# Combining SVMs with Various Feature Selection Strategies

Yi-Wei Chen and Chih-Jen Lin
Department of Computer Science, National Taiwan University,
Taipei 106, Taiwan

## SVM Idea

- Map features into a higher dimensional space
- Find separating hyperplane with maximum margin
- Amounts to solving the quadratic optimization problem:

min [w, b, ?] 0.5 * $\mathbf{w}^T\mathbf{w}$ + C * ? $?_i$
*subject to* $y_k$ * ( $\mathbf{w}^T$ * F($x_k$) + b) = 1 – $?_k$
*and* $?_k$ = 0

## Finding the parameters

- Parameter ? of the RBF kernel
- Parameter C of the SVC
- Simple heuristic:
  - Create grid with pairs of (C, ?)
    - $\log_2$C in {-5, -3, .. , 15}
    - $\log_2$? in {-15, -13, .. , 3}
- Perform 5-fold CV on each (C, ?)-pair
- Choose (C, ?)-pair with smallest CV-BER

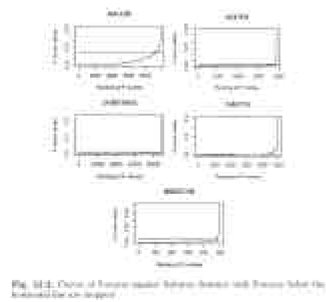
## Feature Selection Strategies

- 4 strategies were tried:
  - No selection (SVM)
  - F-score (F-score + SVM)
  - F-score + random forest (F-score + RF + SVM)
  - Random forest + radius-bound SVM (RF + RM-SVM)

## F-score

- $F = \dfrac{\textbf{between class}\text{ variance}}{\text{pooled }\textbf{within class}\text{ variance}}$
- Disadvantage: F-score doesn't incorporate **mutual information** among features
- Choose the threshold for features to select with the following heuristic:
  - Calculate all F-scores and select some thresholds
  - For each threshold, train a SVC using 5 different (random) splittings of the training set (~ 5-fold cross-validation)
  - Choose the threshold with lowest average validation error

## F-score diagrams



## Random Forest

- Can be used for classification as well as feature importance
- Will be covered later in the lecture
- Suitable for rather small feature sets
- They found, that random-forest feature selection kept **all** the features obtained from the F-score selection process

## Radius Margin Bound SVM

- RBF kernel with feature-wise scaling:
  $k(x, x') = \exp( ?\ ?_i * (x_i – x'_i)^2 )$
- This is rather time-consuming and only applicable to small feature sets
- Thus, they only apply it only to MADELON (500 features)
- But the performance is not significantly better than a standard SVM (next slide)

## Experimental Results

**Table 12.1.** Comparison of different methods during the development period: BERs of validation sets (in percentage); bold-faced entries correspond to approaches used to generate our final submission

| Dataset | ARCENE | DEXTER | DOROTHEA | GISETTE | MADELON |
|---|---|---|---|---|---|
| SVM | 13.31 | 11.67 | 33.98 | 2.10 | 40.17 |
| F+SVM | **21.43** | **8.00** | 21.38 | **1.80** | 13.00 |
| F+RF+SVM | 21.43 | 8.00 | **12.51** | 1.80 | 13.00 |
| RF+RM-SVM[4] | – | – | – | – | **7.50** |

**Table 12.3.** F-score threshold and the number of features selected in F+SVM

| Dataset | ARCENE | DEXTER | DOROTHEA | GISETTE | MADELON |
|---|---|---|---|---|---|
| F-score threshold | 0.1 | 0.015 | 0.05 | 0.01 | 0.005 |
| #features selected | 661 | 209 | 445 | 913 | 13 |
| #total features | 10000 | 20000 | 100000 | 5000 | 500 |

