Chapitre : Recherche d'information et apprentissage

Slides empruntés

De la présentation Tie-Yan Liu

Microsoft Research Asia

Conventional Ranking Models

- Query-dependent
  - Boolean model, extended Boolean model, etc.
  - Vector space model, latent semantic indexing (LSI), etc.
  - BM25 model, statistical language model, etc.
- Query-independent
  - PageRank, TrustRank, BrowseRank, Toolbar Clicks, etc.

Generative vs. Discriminative

- All of the probabilistic retrieval models (PRP, LM, Inference model presented so far fall into the category of *generative models*
  - A generative model assumes that documents were generated from some underlying model (in this case, usually a multinomial distribution) and uses training data to estimate the parameters of the model
  - probability of belonging to a class (i.e. the relevant documents for a query) is then estimated using Bayes' Rule and the document model

Discriminative model for IR

- Discriminative models can be trained using
  - explicit relevance judgments
  - or click data in query logs

- Click data is much cheaper, more noisy

Relevance judgement

- Degree of relevance l_k
  - Binary: relevant vs. irrelevant
  - Multiple ordered categories: Perfect > Excellent > Good > Fair > Bad
- Pairwise preference $l_{u,v}$
  - Document *A* is more relevant than document *B*
- Total order π_l
  - Documents are ranked as {A,B,C,..} according to their relevance

Apprentissage de l'ordonnacement : Learning to rank

Machine learning can help

- Machine learning is an effective tool
  - To automatically tune parameters.
  - To combine multiple evidences.
  - To avoid over-fitting (by means of regularization, etc.)
- "Learning to Rank"
  - In general, those methods that use machine learning technologies to solve the problem of ranking can be named as "learning to rank" methods.

Machine learning

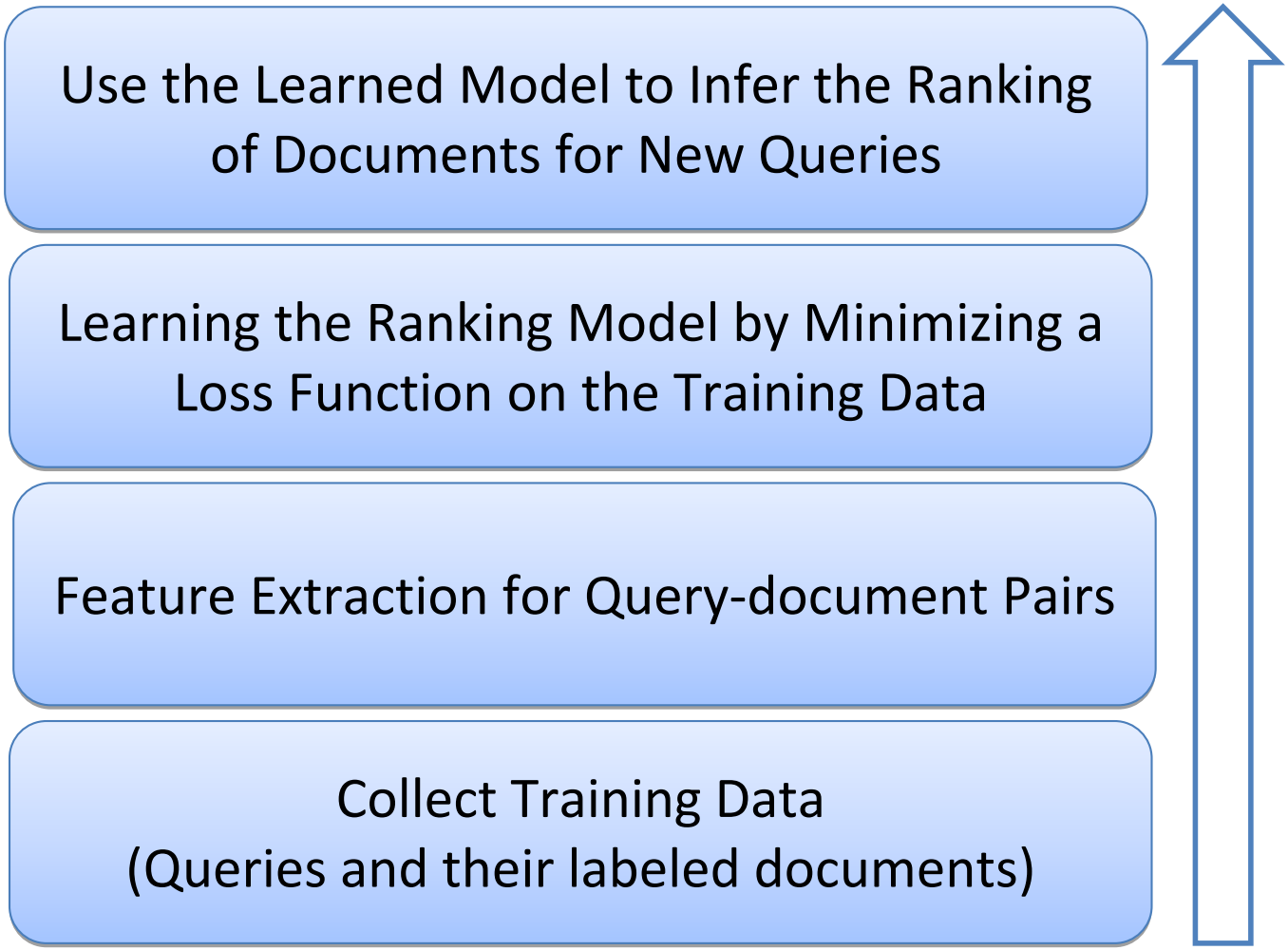
- Given a training set of examples, each of which is a tuple of: a query q, a document d, a relevance judgment for d on q

