Current Projects Overview

(SHOGUN, ARTE, Splicer, POIMs, Fast(er) SVMs)

Sören Sonnenburg[†]

[†] Fraunhofer FIRST.IDA, Berlin

Fraunhofer Institut
Rechnerarchitektur
und Softwaretechnik

Fraunhofer FIRST Institut Rechnerarchitektur und Softwaretechnik

Outline

Machine Learning Open Source Software

with Cheng Soon Ong and Mikio Braun

- Overview
  - Recognized that there is the need to have OSS in ML (better reproducibility,...)
  - We organized a NIPS'06 workshop about MLOSS.
  - Conclusion: There is a market for MLOSS.
- Status
  - Asked JMLR for a special track on MLOSS and looks like we will become JMLR editors...
  - Currently writing a JMLR position paper about MLOSS (submitted draft#2 yesterday).
- Future ...

Outline

Splice Site Recognition

with Petra Philips, Gabriele Schweikert, Gunnar Rätsch

- Overview
  - We recognized other people are still publishing splice papers in (BMC) Bioinformatics using weak methods.
  - AIM: Show that we outperform other approaches using our standard techniques (which we did not publish in a journal)
- Status
  - Gabi/Petra did experiments, submitted initial paper (end of March)
  - We had trouble with a competing paper, results not reproducible. After several Emails - a corrected version appeared (performance much worse).
  - We are overdue..

Splice Site Recognition

- Future (TODO ASAP!!)
  1. re-do experiments (Jonas ?)
  2. extend comparison with other approaches
  3. polish methods part
  4. put everything on website
     - models,data,program to train/test
     - silencer/enhancer tables
     - parameters
     - whole genome predictions (custom tracks)

Outline

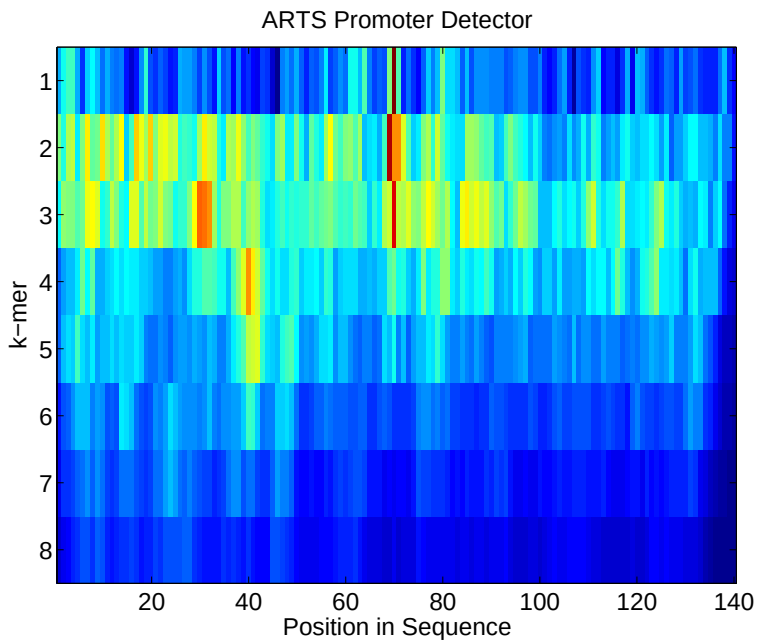
Interpretability, POIMs and Consensus Sequences

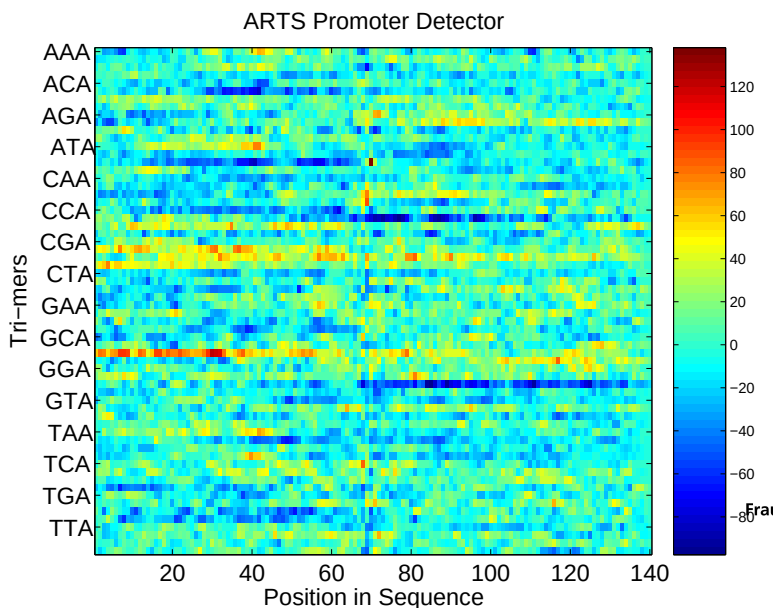
with Alexander Zien, Petra Philips

- Overview
  - We would like to better understand what exactly our (very precise, layer 1) ML-models learn.
  - Could be used to identify (altsplice/splice/tss/...) silencers/enhancers (hope to do better than MKL approximations).
- Two Ideas
  1. We know SVM is a linear classifier in very high dimensional kernel feature space, $\mathbf{w} = \sum_{i=1}^{N} \alpha_i \Phi(\mathbf{x}_i)$. We learned that $\mathbf{w}$. Make use of it!
  2. $f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b$, determine maximum scoring $\mathbf{x}$, i.e. $\text{argmax}_x f(\mathbf{x})$.

