Support Vector Machines for Large Scale Text Mining in R

Ingo Feinerer[1] Alexandros Karatzoglou[2]

[1]Vienna University of Technology, Austria

[2]Telefonica Research, Spain

COMPSTAT'2010

Motivation

- Machine learning and data mining require classification
- Large amounts of data
- Use R for data intensive operations
- Text mining is especially resource hungry
- Highly sparse matrices
- Need of scalable implementations

Large Scale Linear Support Vector Machines

Modified Finite Newton l_2-SVM

Given

- m binary labeled examples $\{x_i, y_i\}$ with $y_i \in \{-1, +1\}$, and
- the SVM optimization problem

$$w^* = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{m} c_i l_2(y_i w^T x_i) + \frac{\lambda}{2} \|w\|^2$$

the modified finite Newton l_2-SVM method gives an efficient primal solution.

R Extension Package `svmlin`

Features

Implements l_2-SVM algorithm.

- Extends original C++ version of `svmlin` by Sindhwani and Keerthi (2007).

Adds support for

- multi-class classification (one-against-one and one-against-all voting schemes),
- cross-validation, and
- a broad range of sparse matrix formats (`SparseM`, `Matrix`, `slam`).

R Extension Package `svmlin`

Interface

```
model <- svmlin(matrix,
                labels,
                lambda = 0.1,
                cross = 3)
```

- Regularization parameter of $\lambda = 0.1$
- 3-fold cross-validation
- `model` can be used with the `predict()` function

R Extension Package `tm`

Text mining framework in R

- Functionality for managing text documents
- Abstracts the process of document manipulation
- Eases the usage of heterogeneous text formats (XML, ...)
- Meta data management
- Preprocessing via transformations and filters

Exports

- (Sparse) term-document matrices
- Interfaces to string kernels

Available via CRAN

Data

Reuters-21578

- News articles by Reuters news agency from 1987
- 21578 short to medium length documents in XML format
- Wide range of topics (M&A, finance, politics, ...)

SpamAssassin

- Public mail corpus
- Authentic e-mail communication with classification into normal and unsolicited mail of various difficulty levels
- 4150 ham and 1896 spam documents

20 Newsgroups

- 19997 e-mail messages taken from 20 different newsgroups
- Wide field of topics, e.g., atheism, computer graphics, or motorcycles

Preprocessing

Creation of term-document matrices

- 42 seconds for Reuters-21578
- 31 seconds for SpamAssassin
- 75 seconds for 20 Newsgroups

Term-document matrix size

- Reuters-21578: 65973 terms, 21578 documents, 24 MB
- SpamAssassin: 151029 terms, 6046 documents, 24 MB
- 20 Newsgroups: 175685 terms, 19997 documents, 46 MB