- Learn weights from this training set, so that the learned scores approximate the relevance judgments in the training set

Discriminative Training

- An automatic learning process based on the training data
- With the four pillars of discriminative learning
  - Input space, (features vectors)
  - Output space (+1/-1; real value, ranking)
  - Hypothesis space (function mapping the input to the output)
  - Function quality (Loss function: risk, error between the hypothesis and the ground truth)

Learning to rank: general approach



Example of features

Table 2.3: Example Features of Learning to Rank for Web Search

Feature	Type	Explanation
Number of occurrences	Matching	number of times query exactly occurs in title, anchor, URL, extracted title, associated query, and body
BM25	Matching	BM25 scores on title, anchor, URL, extracted title, associated query, and body
N-gram BM25	Matching	BM25 scores of n-grams on title, anchor, URL, extracted title, associated query, and body
Edit Distance	Matching	edit distance scores between query and title, anchor, URL, extracted title, associated query, and span in body (minimum length of text segment including all query words [94])
Number of in-links	Document	number of in-links to the page
PageRank	Document	importance score of page calculated on web link graph
Number of clicks	Document	number of clicks on the page in search log
BrowseRank	Document	importance score of page calculated on user browsing graph
Spam score	Document	likelihood of spam page
Page quality score	Document	likelihood of low quality page

Categorization: Basic Unit of Learning

- Pointwise
  - Input: single document
  - Output: scores or class labels (relevant/non relevant)
- Pairwise
  - Input: document pairs
  - Output: partial order preference
- Listwise
  - Input: document collections
  - Output: ranked document List

Catergoriztion of the algorithms

Category	Algorithms
Pointwise Approach	**Regression**: Least Square Retrieval Function (TOIS 1989), Regression Tree for Ordinal Class Prediction (Fundamenta Informaticae, 2000), Subset Ranking using Regression (COLT 2006), … **Classification**: Discriminative model for IR (SIGIR 2004), McRank (NIPS 2007), … **Ordinal regression**: Pranking (NIPS 2002), OAP-BPM (EMCL 2003), Ranking with Large Margin Principles (NIPS 2002), Constraint Ordinal Regression (ICML 2005), …
Pairwise Approach	Learning to Retrieve Information (SCC 1995), Learning to Order Things (NIPS 1998), Ranking SVM (ICANN 1999), RankBoost (JMLR 2003), LDM (SIGIR 2005), RankNet (ICML 2005), Frank (SIGIR 2007), MHR(SIGIR 2007), GBRank (SIGIR 2007), QBRank (NIPS 2007), MPRank (ICML 2007), IRSVM (SIGIR 2006), LambdaRank (NIPS 2006),…
Listwise Approach	**Non-measure specific**: ListNet (ICML 2007), ListMLE (ICML 2008), BoltzRank (ICML 2009) … **Measure-specific**: AdaRank (SIGIR 2007), SVM-MAP (SIGIR 2007), SoftRank (LR4IR 2007), RankGP (LR4IR 2007), …

The Pointwise approach

	The Pointwise Approach		
	Regression	Classification	Ordinal Regression
Input Space	Single documents y_j		
Output Space	Real values	Non-ordered Categories	Ordinal categories
Hypothesis Space	Scoring function $f(x)$		
Loss Function	Regression loss	Classification loss	Ordinal regression loss
	$L(f; x_j, y_j)$		

The Pointwise approach

- Reduce ranking to
  - Regression
    - Subset Ranking
  - Classification
    - Discriminative model for IR
    - MCRank
  - Ordinal regression
    - PRanking
    - Ranking with large margin principle

$$q \leftrightarrow \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \downarrow$$

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$$

Exemple pointwise

- Collecter des exemples d'entraînement (q, d, y) triplets
  - Pertinence r est binaire (peut être graduée)
  - Document représenté par deux « *features* »
    - Le vecteur x=(α, ω), représenté par deux caractéristiques α est la similarité (entre q et d) , ω est la proximité entre les termes de la requête dans le document
      - ω est la taille de la partie du texte du document qui inclut tous les mots de la requête
- Deux exemples d'approches :
  - Régression linéaire
  - Classification

Pointwise approach: linear regression

- La pertinence est vue comme une valeur de score
- But apprendre la fonction de score qui combine les différentes caractéristiques

$$f(x) = \sum_{i=1}^{m} w_i x_i + w_0$$

  - w les poids ajustés par apprentissage
  - (x_1, ..x_m) les caractéristiques du document-requête

- Trouver les w_i qui réduisent l'erreur suivante :

$$L(f; x_i, y_i) = |f(x) - y_i|^2 \quad L(f, x, y) \rightarrow \frac{1}{2n} \sum_{i=1} (y_i - f(x_i))^2$$

