CS630 Representing and Accessing Digital Information

Transduction

Thorsten Joachims
Cornell University

Transductive Learning Process

Sampling Training data

- select random subset of l examples from DB of size *n*
  => $Z = [x_1, ..., x_l]$
- receive labels for these examples positive (+*1*) / negative (-*1*)
  => $Z = [(x_1, y_1), ..., (x_l, y_l)]$

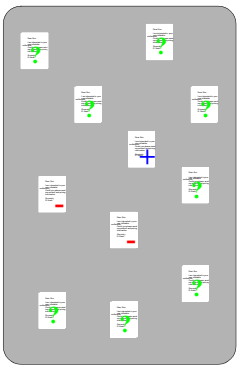
Goal of Learner

- predict the labels of the remaining examples $Z^*_x = [x^*_1, ..., x^*_k]$

Opportunity

- Learning algorithm can study the test examples $Z^*_x = [x^*_1, ..., x^*_k]$

Document DB



Example: Exploiting the Test Set

How would you classify the test set for Term/document matrix Aí

		nuclear	physics	atom	pepper	basil	salt	and
+	**D1**	**1**						**1**
?	D2	1	1	1				1
?	D3			1				1
?	D4				1	1		1
?	D5				1		1	1
-	**D6**					**1**	**1**	**1**

[Joachims, 1999]

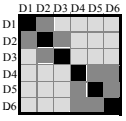
- training set {D1, D6}
- test set {D2, D3, D4, D5}

s-t MinCut for Transduction [Blum & Chawla, 01]

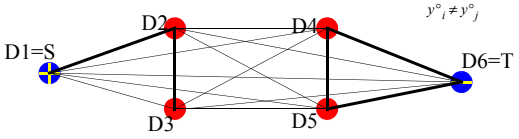
	nuclear	physics	atom	pepper	basil	salt	and
D1	**1**						**1**
D2	1	1	1				1
D3			1				1
D4				1	1		1
D5				1		1	1
D6					**1**	**1**	**1**

=>

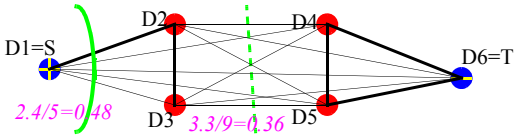
Adjacency matrix B=*AíA*:



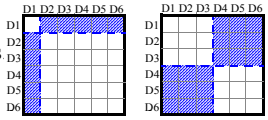
Find minimum cut with zero training error: $cut(B, y) = \sum_{y^\circ_i \neq y^\circ_j} B_{ij}$.



Ratio Cuts for Transduction [Joachims, 03]



s-t MinCut: Minimize sum of cut edges



Ratio Cut: Minimize average of cut edges => divide by "area"

$$ratiocut(B, y) = \frac{cut(B, y)}{|\{i : y = 1\}||\{i : y = -1\}|}$$

Spectral Graph Transducer (SGT)

Preprocessing:

- Compute k-NN graph A_{knn} and symmetricize $A = A_{knn} + A'_{knn}$.
- Compute eigendecomposition of (normalized) Laplacian $L_{ratio} = (A - B) = VDV'$ ($L_{norm} = B^{-1}(A - B) = VDV'$).
- Replace eigenvalues with $diag(D_{sparse}) = (\infty, 1, 4, 9, ..., d^2, \infty, ..., \infty)$ [Chapelle et al., 2002] [Belkin & Niyogi, 2002].

Prediction:

- Estimate $\gamma_i = \sqrt{\frac{l_{neg}}{l_{pos}}}$, $C_{ii} = \frac{l}{2l_{pos}}$ (pos), $\gamma_i = -\sqrt{\frac{l_{pos}}{l_{neg}}}$, $C_{ii} = \frac{l}{2l_{neg}}$ (neg).
- Solve

$$min_{w \in \Re^n} \; w'D_{sparse}w + c(Vw - \gamma)'C(Vw - \gamma) \qquad st \;\; w'w = n$$

via eigenvalue problem of size *2d* [Gander et al, 1989].

