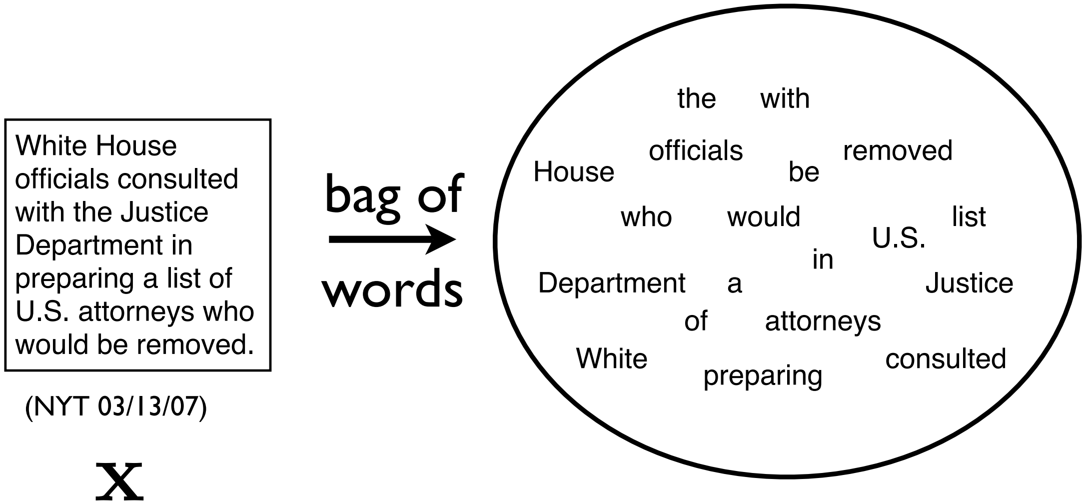
Does the word order matter?

- Doesn't appear to matter to us...



(Wolf et al. 2006)

Documents as feature vectors



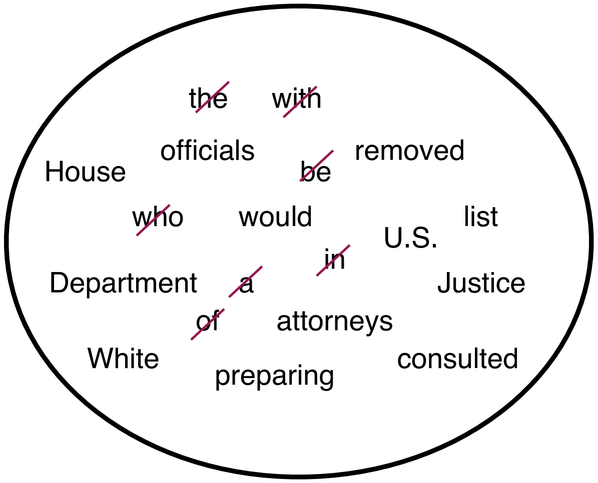
Documents as feature vectors

White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.

(NYT 03/13/07)

$\mathbf{x}$

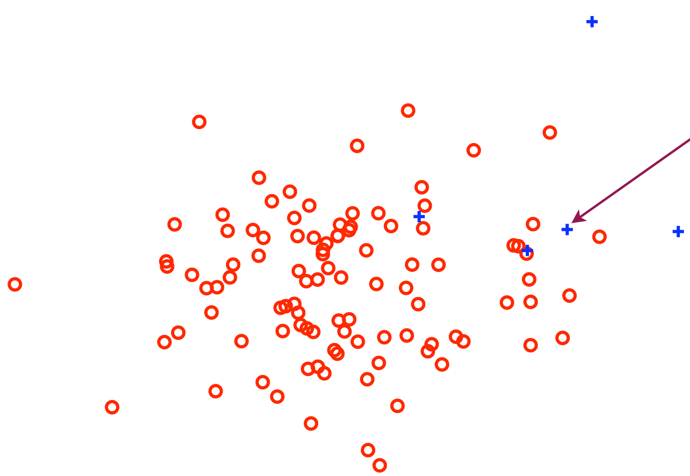
bag of words →



Documents as feature vectors

Document classification

- A few relevant documents (+1)
- A large number of irrelevant documents (-1)



$$\phi(\mathbf{x}) = \begin{array}{cl} \left[\begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \cdots \end{array}\right] & \begin{array}{l} \text{politics} \\ \text{Justice} \\ \text{government} \\ \text{president} \\ \text{House} \\ \cdots \end{array} \end{array}$$