  - → pertinence (y=1), non pertinence (y=0)

Exemple Régression

- Apprendre une fonction de score qui combine les deux « features » $(x_1,x_2)= (\alpha, \omega)$

$$f(d, q) = w_1 * \alpha(d, q) + w_2 * \omega(d, q) + w_0$$

example	docID	query	cosine score	ω	judgment
Φ_1	37	linux operating system	0.032	3	*relevant*
Φ_2	37	penguin logo	0.02	4	*nonrelevant*
Φ_3	238	operating system	0.043	2	*relevant*
Φ_4	238	runtime environment	0.004	2	*nonrelevant*
Φ_5	1741	kernel layer	0.022	3	*relevant*
Φ_6	2094	device driver	0.03	2	*relevant*
Φ_7	3191	device driver	0.027	5	*nonrelevant*

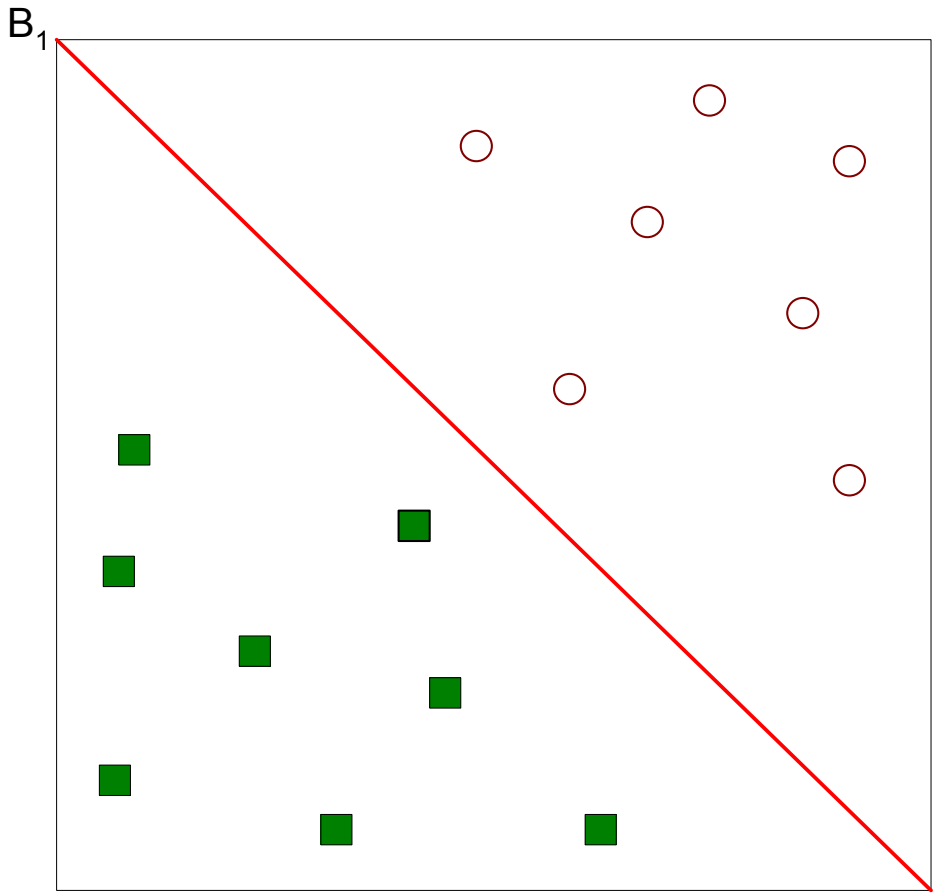
Pointwise approach: Classification (SVM)

- Ramène la RI à un problème de classification:
  - Une requête, un document, une classe (Pertinent, non pertinent) (plusieurs catégories)