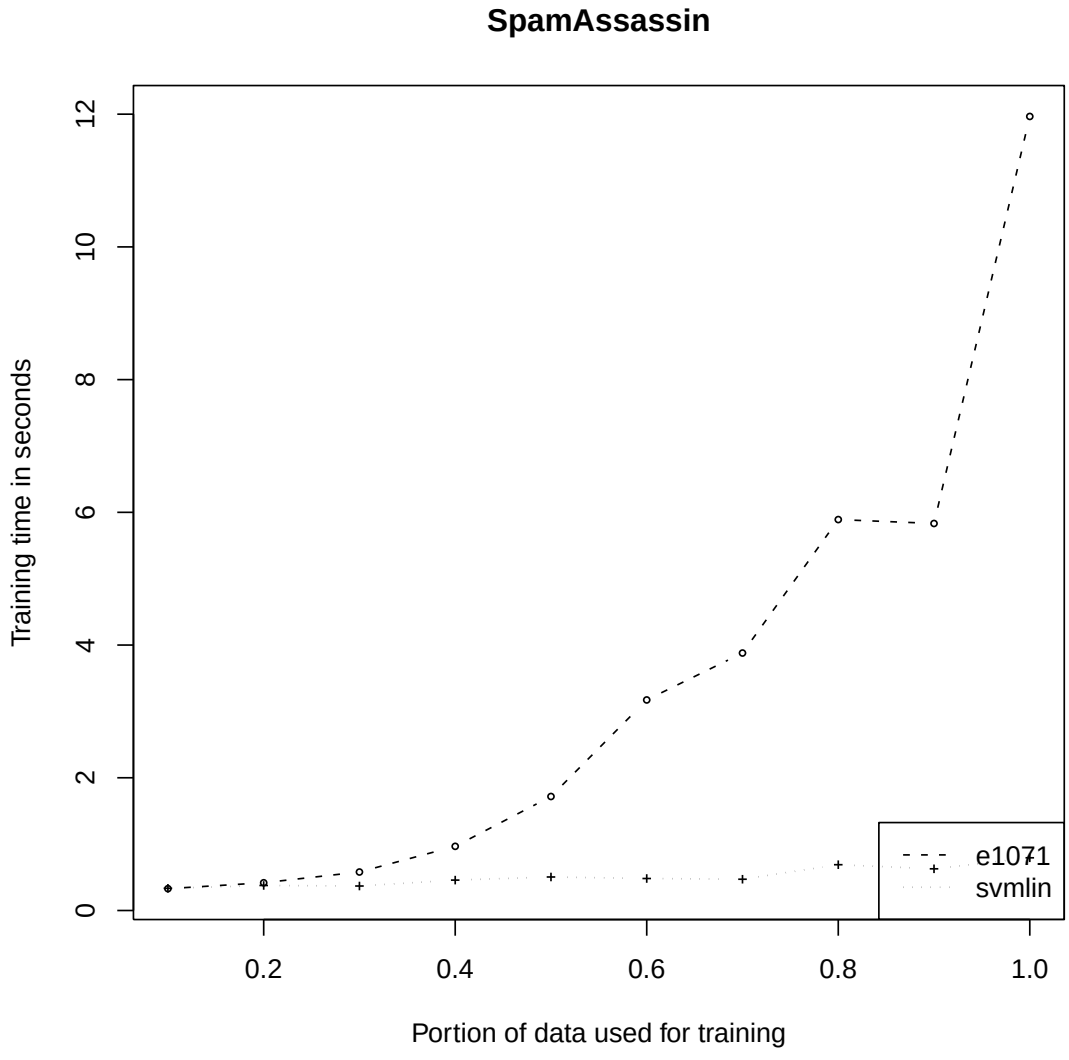
Protocol

Compare SVM implementations

- Runtime of `svm` (package `e1071`) vs. `svmlin`
- For `svm` we use a linear kernel and set the cost parameter to $\frac{1}{\lambda}$
- Initially sample $\frac{1}{10}$ from data set for training
- Increase training data in $\frac{1}{10}$ steps
- Compare classification performance using 10-fold cross-validation