Document classification

- Why is the problem challenging?
  - lots of possible words
  - only a small subset appears in any particular document
  - most frequent words are not content words
  - document classes typically tied to words that occur relatively infrequently
  - any two documents in the same class may have only a few content words in common

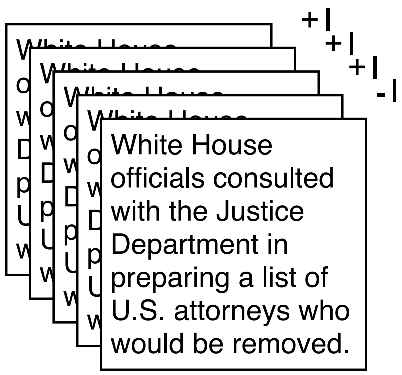
Some tricks

- We can transform the counts in the feature vectors so as to emphasize more "relevant" words
- TFIDF weighting

$$\phi_w(\mathbf{x}) \quad = \quad \overbrace{\left(\begin{array}{c}\text{freq. of word} \\ w \text{ in doc. } \mathbf{x}\end{array}\right)}^{\text{TF}} \cdot \overbrace{\log\left[\frac{\#\text{ of docs}}{\#\text{ of docs with word } w}\right]}^{\text{IDF}}$$

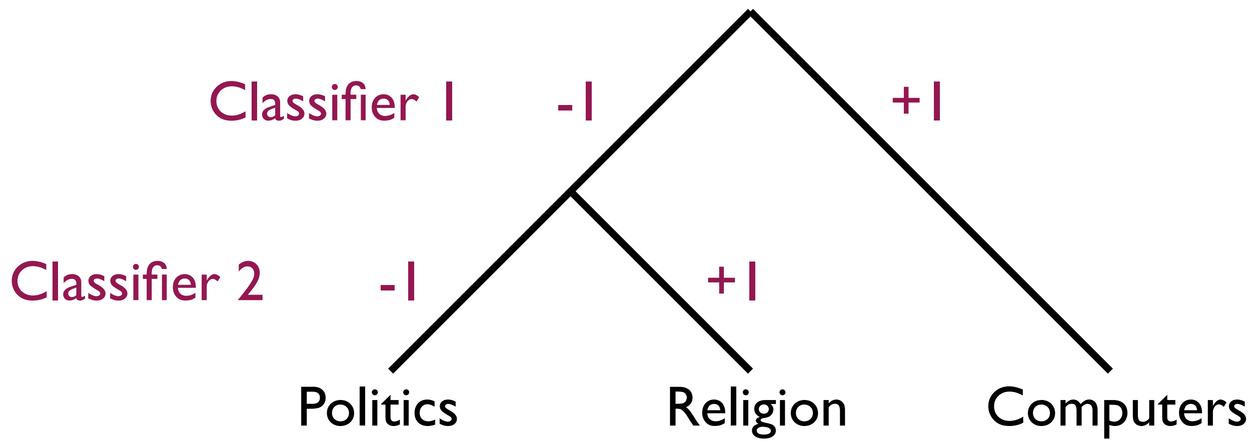
Learning with documents

- Binary classification
  - e.g., filtering spam, retrieving similar documents from literature, web,...
- Multi-class classification
  - e.g., topic annotation
- Preference learning
  - e.g., rating reviews
- Learning relations
  - e.g., whether two documents are similar



Multi-class classification

- We can turn a multi-class classification problem back into a binary problem by using multiple classifiers



- There are many other ways of using binary classifiers to solve multi-class problems (cf. "output codes")