- On cherche une fonction de décision de la forme :
  - f(x)=sign <(x.w)+b>

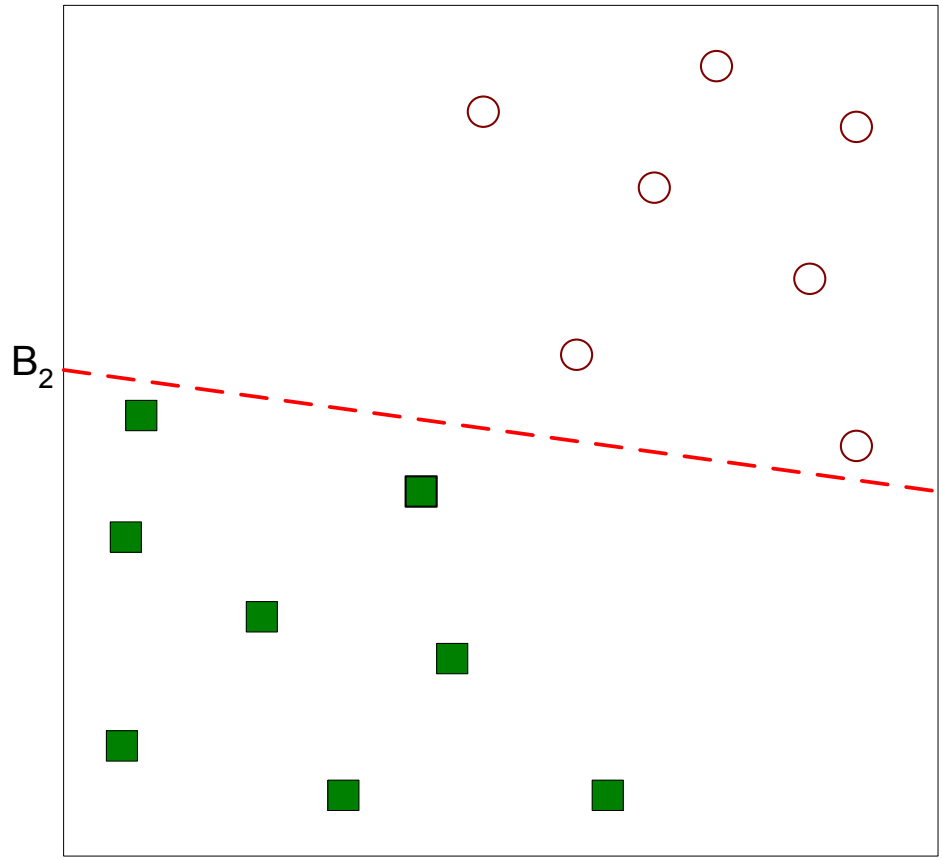
  - On souhaite $f(x) \leq -1$ pour non pertinent et $f(x) \geq 1$ pour pertinent

Support Vector Machines



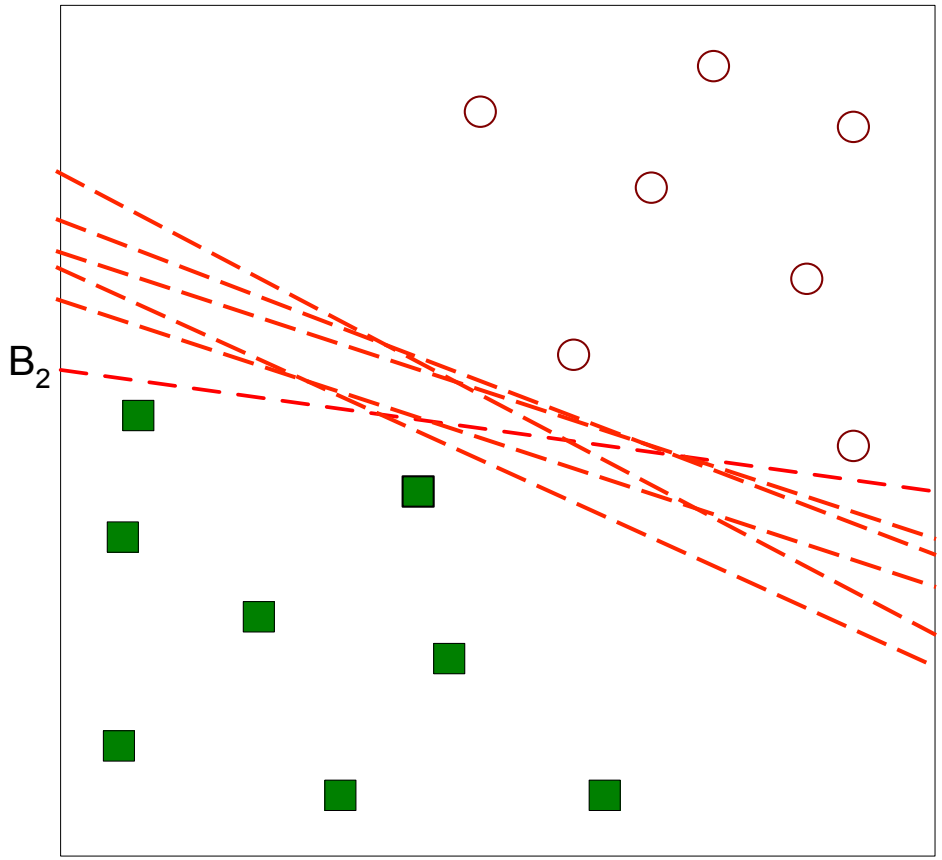
- Find a linear hyperplane (decision boundary) that will separate the data
- One Possible Solution