- prediction $y_i^\circ = sign(z_i - \Theta)$ with $z = Vw$ and $\Theta = \frac{1}{2}(\gamma_{pos} - \gamma_{neg})$

Transductive Support Vector Machines [Vapnik]

Objective: maximize margin δ on both training and test examples

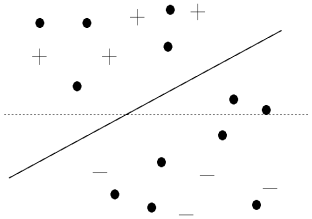
Training sample: $Z = [(x_1, y_1), \ldots, (x_l, y_l)]$

Test sample: $Z^*_x = [x^*_1, \ldots, x^*_k]$

Find solution $W^*(y^*, w) = \frac{1}{\delta^2}$ of

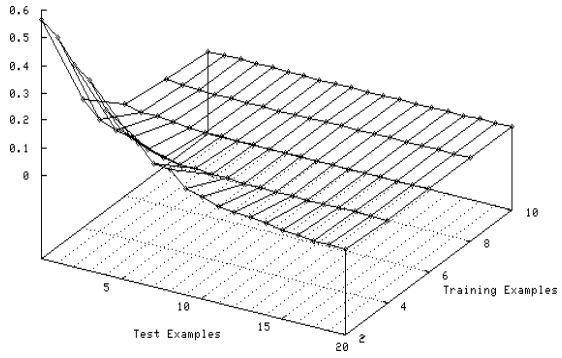
$$\min_{y^*_1 \ldots y^*_k \in \{-1,1\}} \; \min_{w \in \Re^d} \; w \cdot w$$

$$\text{subject to} \quad \begin{array}{l} y_1[w \cdot x_1 + b] \geq 1 \\ \ldots \\ y_l[w \cdot x_l + b] \geq 1 \\ y^*_1[w \cdot x^*_1 + b] \geq 1 \\ \ldots \\ y^*_k[w \cdot x^*_k + b] \geq 1 \end{array}$$

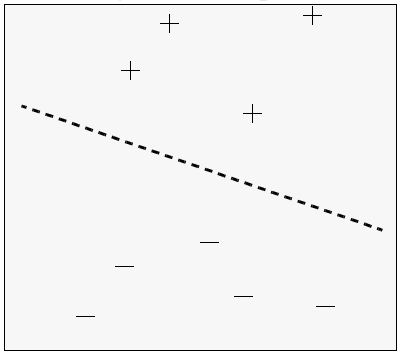


Simulation

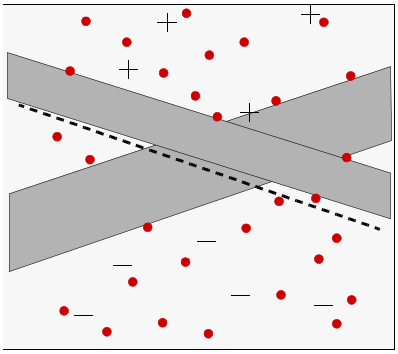
Target concept: *TCat([1:0:1],[0:1:1],[4:4:8])*



Why Does Adding Test Examples Reduce Error?

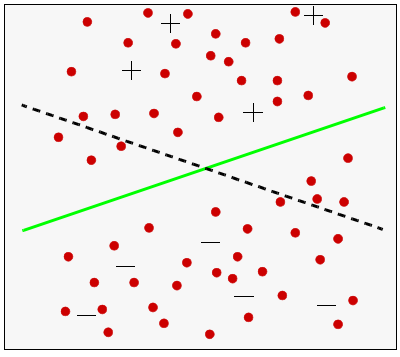


Why Does Adding Test Examples Reduce Error?



Margin $\delta \geq \frac{1}{\sqrt{2}}$

Why Does Adding Test Examples Reduce Error?