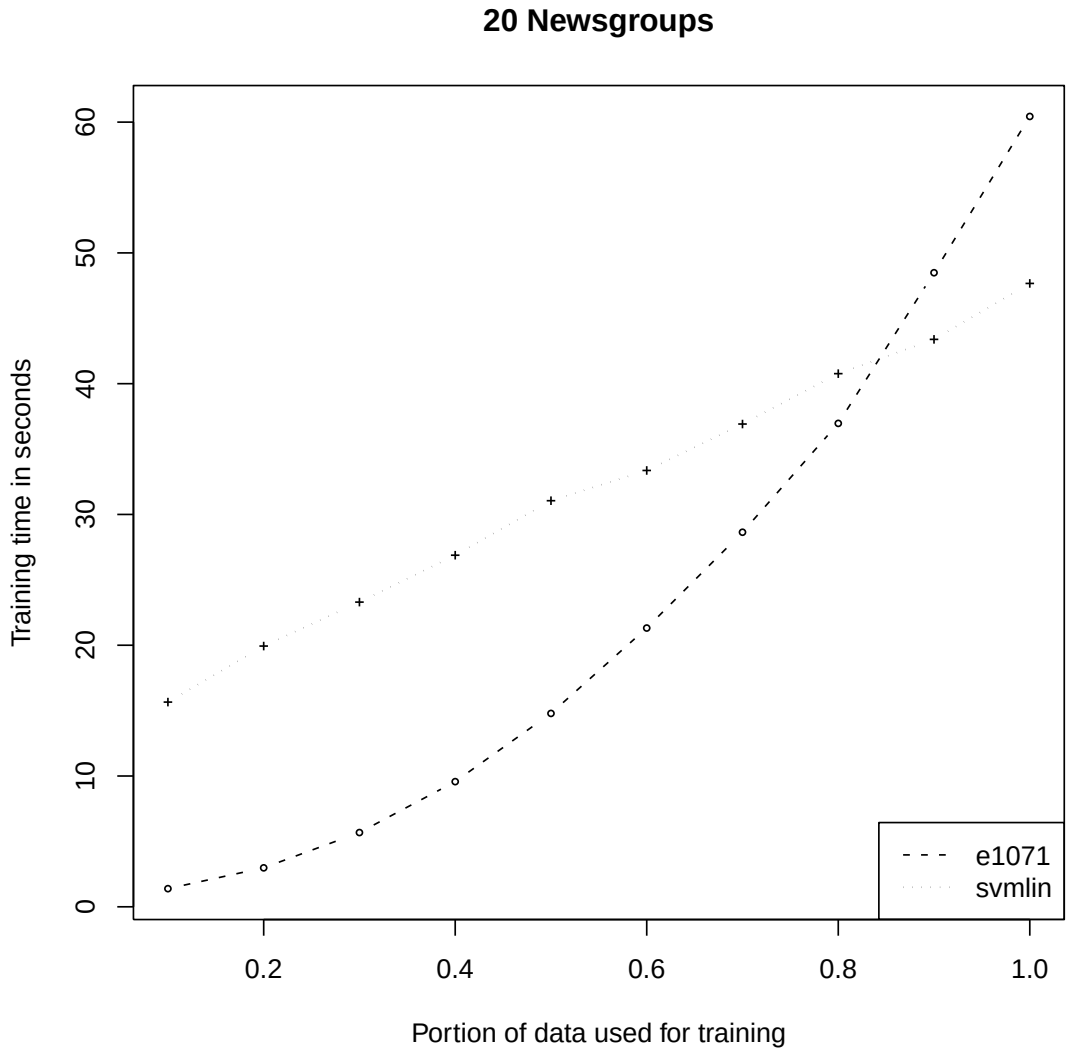
Results

SpamAssassin



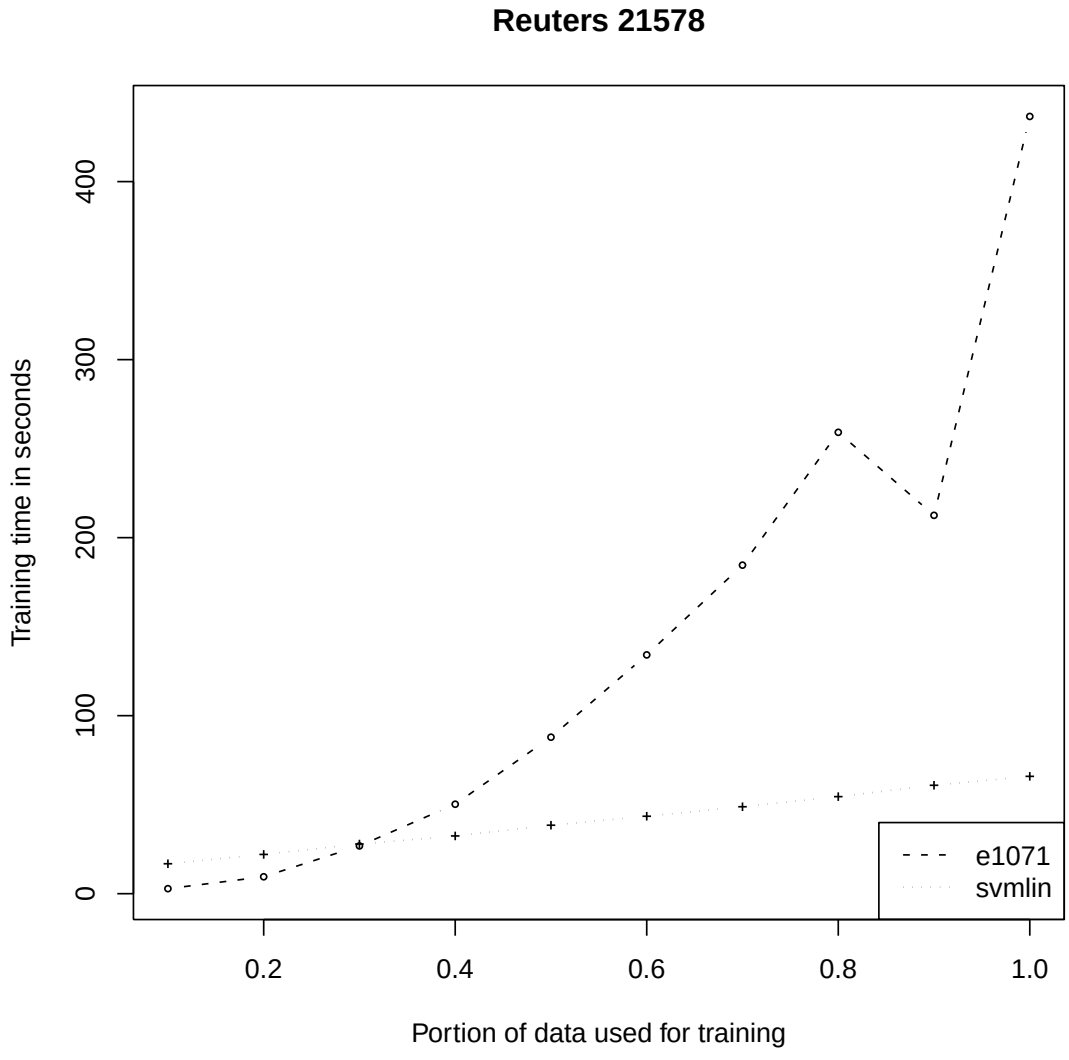
Results

20 Newsgroups



Results

Reuters-21578



Conclusion

- `svmlin` extension package
- Takes advantage of sparse data
- Computations are done in primal space (no kernel necessary)
- Comparison with state-of-the-art `svm`
- Linear scaling, faster training times