Support Vector Machines



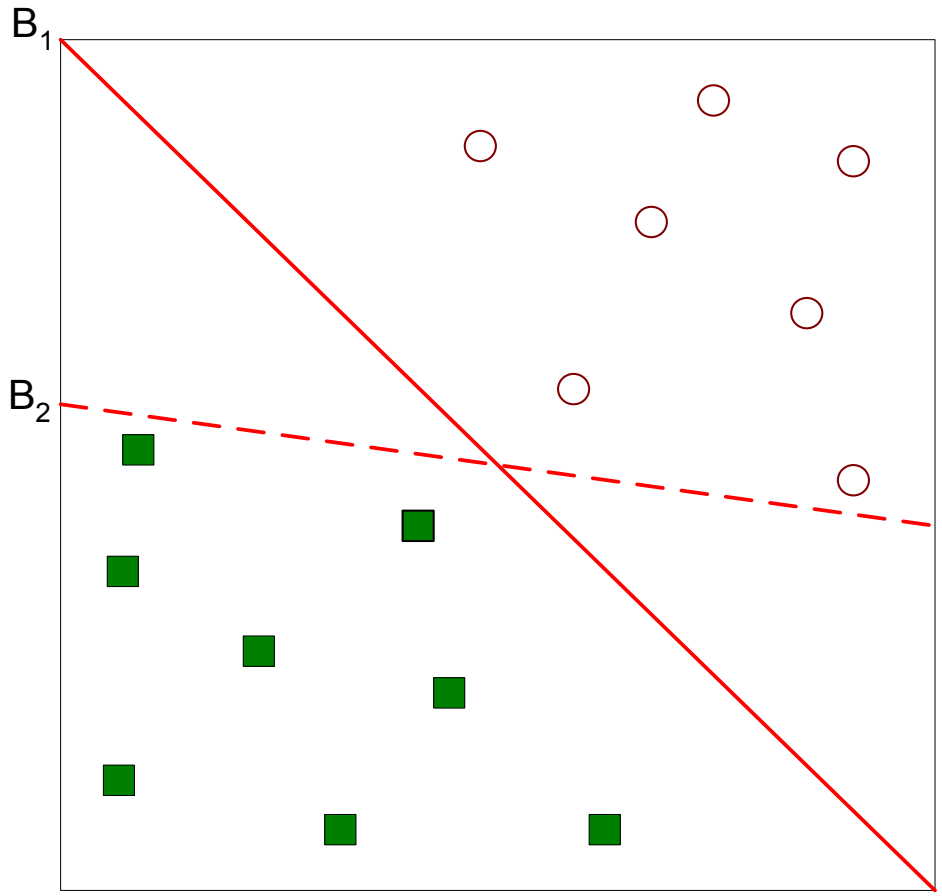
- Another possible solution

Support Vector Machines



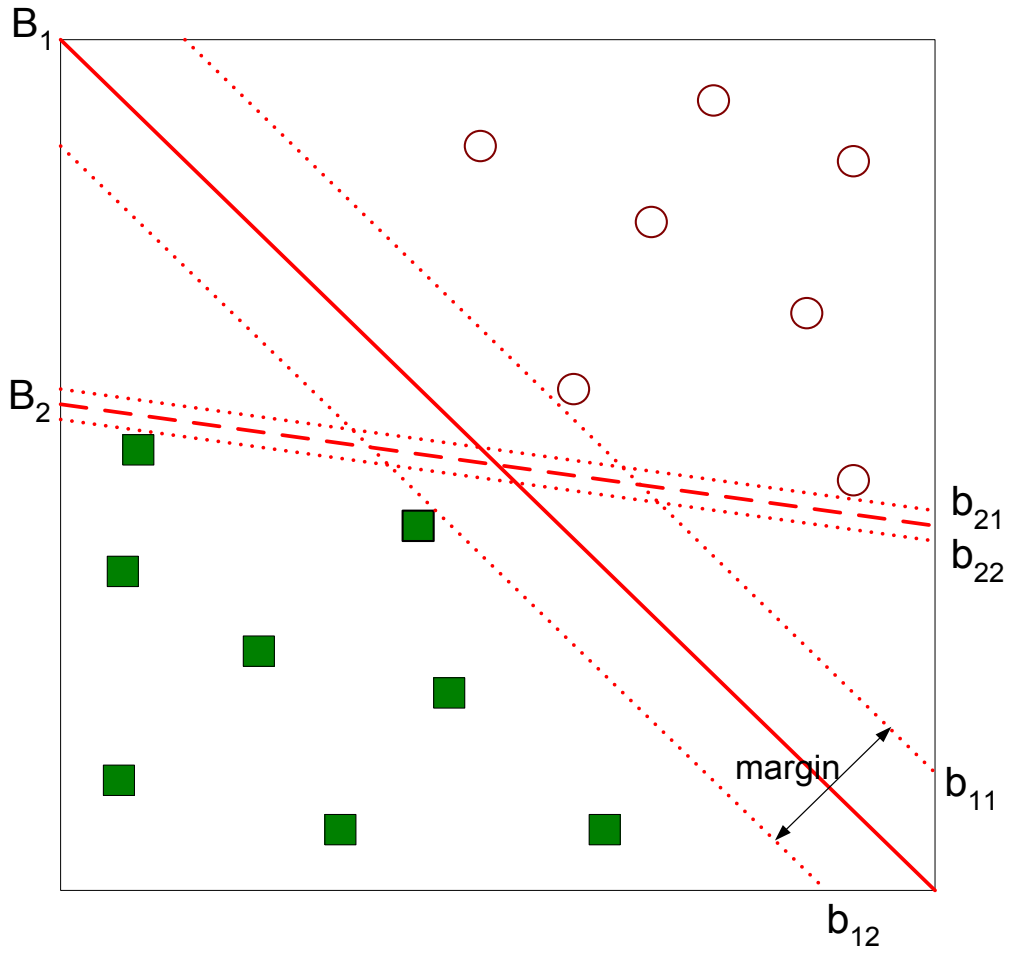