Some multi-class results

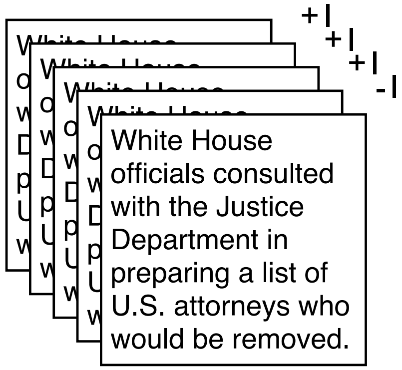
- SVM performs well in comparison to other classifiers on document classification tasks

20 Newsgroups	800		250		100		30	
	SVM	NB	SVM	NB	SVM	NB	SVM	NB
OVA	0.131	0.146	0.167	0.199	0.214	0.277	0.311	0.445
Dense 15	0.142	0.176	0.193	0.222	0.251	0.282	0.366	0.431
BCH 15	0.145	0.169	0.196	0.225	0.262	0.311	0.415	0.520
Dense 31	0.135	0.168	0.180	0.214	0.233	0.276	0.348	0.428
BCH 31	0.131	0.153	0.173	0.198	0.224	0.259	0.333	0.438
Dense 63	0.129	0.154	0.171	0.198	0.222	0.256	0.326	0.407
BCH 63	0.125	0.145	0.164	0.188	0.213	0.245	0.312	0.390

Industry Sector	52		20		10		3	
	SVM	NB	SVM	NB	SVM	NB	SVM	NB
OVA	0.072	0.357	0.176	0.568	0.341	0.725	0.650	0.885
Dense 15	0.119	0.191	0.283	0.363	0.461	0.542	0.738	0.805
BCH 15	0.106	0.182	0.261	0.352	0.438	0.518	0.717	0.771
Dense 31	0.083	0.145	0.216	0.301	0.394	0.482	0.701	0.769
BCH 31	0.076	0.140	0.198	0.292	0.371	0.462	0.676	0.743
Dense 63	0.072	0.135	0.189	0.279	0.363	0.453	0.674	0.745
BCH 63	0.067	0.128	0.176	0.272	0.343	0.443	0.653	0.734

Learning with documents

- Binary classification
  - e.g., filtering spam, retrieving similar documents from literature, web,...
- Multi-class classification
  - e.g., topic annotation
- Preference learning
  - e.g., rating reviews
- Learning relations
  - e.g., whether two documents are similar



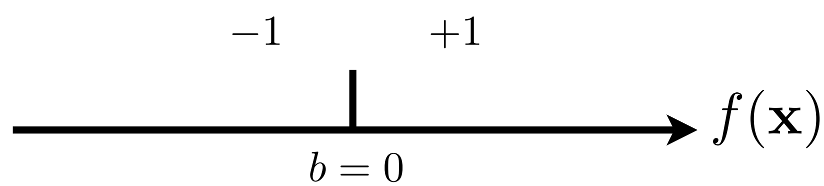
Preference learning

- Predicting ratings on a 1-5 (star) scale is NOT a multi-class classification problem but a problem known as ordinal regression
  - class labels are symbols:
    - it doesn't make sense to compare the "magnitude" of label '1' with label '2' since they are just different symbols associated with the classes
  - 1-5 rating scale is ordinal
    - comparison is relevant: 1 < 2 < 3 < 4 < 5

Preference learning

$$\hat{y} = \operatorname{sign}\left(\overbrace{\sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + w_0}^{\text{discriminant function } f(\mathbf{x})} \right)$$

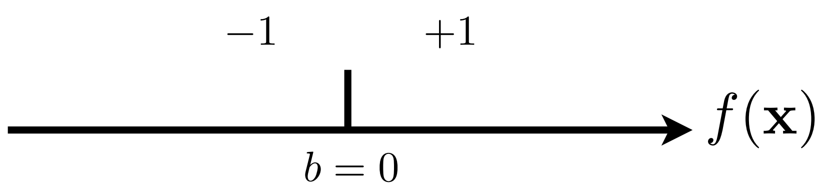
binary classification



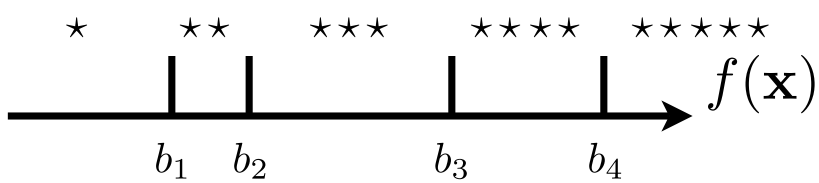
Preference learning

$$\hat{y} = \mathrm{sign}\left(\overbrace{\sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + w_0}^{\text{discriminant function } f(\mathbf{x})} \right)$$