## Challenge Results

**Table 12.4. NIPS 2003 challenge results on December $1^{st}$**

| Dec. $1^{st}$ | Our best challenge entry | | | | | The winning challenge entry | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | Score | BER | AUC | Feat | Probe | Score | BER | AUC | Feat | Probe | Test |
| OVERALL | 52.00 | 9.31 | 90.69 | 24.9 | 12.0 | 88.00 | 6.84 | 97.22 | 80.3 | 47.8 | 0.4 |
| ARCENE | 74.55 | 15.27 | 84.73 | 100.0 | 30.0 | 98.18 | 13.30 | 93.48 | 100.0 | 30.0 | 0 |
| DEXTER | 0.00 | 6.50 | 93.50 | 1.0 | 10.5 | 96.36 | 3.90 | 99.01 | 1.5 | 12.9 | 1 |
| DOROTHEA | -3.64 | 16.82 | 83.18 | 0.5 | 2.7 | 98.18 | 8.54 | 95.92 | 100.0 | 50.0 | 1 |
| GISETTE | 98.18 | 1.37 | 98.63 | 18.3 | 0.0 | 98.18 | 1.37 | 98.63 | 18.3 | 0.0 | 0 |
| MADELON | 90.91 | 6.61 | 93.39 | 4.8 | 16.7 | 100.00 | 7.17 | 96.95 | 1.6 | 0.0 | 0 |

**Table 12.5. NIPS 2003 challenge results on December $8^{th}$**

| Dec. $8^{th}$ | Our best challenge entry | | | | | The winning challenge entry | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | Score | BER | AUC | Feat | Probe | Score | BER | AUC | Feat | Probe | Test |
| OVERALL | 49.14 | 7.91 | 91.45 | 24.9 | 9.9 | 88.00 | 6.84 | 97.22 | 80.3 | 47.8 | 0.4 |
| ARCENE | 68.57 | 10.73 | 90.63 | 100.0 | 30.0 | 94.29 | 11.86 | 95.47 | 10.7 | 1.0 | 0 |
| DEXTER | 22.86 | 5.35 | 96.86 | 1.2 | 2.9 | 100.00 | 3.30 | 96.70 | 18.6 | 42.1 | 1 |
| DOROTHEA | 8.57 | 15.61 | 77.56 | 0.2 | 0.0 | 97.14 | 8.61 | 95.92 | 100.0 | 50.0 | 1 |
| GISETTE | 97.14 | 1.35 | 98.71 | 18.3 | 0.0 | 97.14 | 1.35 | 98.71 | 18.3 | 0.0 | 0 |
| MADELON | 71.43 | 7.11 | 92.89 | 3.2 | 0.0 | 94.29 | 7.11 | 96.95 | 1.6 | 0.0 | 1 |

## Their Conclusion

- Pure SVM without feature selection works well on GISETTE and ARCENE
- On MADELON the winning team used a Bayesian SVM, which gives very similar (but better) results
- They tried to determine, which feature selection methods work best with SVMs, but broader investigation on different data sets is needed

## Combining a Filter Method with SVMs

Thomas Navin Lal, Olivier Chapelle and Bernhard Schölkopf
Max-Planck-Institute for Biological Cybernetics, Tübingen, Germany

## General approach

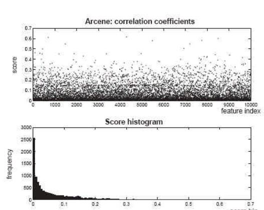
- For ARCENE, DEXTER, GISETTE and MADELON a hard-margin SVM is trained
- For DOROTHEA (which is unbalanced) a soft-margin SVM is trained
- For DOROTHEA, GISETTE and MADELON a gaussian kernel is used
- For ARCENE and DEXTER a linear kernel is used.

## Finding the parameters

- C is found by 20-fold cross-validation (for the soft-margin SVM)
- The gaussian kernel parameter s is found by a heuristic approach:
  - For each k, let $t_k$ be the distance of $x_k$ to the set formed by all points of the other class
  - s is then set to the mean of the $t_k$ values

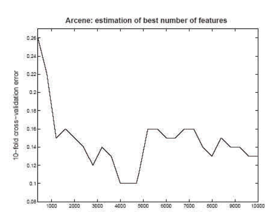
## Feature ranking

- Features are ranked using Fisher-score
- Only few features have a high score:




## Number of features

- For different numbers of best features N, a SVM is trained using 10-fold cross-validation
- The N with lowest average test-error is chosen



## Challenge results

## Comparison

- Both teams use SVM classifiers
- The difference in performance must be related to finding the hyperparameters
- First group searches for both parameters together (parameter grid)
- Second group does an independant search for each parameter
- Choosing the number of best features to use (with F-score feature selection) is done in a similar way (5- and 10-fold CV)

## My conclusion

- The two teams did exactly what we did when experimenting with GISETTE:
  - Trying to find optimal parameters for the model, which would lead to the smallest error
- Often, (simple) heuristics are used for this task
- An idea would be to use more sophisticated heuristic methods to do a more structured search in the parameter space