Status

- Status
  1. POIMS, show average weight of $k-$mer at certain position. $\Rightarrow$ implemented in shogun. In use for TSS/PolyA/Splice
  2. Max. Scoring sequence is a dynamic programming problem (basically works like viterbi in markov chain, with states k-mers) $\Rightarrow$ implemented for WD and spec kernel in shogun (drawback computationally demanding and for WD kernel skips $k-$mers with weight 0.





Future

- We think it works (more or less) for TSS/splice signals.
- Future
  1. We need further understanding...
  2. Alex has ideas how to also make use of 0-weight $k-$mers, it is still unclear whether the max. scoring sequence is meaningful or not.
- Soon: Write Bioinformatics method paper explaining the approach/providing a tool...
- Later: Would be nice to detect real+new silencer/enhancer to be wetlab confirmed. Anyone ?

Outline

TSS & PolyA detection

with Petra Philips, Alexander Zien, Regina Bohnert, Gunnar Rätsch

- Overview
  - Lets find TSS and PolyA's using SVMs and our kernel machinery and get state-of-the-art results :-)
- Status
  - Method is SVM using WDS,Spectrum kernel (ISMB'06)
  - Are doing this for many organisms, more data genome wide + wetlab validation

Future

- training using mGene framework
- adjust ISMB paper
- submit to PLoS (before mGene is out) ...

Outline

SHOGUN

with Gunnar Rätsch, Fabio De Bona, Andre Noll
Known Features:

- Toolbox's focus is on kernel methods esp. Support Vector Machines (SVMs) for computational biology
- Includes a variety of common kernels (Linear, Polynomial, Gaussian) and recent String Kernels
- Kernels can be combined; weighting can be learned using Multiple Kernel Learning.
- Tuned for large scale data sets (parallelized SVM training on 10,000,000 DNA sequences in 27hrs, parallelized SVM testing on 7 billion examples)

New Methods I

- more SVMs :-)
  - GNNP-SVM (Generalized Nearest Point Problem), L2-penalized slacks
  - GMNP-SVM (Generalized Nearest Point Problem for Multiclass), mulitclass svm 1 vs. all
  - LibSVM multiclass 1 vs. 1
  - LibSVM oneclass
  - SVM-Lin (L2-SVM-MFN - Modified Finite Newton Method), L2-penalized slacks, no bias
  - GPDT-SVM (Gradient Projection-based Decomposition Technique), same chunking technique as SVMLight, but faster core solver (qpsize 500 is default!); was a lot faster in experiments; implemented `linadd`
  - Subgradient SVM (only linear, submitted for NIPS'07)

New Methods II

- for LPM (L1-penalized **w**)
  - LPM (via CPlex)
  - LPBoost
  - SubGradient LPM
- for regression...
  - (LibSVR/SVRLight) Kernel Ridge regression
- LDA
- many distances for strings (Manhatten,...)

New Methods III

- general alphabet (PROTEIN,DNA,ASCII,BYTE) - spectrum kernel for spam etc
- to compute POIMs
- to compute consensus sequences
- create virtual data from a single string by a sliding window (genome wide evaluations with little memory requirements possible)

Interfaces

- matlab
  - `sg('send_command', 'new_classifier LDA')`
  - `sg('send_command', 'train_classifier')`
  - `[b,w]=sg('get_classifier')`
  - `out=sg('classify')`
  - can now deal with sparse features directly
- python-modular
  - swig based, easy to extend and develop
  - really object oriented
  - great for complex scenarios ($>$ 1 classifier etc...)

Under the hood

- Buildbot (Fabio)
  - Aim is to automagically build shogun for each interface on each svn commit
  - to detect compile breakage and
  - to detect bugs (methods that worked before...)
- Testsuite (Jonas)
- Code Cleanup (Andre, me)
  - all char-matrices are now strings (use a single kernel implementation now)
  - generic math functions,...
  - more examples (fixed non-working ones)

Future

1. buildbot really for all interfaces (Fabio & Andre ?)
2. tests for all interfaces (Jonas ?)
3. make other interfaces consistent
4. finish R interface via swig and publish in R community (Fabio ?)
5. more examples and doxygen source code documentation (me)
6. SubgradientSVM for kernels (me)
7. SVM-Perf with bias, CPLEX primal, CPLEX dual SVM for reference (me)
8. manual anyone ?

Outline

SubgradientSVM/LPM

with Vojtech Franc

- Motivation
  - Training SVMs/LPMs in even shorter time...
- Status
  - For LPM up to 500 times faster than CPLEX, up to 30 times faster than LPBoost
  - For SVM up to 87 times faster than SVMLight; 1Mio Splice data in 18 minutes instead of 18 hrs
  - Only Linear, problems for very sparse datasets or very small epsilon
- Future
  - Fix these issues: $\varepsilon-$subgradients ? Solve problem for most dense vec's first ?
  - Kernelize.