binary classification



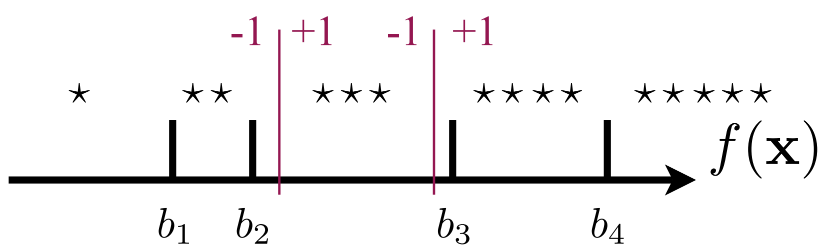
ordinal regression



Preference learning as classification

- We can turn the ordinal regression problem into a classification problem by associating two classification constraints with each response
- For example, if the response for x should be '***', then we would try to enforce

$$+1 \cdot (f(\mathbf{x}) - b_2) \geq 1 \qquad -1 \cdot (f(\mathbf{x}) - b_3) \geq 1$$



- We will have to learn b's in addition to f(x)

Learning similarities

- If we assume that the documents are either similar or not, then we need a classifier over pairs of documents
- The training set would consist of labeled pairs

pair of documents			label
(White House officials consulted with the Justice Department in preparing a list of U.S. attorneys who would be removed.	,	Prime Minister Nuri Kamal al-Maliki met with tribal leaders and the provincial governor of Ramadi.)	-1
$\mathbf{x}_i$		$\mathbf{x}_j$	
	$\dots$		$\dots$

- Most pairs can be assumed to be dissimilar (cf. imbalanced classes)

A classifier over pairs of examples

- Can we still use a linear classifier? If yes, we need a feature vector
- We could simply use a feature vector for each pair by concatenating the individual feature vectors

$$\tilde{\phi}(\mathbf{x}_i, \mathbf{x}_j) = \begin{bmatrix} \phi(\mathbf{x}_i) \\ \phi(\mathbf{x}_j) \end{bmatrix}$$

  - this is not the best way to solve the problem

Problems we will cover

- Computational biology
  - cancer classification
  - functional classification of genes
- Information retrieval
  - document classification/ranking
- Recommender systems
  - predicting user preferences (e.g., movies)

Recommender problems

- Lot's of companies rely on predicting user preferences

movies

users

	2		1			4				5	
	5		4						1		3
		3		5			2				
4						5		3			
		4		1	3				5		
			2				1				4
	1					5		5		4	
		2			5				4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3				5	
2			1		1						
		5			2			4		4	
	1		3		1	5		4		5	
1		2			4				5		

- The machine learning task here is to fill the entries of the rating matrix based on only a few ratings from each user

Recommender problems: movies

- The dimensions in practice
  - 400,000 users
  - 17,000 movies
  - 1% known ratings

movies

users

	2		1			4				5	
	5		4						1		3
		3		5			2				
4						5		3			
		4		1	3				5		
			2				1				4
	1					5		5		4	
		2			5				4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3				5	
2			1		1						
		5			2			4		4	
	1		3		1	5		4		5	
1		2			4				5		

Recommender problems: movies

- Users rate movies very differently
  - very positive, very negative, etc.
- Missing ratings are ambiguous
  - movies that users have not yet seen
  - movies they saw but didn't rate
  - movies they know they don't want to see
  - movies they don't realize they would like to see

movies

users

	2		1			4				5	
	5		4						1		3
		3		5			2				
4						5		3			
		4		1	3				5		
			2				1				4
	1					5		5		4	
		2			5				4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3				5	
2			1		1						
		5			2			4		4	
	1		3		1	5		4		5	
1		2			4				5		

Where's the information?

