Feature Selection using/for Transductive Support Vector Machine

Mr. Zhili Wu
Dr. Chun-hung Li

Department of Computer Science
Hong Kong Baptist University

Contents

- Introduction to Feature Selection
- Why TSVM works
- Technique sharing – not limited by TSVM
- Several technique highlights
- Conclusion
- Your comments & doubts

Feature selection (Competition)
- Impact of Weston's *Dataset selection*

- Your algorithm A*
- Other's algorithms A_1, ..., A_n
- You have $M=2^d-1$ possible feature sets for a d-dimensional dataset: $F_1,\dots,F_M$
- $L(A,D(F_i))$ = loss of algorithm A on dataset $D(F_i)$
- Your goal: find a feature set F* in $F_1,\dots,F_M$ so that $L(A^*,D(F^*)) < \min_{1,\dots,n} (L(A_i , D(F^*))$

"No Free Feature" Theorem

- **From "No Free Brunch" (Weston *NIPS 2002*)**

- The generalization error of two datasets for all algorithms is the same

$$E_A[\, R_{gen}^A[D]] = E_A[\, R_{gen}^A[D']]$$

- Since any two feature sets induce two new datasets

$$E_A[\, R_{gen}^A[D(F)]] = E_A[\, R_{gen}^A[D(F')]]$$

- **Consequence: Techniques are very important!**

Transductive SVM (SVMLight by Joachims)

Algorithm TSVM:

Input: – training examples $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$
– test examples $\vec{x}^*_1, \ldots, \vec{x}^*_k$

Parameters: – C, C^*: parameters from OP(2)
– num_+: number of test examples to be assigned to class +

Output: – predicted labels of the test examples $y^*_1, \ldots, y^*_k$

$(\vec{w}, b, \vec{\xi}, \_) := solve\_svm\_qp([(\vec{x}_1, y_1) \ldots (\vec{x}_n, y_n)], [], C, 0, 0);$

Classify the test examples using $< \vec{w}, b >$. The num_+ test examples with the highest value of $\vec{w} * \vec{x}^*_j + b$ are assigned to the class + ($y^*_j := 1$); the remaining test examples are assigned to class − ($y^*_j := -1$).

$C^*_- := 10^{-5};$ // some small number

$C^*_+ := 10^{-5} * \frac{num_+}{k - num_+};$

while$((C^*_- < C^*) \,||\, (C^*_+ < C^*))\{$ // Loop 1

  $(\vec{w}, b, \vec{\xi}, \vec{\xi}^*) := solve\_svm\_qp([(\vec{x}_1, y_1) \ldots (\vec{x}_n, y_n)], [(\vec{x}^*_1, y^*_1) \ldots (\vec{x}^*_k, y^*_k)], C, C^*_-, C^*_+);$

  while$(\exists m, l : (y^*_m * y^*_l < 0) \& (\xi^*_m > 0) \& (\xi^*_l > 0) \& (\xi^*_m + \xi^*_l > 2))\ \{$ // Loop 2

    $y^*_m := -y^*_m;$ // take a positive and a negative test

    $y^*_l := -y^*_l;$ // example, switch their labels, and retrain

    $(\vec{w}, b, \vec{\xi}, \vec{\xi}^*) := solve\_svm\_qp([(\vec{x}_1, y_1) \ldots (\vec{x}_n, y_n)], [(\vec{x}^*_1, y^*_1) \ldots (\vec{x}^*_k, y^*_k)], C, C^*_-, C^*_+);$

  }

  $C^*_- := min(C^*_- * 2, C^*);$

  $C^*_+ := min(C^*_+ * 2, C^*);$

}

return$(y^*_1, \ldots, y^*_k);$

Simpler Explanation to TSVM

1. Train a SVM on labeled data only
2. Predict unlabeled data to an assigned fraction of Pos, others being Neg
3. Train the whole dataset
   - switch some pairs of Pos/Neg for some goodness measure, repeat 3
4. Repeat 2 & 3 till unlabeled data contribute much

