Abstract
We present the results of our submission for the Nips 2003 Feature Extraction challenge, which was organized by I. Guyon et al.

1 Background
Two possible drawbacks when learning on entire dataset:
- Computational load too high when using a complicated learner on the whole training data set.
- Statistical stability of classifier is affected by "learning" garbage-features.
It is thus often advisable to divide the learning in two steps:
1. Find relevant features.
2. Do the learning only on the relevant features.

2 Contest
- Five datasets from various areas.
- Two-class classification problems.
Challenge participants were classified according to their balanced error rate (BER) on the test set.

3 Some Observations
- Support Vector Machines are usually good classifiers to get started with.
- Often the choice of filter is not too crucial.

4 Support Vector Machines
As we use support vector machines all over again, we think it is useful to review some fundamentals of the SVMs, such that we have them handy.

\[
\begin{align*}
\text{Soft margin SVM with 2-norm regularization,} \\
&\text{minimize } ||w||^2 + C \sum_j \xi_j \\
&\text{subject to } y_j(w \cdot x_j + w_0) \geq (1 - \xi_j)
\end{align*}
\]
Radial basis kernel for the "kernel-trick":
\[
k(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right)
\]

5 Results
5.1 Dorothea: Thrombin dataset
Learning on training and validation data set.

\[
\text{my SVC = svc(} \text{``coef0=1'', 'degree=3', 'gamma=0', 'shrinkage=0.1''});}
\]
\[
\text{my model = chain(} \text{``standardize, Ftest('f max=2740''), normalize, my SVC''});}
\]

5.2 Arcene: cancer diagnosis
Learning on training and validation data set.

\[
\text{my SVC = svc(} \text{``coef0=1'', 'degree=3', 'gamma=0', 'shrinkage=0.1''});}
\]
\[
\text{my model = chain(} \text{``standardize, Ftest('f max=2740''), normalize, my SVC''});}
\]

5.3 Gisette: handwritten digits
Construct new features by convolution with an exponential kernel.
\[
\text{my classif = svc(} \text{``coef0=1'', 'degree=4', 'gamma=0', 'shrinkage=0.03''});}
\]
\[
\text{my model = chain(} \text{``convolve(exp_ker(} \text{``dim1=9', 'dim2=9', 'sigma1=1.7', 'sigma2=1.7''), s2n(} \text{``f max=4500''), normalize, my classif''});}
\]

5.4 Madelon: random data
Learning on training and validation data set.

\[
\text{Data is inherently non-linear. Don't punish non-linear functions too much (shrinkage of SVM small) and use a non-linear filter method (relief).}
\]
\[
\text{my classif = svc(} \text{``coef0=1'', 'degree=0', 'gamma=0.3', 'shrinkage=0.3''});}
\]
\[
\text{my model = chain(} \text{``probe(relief, {'p num=2000', 'f max=20'}), standardize, my classif''});}
\]

5.5 Dexter: filters text
\[
\text{my classif = svc(} \text{``coef0=1'', 'degree=1', 'gamma=0', 'shrinkage=0.5''});}
\]
\[
\text{my model = chain(} \text{``s2n('f max=2000'), normalize, my classif''});}
\]

6 Overall results
Comparison to the best challenge entry for each data set.