- The problem is actually a bit easier than what it seems
  - many users are similar, many movies are similar

movies

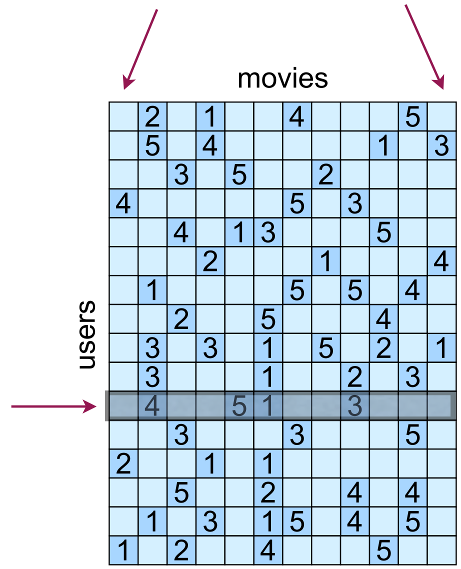
users

	2		1			4				5	
	5		4						1		3
		3		5			2				
4						5		3			
		4		1	3				5		
			2				1				4
	1					5		5		4	
		2			5				4		
	3		3		1		5		2		1
	3				1			2		3	
	4			5	1			3			
		3				3				5	
2			1		1						
		5			2			4		4	
	1		3		1	5		4		5	
1		2			4				5		

How do we solve this with SVMs?