Margin $\delta \geq \frac{1}{\sqrt{2}}$

Experiment: Reuters-21578

Reuters Newswire Stories

- 90 categories
- 9603 training documents
- 3299 test documents

Experiment

- 10 most frequent categories
- 17 training documents
- 3299 test documents
- ca. 700-12000 features

	Bayes	SVM	TSVM
earn	78.8	91.3	95.4
acq	57.4	67.8	76.6
money-fx	43.9	41.3	60.0
grain	40.1	56.2	68.5
crude	24.8	40.9	83.6

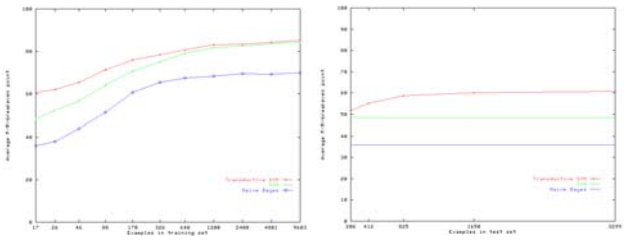
	Bayes	SVM	TSVM
trade	22.1	29.5	34.0
interest	24.5	35.6	50.8
ship	33.2	32.5	46.3
wheat	19.5	47.9	54.4
corn	14.5	41.3	43.7

=> avg. TSVM run-time: ~ 1 minute 40 seconds

Training Set vs. Test Set

Increasing training set size:
- avg. over 10 Reuters categories
- 3299 test documents
- feature selection: MI with local dictionaries of 1000 for Bayes

Increasing test set size:
- avg. over 10 categories
- 17 training documents



Co-Training (Blum & Mitchell)

Idea: Exploit two sufficiently redundant representations $X = A \times B$.

Scenarios:
- Web-page body text / Hyperlinks pointing to page
- sound of person saying " hello" / image of lip movements

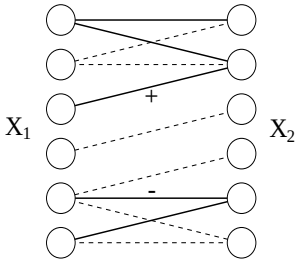
Training example: $(\langle a_i, b_j \rangle, y)$

Test example: $\langle a_i, b_j \rangle$

Composition: $\langle a_i, b_j \rangle \in X_1 \times X_2$

Hypotheses: $H_1 \times H_2$

Compatible:
$\langle a_i, b_j \rangle$ ♥ $h_1(a_i) = h_2(b_j)$

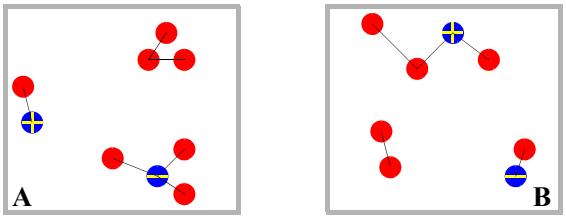


Co-Training [Blum & Mitchell]

Idea: Exploit two sufficiently redundant representations $X = A \times B$.

Scenario:
- Web-page body text (A) / Hyperlinks pointing to page (B)

Compatible: Perfect classifiers on A and B do not disagree!

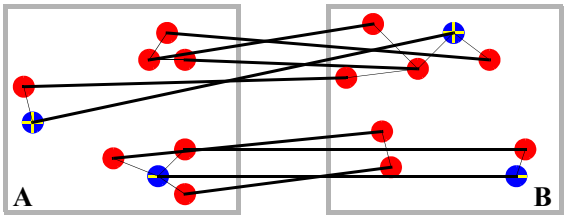


Co-Training [Blum & Mitchell]

Idea: Exploit two sufficiently redundant representations $X = A \times B$.

Scenario:
- Web-page body text (A) / Hyperlinks pointing to page (B)

Compatible: Perfect classifiers on A and B do not disagree!



=> SGT maximizes consistency between two k-NN classifiers.

Co-Training Experiment

	SGT	KNN	TSVM	SVM	B&M
cotrain	**3.3**	-	-	-	5.0
page+link	5.9	10.1	**4.3**	20.3	-
page	6.2	13.3	**4.6**	21.6	12.9
link	22.1	13.1	**8.9**	18.5	12.4

- Dataset: classifying course homepages from Blum and Mitchell
- 12 training examples, 1039 test examples
- Error on test set averaged over 100 random test/training splits
- Parameters:
  - SGT: cosine similarity, $c = 3200$, $d = 80$, 200NN in each view
  - others: optimized on the test set