Why TSVM Works for FS Competition

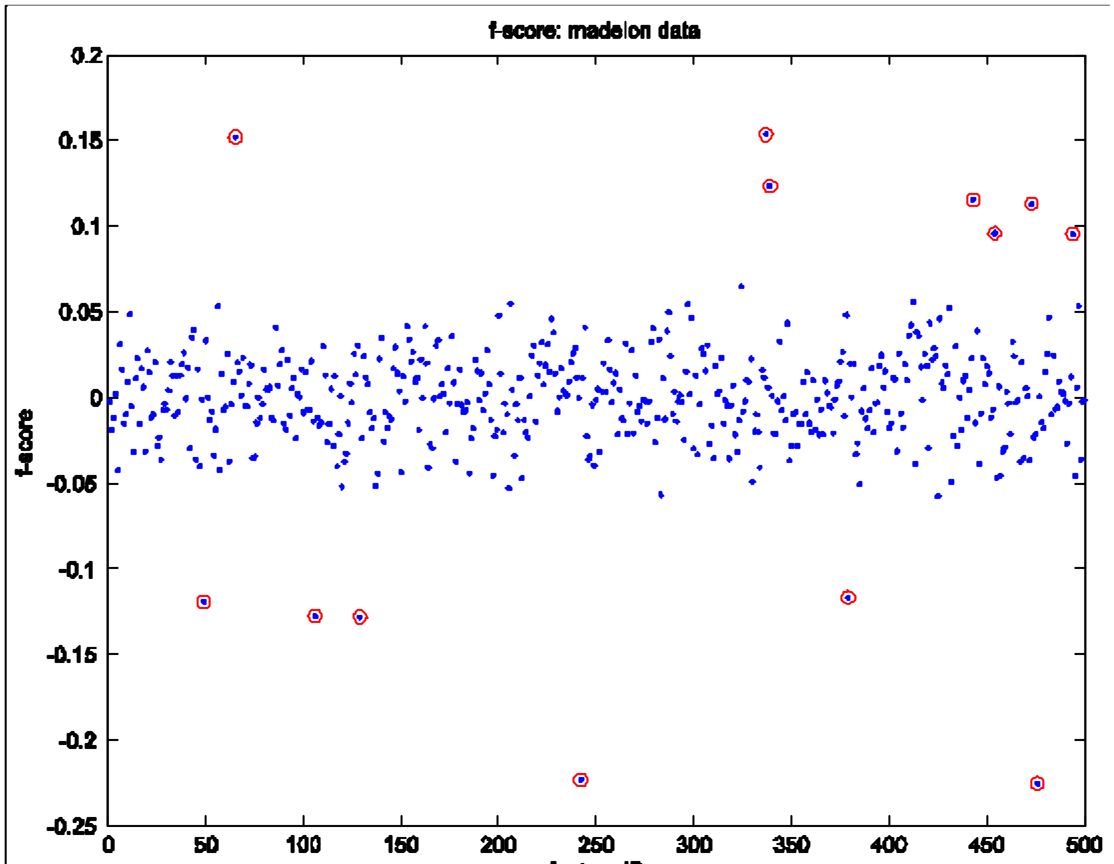
- **unlabeled (validating+testing) data provided**
- **accuracy is the first priority measure**
- **Fraction of Pos/Neg unlabeled samples provided**
- **Also, effective & compatible tools:**
  - Dr. Chih-Jen Lin's SVMLIB
  - SVMLIB + SVMLIGHT

Feature Selection Using/for Transductive SVM (TSVM) – Technique Summary

	Arcene	Gisette	Dexter	Dorothea	Madelon
	Normalize 1 (0 mean, unit std)				
Score		Fisher Score	F-score	Odd Ratio	F-score
	7~20 PCs by PCA		D_ij/Sqrt(row-sum*col-sum)	D_ij/Sqrt(row-sum*col-sum)	Normalize 1
			Scale feature by f-score	Scale feature by f-score	
Kernel	RBF (C=2^5, g=2^-6)	Poly 2	Linear	Linear (C+/C-=19.5)	RBF (g=1,c=1)
Transduction	Yes	yes	Yes	No	Yes
Further feature reduction			Use w to select feature and rescale feature		
Remarks:	Model selection by CV seems to overfit ?			MI, BNS, BER score, F-score	T-test
BER & (Rank by submissions on 1st/Dec)		1.58(11th)	4.4(6th)	11.52(11th)	

Madelon – A Fisher-Score Variant

- $(\mu_+ - \mu_-)/(s_+ + s_-)$
- 13 features are selected



Dorothea oddRatio

		Feature value	
		0	1
Truth	Class -1	a	b
	Class +1	c	d

- **ExpProb oddRatio[1] - for unbalanced class**

  exp(P(1|class+) – P(1|class-)) = exp(d/(c+d) – b/(a+b))

- **Other Measures like BNS [2], MI, …**

- **Is BER a score indicating goodness of features?**

  The balanced error rate (BER) is the average of the errors on each class:

  BER = 0.5*(b/(a+b) + c/(c+d)).

1. Feature selection for unbalanced class distribution and Naïve Bayes, Dunja Mladenic, Marko Grobelnik
2. An Extensive Empirical Study of Feature Selection Metrics for Text Classification , *George Forman* , JMLR 2003 special issue on variable and feature selection

Dexter: A Simple Linear-TSVM-RFE

1. Prune some features using scores easily calculated
2. Rescale remaining features by scores
3. Train a Linear TSVM (with good generalization ability)
4. Calculate the feature weight w
5. Rank features and rescale features by w
6. Repeat 3~5 till a balance of feature relevance & accuracy

Conclusion

1. No Free Feature
2. TSVM
3. Techniques
    1. Scoring Methods
    2. TSVM RFE

4. Other important issues not mentioned:
    1. Model selection
    2. Normalization
    3. ...

Your Comments!

Thanks !