- Suppose we were able to obtain a feature vector for each movie

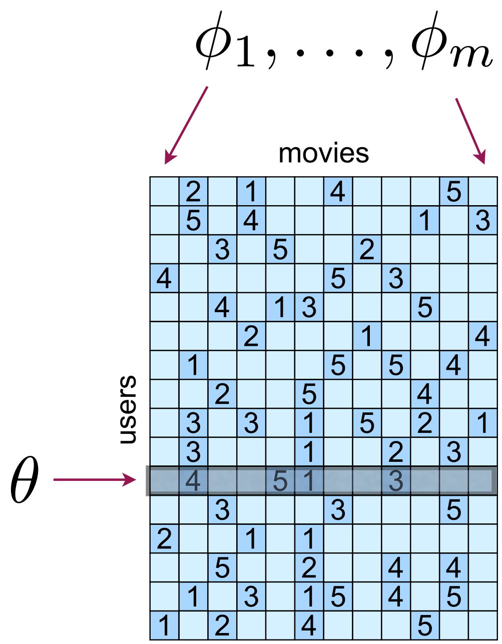
$$\phi_1, \ldots, \phi_m$$

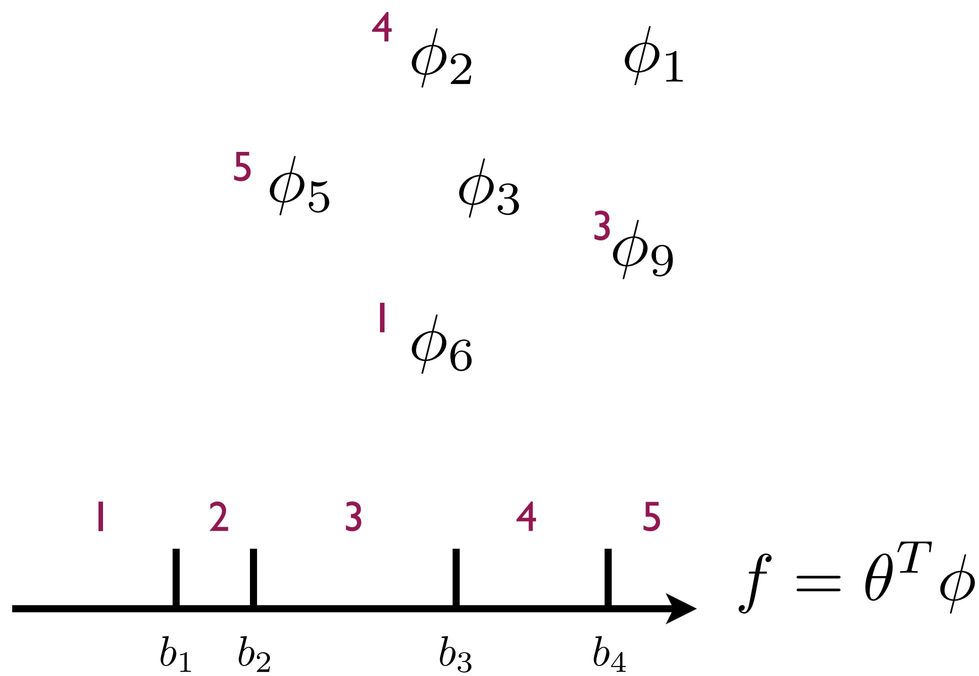


- We could then try to solve the problem separately for each user

Ordinal regression: users

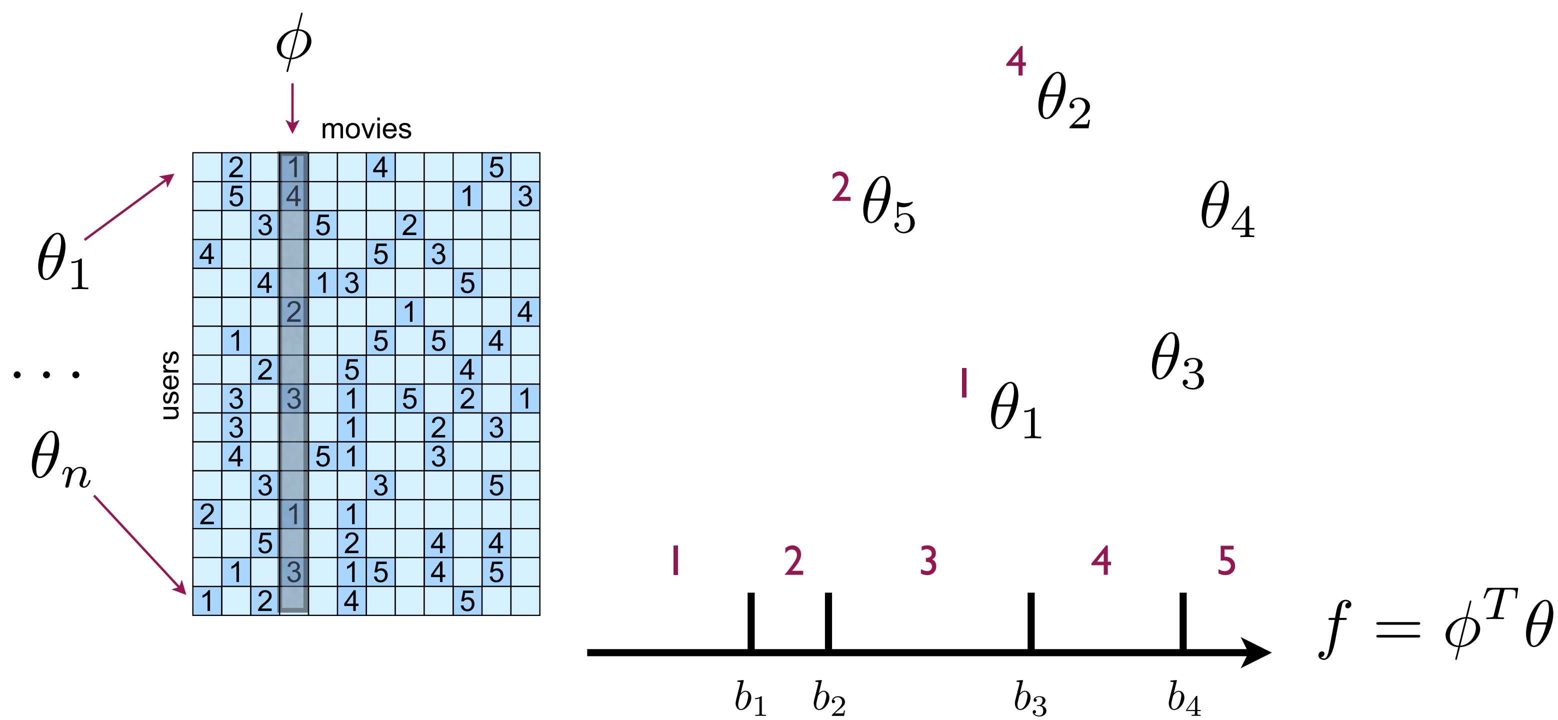
- The problem for predicting the ratings of each user becomes an ordinal regression problem exactly of the type we have already discussed