- Other possible solutions

Support Vector Machines



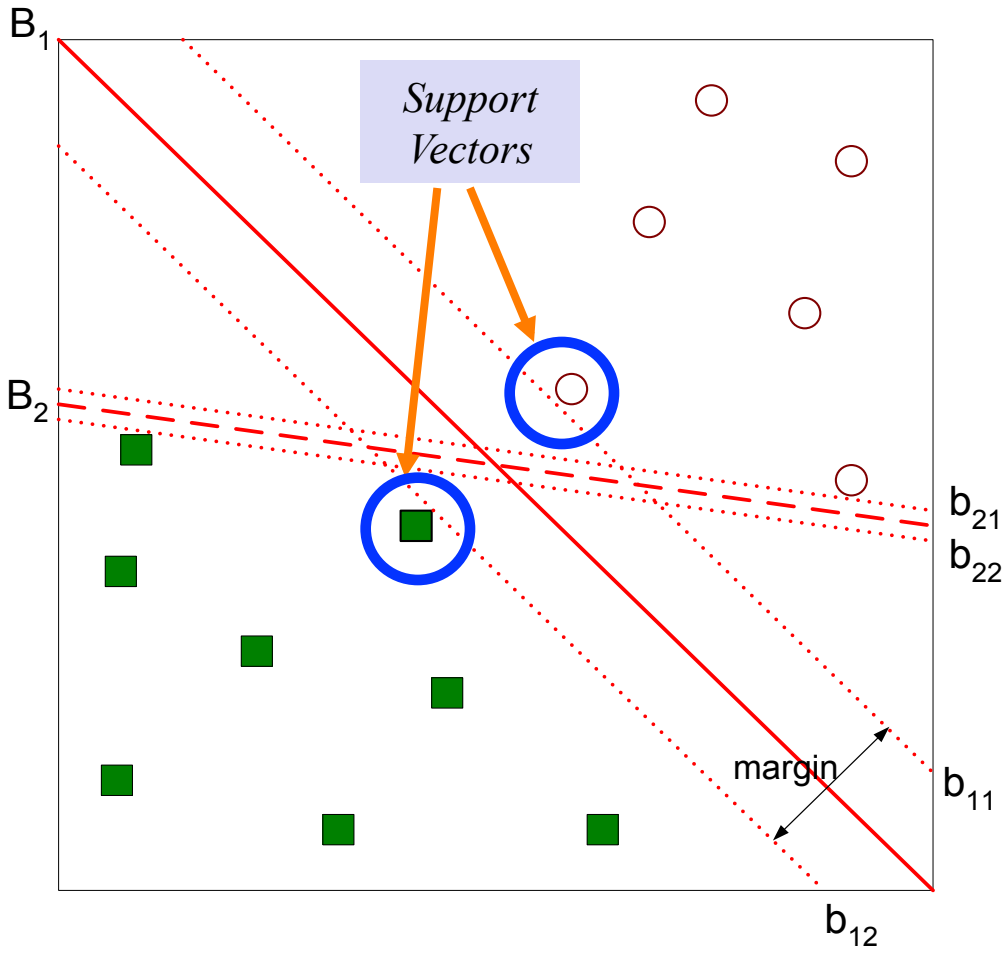
- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines

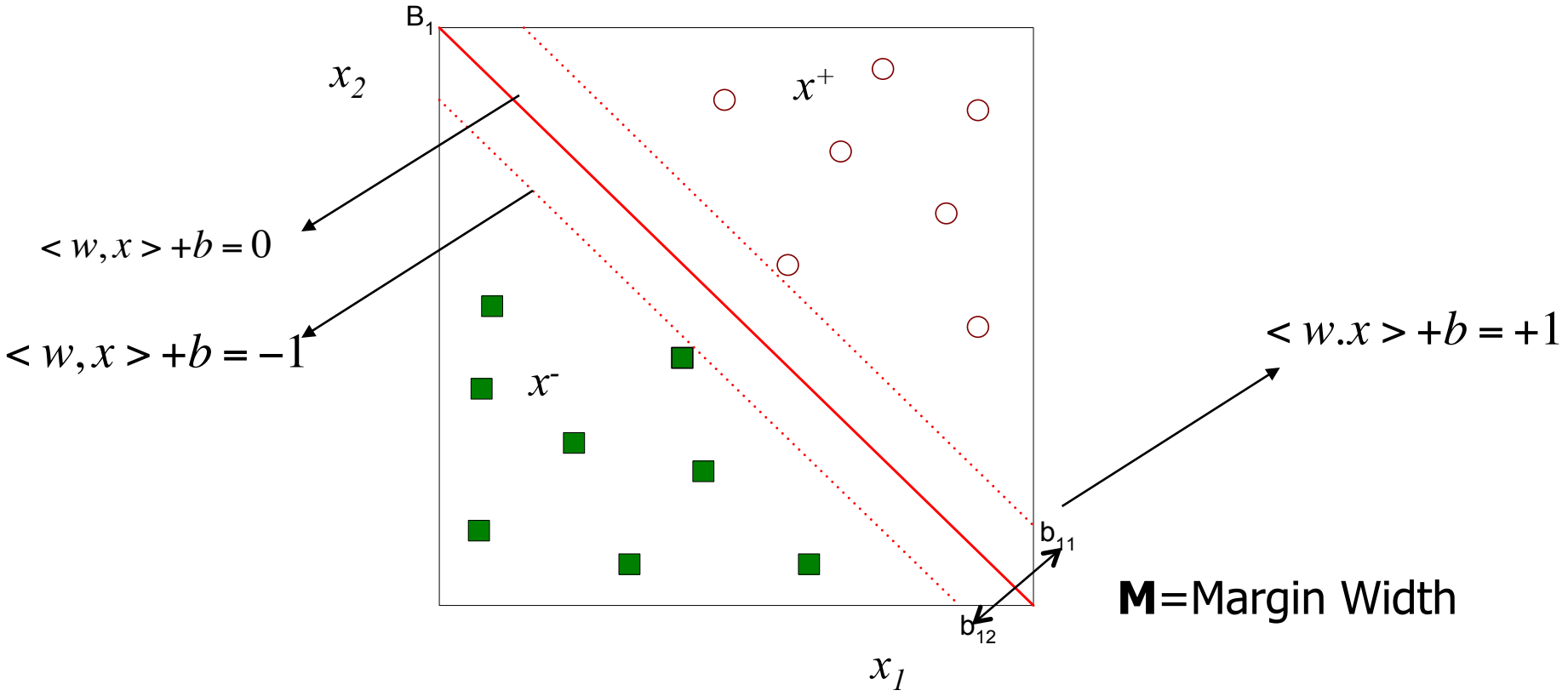


- Find hyperplane maximizes the margin => B1 is better than B2

Support Vector Machines



Support Vector Machines



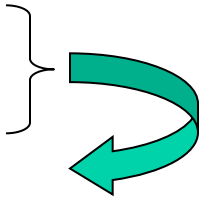
$$f(\vec{x}) = \begin{cases} 1 & \text{if } <w,x> + b \geq 1 \\ -1 & \text{if } <w,x> + b \leq -1 \end{cases}$$

$$M = \frac{(x^+ - x^-) \cdot w}{|w|} = \frac{2}{|w|}$$

Linear SVM

- Goal: **1) Correctly classify all training data**

$$w.x_i + b \geq 1 \quad \text{if } y_i = +1$$
$$wx_i + b \leq 1 \quad \text{if } y_i = -1$$
$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$



2) Maximize the Margin

$$M = \frac{2}{|w|}$$

same as minimize

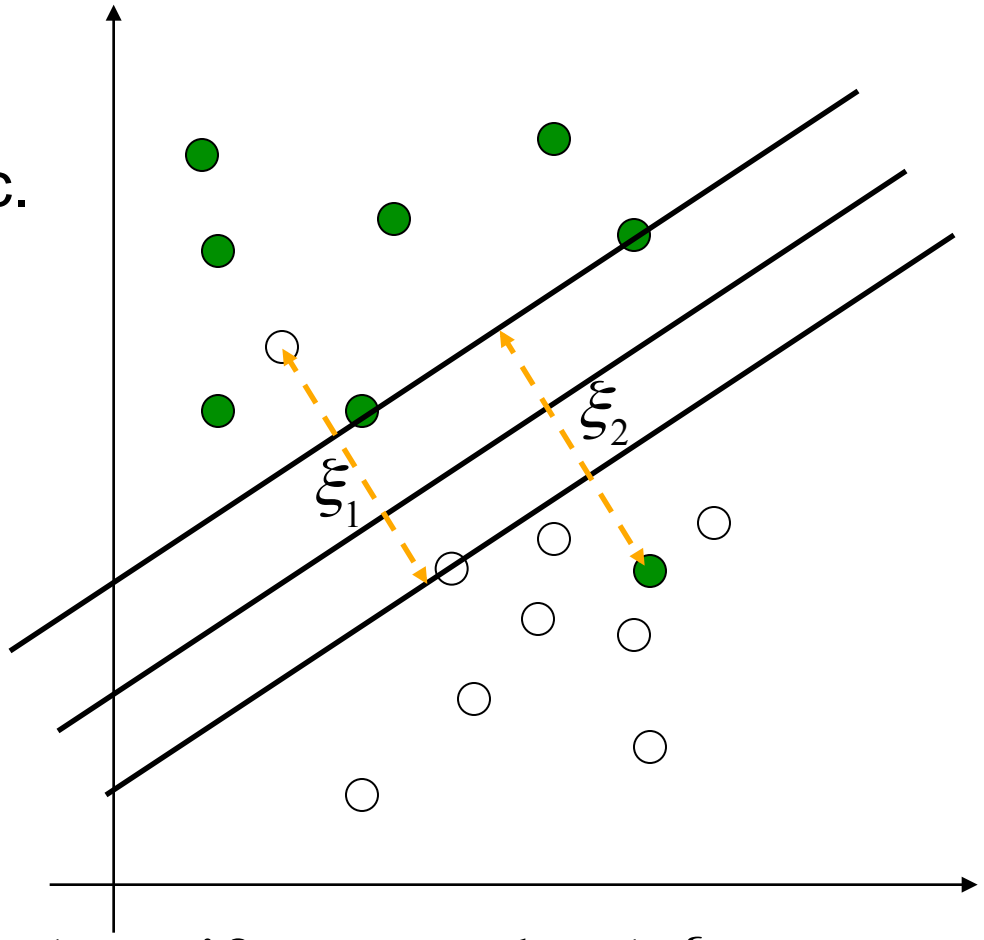
$$\frac{1}{2} w^t w$$

- **We can formulate a Quadratic Optimization Problem and solve for w and b**
- Minimize $\frac{1}{2} w^t w$

subject to $y_i (wx_i + b) \geq 1 \quad \forall i$

Linear SVM(if no separable)