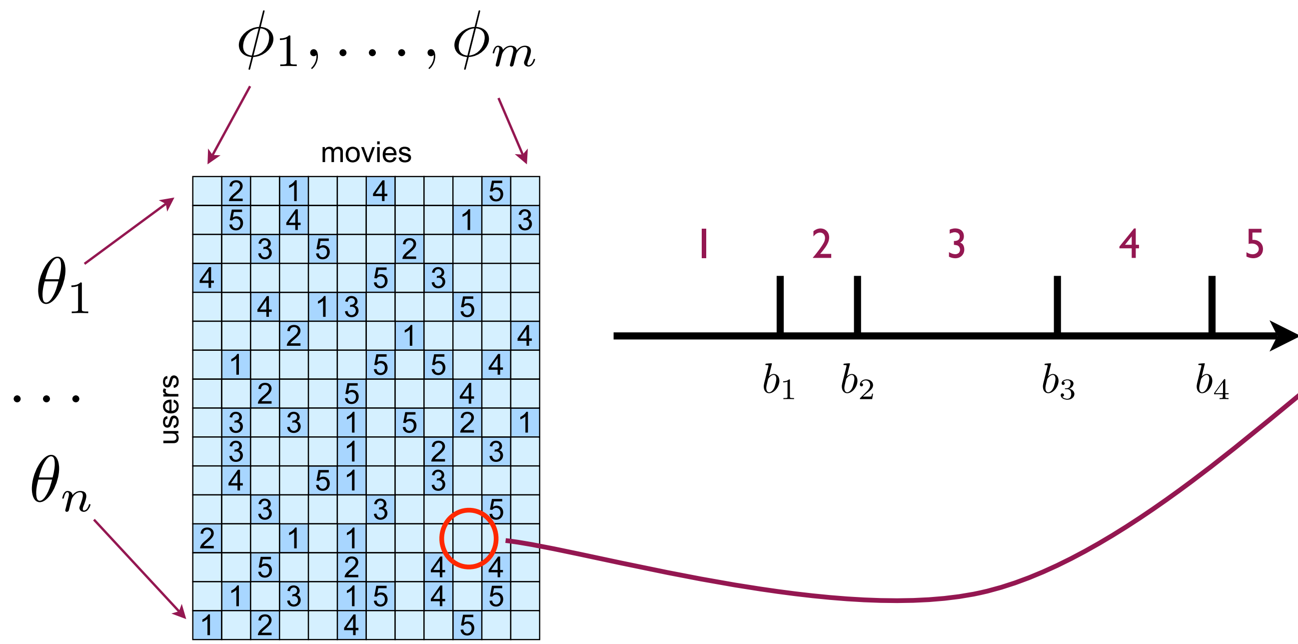
Ordinal regression: movies

- Suppose we have obtained feature vectors for each user
- The problem is again ordinal regression



Solution

- We can iteratively solve feature vectors for users and movies



$$f_{ij} = \theta_i^T \phi_j$$

Some results

- The SVM based collaborative filtering method works very well in comparison to other methods

Algorithm	MovieLens Weak NMAE	MovieLens Strong NMAE
URP	$.4341 \pm .0023$	$.4444 \pm .0032$
Attitude	$.4320 \pm .0055$	$.4375 \pm .0028$
MMMF	$\mathbf{.4156} \pm .0037$	$\mathbf{.4203} \pm .0138$

(Rennie et al. 2005)

Problems we will cover

- Computational biology
  - cancer classification
  - functional classification of genes
- Information retrieval
  - document classification/ranking
- Recommender systems
  - predicting user preferences (e.g., movies)

Lot's of other problems can also be solved with SVMs...