- Noisy data, outliers, etc.
- Slack variables ξ_i



$$f(x) = \begin{cases} 1 & \text{if } <w,x> + b \geq 1-\xi_i \\ -1 & \text{if } <w,x> + b \leq -1+\xi_i \end{cases}$$

SVM : Hard Margin v.s. Soft Margin

- **The old formulation:**

Find $\mathbf{w}$ *and* b *such that*
Minimize $\frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ *and for all* $\{(\mathbf{x}_i, y_i)\}$
$y_i\,(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- The new formulation incorporating slack variables:

Find $\mathbf{w}$ *and* b *such that*
Minimize $\frac{1}{2}\,\mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ *for all* $\{(\mathbf{x}_i, y_i)\}$
$y_i\,(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$ *and* $\xi_i \geq 0$ *for all* i

- Parameter C can be viewed as a way to control overfitting.

Learning to rank

- Classification (regression) probably isn't the right way to think about approaching ad hoc IR:
  - Classification problems: Map to a unordered set of classes
  - Regression problems: Map to a real value
  - Ordinal regression problems: Map to an *ordered* set of classes
    - A fairly obscure sub-branch of statistics, but what we want here
- This formulation gives extra power:
  - Relations between relevance levels are modeled
  - Documents are good versus other documents for query given collection; not an absolute scale of goodness

Pairwise approach

- As before we begin with a set of judged query-document pairs.
- Instead, we ask judges, for each training query q, to order the documents that were returned by the search engine with respect to relevance to the query.
- We write $d_u < d_v$ for "d_u precedes d_v in the results ordering".
- We again construct a vector of features $x_u = (d_u , q)$ for each document-query pair
- For two documents d_u and d_v, we then form the vector of feature differences:

$$\Phi(d_u , d_v , q) = x_u - x_v$$

Pairwise approach: RankSVM

- If d_u is judged more relevant than d_j, then we will assign the vector $\Phi(d_u, d_v, q)$ the class $y_{u,v} = +1$; otherwise -1.
- This gives us a training set of pairs of vectors and "precedence indicators".
- We can then train an SVM on this training set with the goal of obtaining a classifier that returns

Find **w** and b such that

Minimize $\frac{1}{2}\,\mathbf{w}^T\mathbf{w} + C\sum_{u,v}\xi_{u,v}$ and for all $\{(x_u, x_v, y_{u,v})\}$

$y_{u,v}\,(\mathbf{w}^T(\mathbf{x_u} - \mathbf{x_v}) + b) \geq 1 - \xi_{u,v}$ and $\xi_{u,v} \geq 0$ for all u,v

RankSVM

$$\min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \sum_{u,v: y_{u,v}^{(i)} = 1} \xi_{u,v}^{(i)}$$

$$w^T \left(x_u^{(i)} - x_v^{(i)} \right) \geq 1 - \xi_{u,v}^{(i)}, \text{if } y_{u,v}^{(i)} = 1.$$

$$\xi_{uv}^{(i)} \geq 0, i = 1, \ldots, n.$$

x_u-x_v as positive instance of learning

Use SVM to perform binary classification on these instances, to learn model parameter *w*

- La solution du problème précédent fournit un vecteur w*.

RankSVM: phase de test

- Comment utiliser RSVM lors de la phase de test
  - On récupère les documents qui comportent les termes de la requête, soient (d1, d2, d3, ..)
  - Comment les trier ?
  - Construire toutes les combinaisons entre les docs puis vérifier
  - D_u plus pertinent que d_v ssi $(\mathbf{w}^T (\mathbf{x_u} - \mathbf{x_v}) + b) \geq 1$
- Cette utilisation est toutefois coûteuse. On utilise en fait en pratique directement le score « SVM »

$$RSV(q, d_u) = (w^* . x_u)$$

Pairwise approach

- Reduce ranking to pairwise classification
  - RankNet and Frank
  - RankBoost
  - Ranking SVM
  - MHR
  - IR-SVM

The Listwise approach

	The Listwise Approach	
	Listwise Loss Minimization	Direct Optimization of IR Measure
Input Space	Document set $\mathbf{x} = \{x_j\}_{j=1}^{m}$	
Output Space	Permutation π_y	Ordered categories $\mathbf{y} = \{y_j\}_{j=1}^{m}$
Hypothesis Space	$h(\mathbf{x}) = \text{sort} \circ f(\mathbf{x})$	$h(\mathbf{x}) = f(\mathbf{x})$
Loss Function	Listwise loss $L(h; \mathbf{x}, \pi_y)$	1-surrogate measure $L(h; \mathbf{x}, \mathbf{y})$

Fin