Empowering Citizens. Smarter Societies.

Insight
Centre for Data Analytics

Effective Linear Models for Learning with Sequences and Time Series

Georgiana Ifrim
University College Dublin, Ireland
georgiana.ifrim@ucd.ie

Invited Talk
ECML/PKDD Workshop on New Frontiers in Mining Complex Patterns
16/09/2019

A World Leading SFI Research Centre

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI)under Grant Number SFI/12/RC/2289, co-funded under the European Regional Development Fund

Outline

- Data & Applications
- Sequence Learning Approaches
- Linear Models for Sequences
- Linear Models for Time Series
- Experiments & Results
- Conclusion

Data & Applications: Everything is a Sequence

DNA

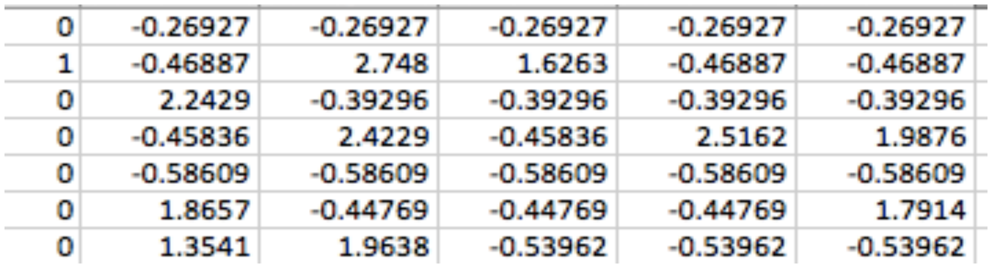
Value	Data points
290.507	AGGGCATCATGGAGCTGTCCAG
679.305	ATCACAATTTTGCCGAGAGCGA
1998.715	GTACACCCCGTTCGGCGGCCCA
447.803	CCTTTAGCCCATCGTTGGCCAA

Malware

Byte sequence

Class	Data points
+1	C7 01 24 04 5F 0E EA DC 00 E9 D6 4A 00 0C 66
+1	74 13 BA EF 01 00 06 68 95 14 88 B7 00 0F 0E
-1	08 F9 C8 1A 80 C1 8B 48 40 00 89 51 10 B8 04
-1	B8 00 00 00 00 50 E8 D8 00 00 00 83 C4 04 53

Sensors



0	-0.26927	-0.26927	-0.26927	-0.26927	-0.26927
1	-0.46887	2.748	1.6263	-0.46887	-0.46887
0	2.2429	-0.39296	-0.39296	-0.39296	-0.39296
0	-0.45836	2.4229	-0.45836	2.5162	1.9876
0	-0.58609	-0.58609	-0.58609	-0.58609	-0.58609
0	1.8657	-0.44769	-0.44769	-0.44769	1.7914
0	1.3541	1.9638	-0.53962	-0.53962	-0.53962

Data & Applications: Everything is a Sequence

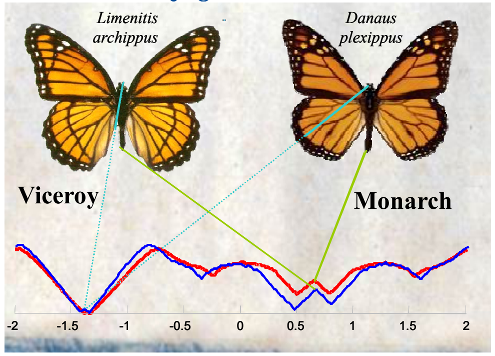
Music

Key	Sequence of notes
C+	C |D |E |F |G |A |B |C

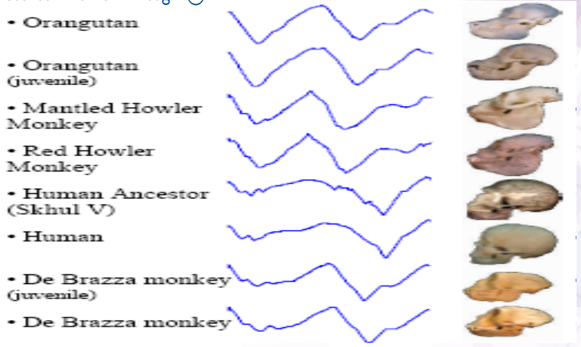
Images

Shapes

source : Eamonn Keogh ©

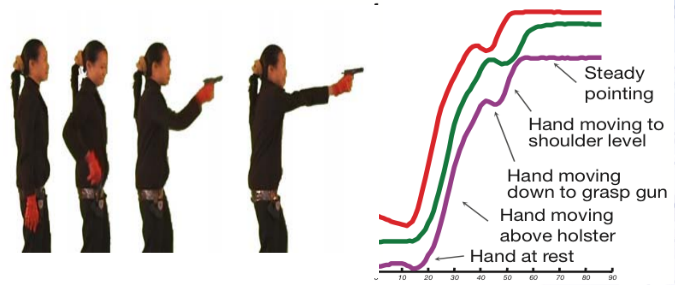


source : Eamonn Keogh ©



Video

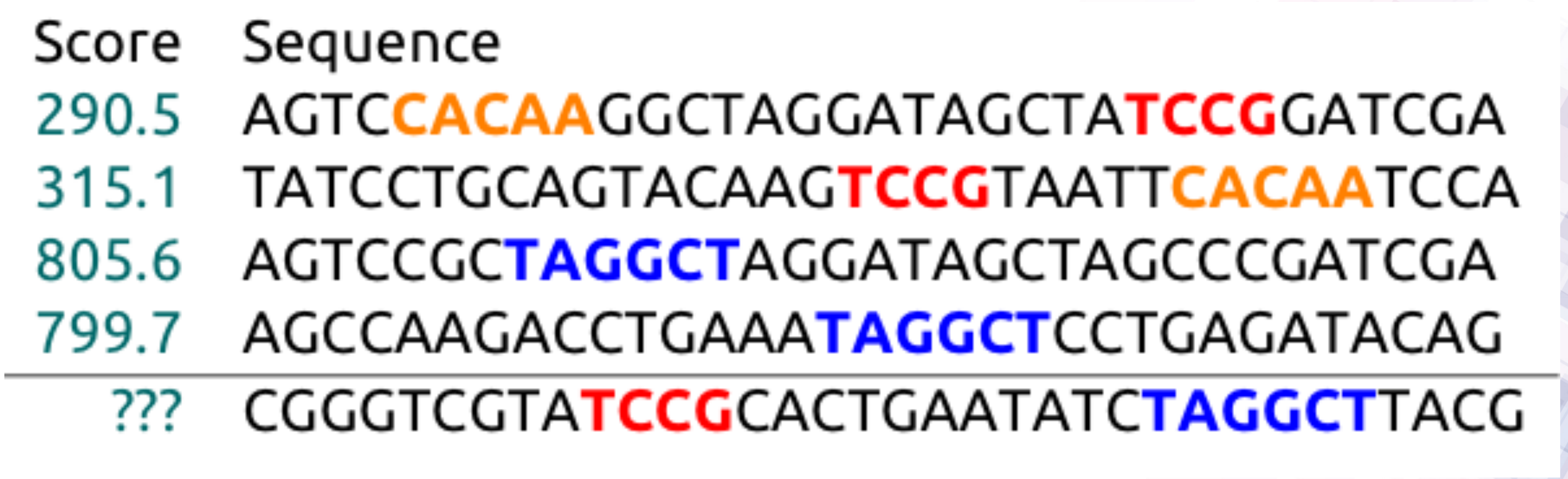
Gun time series annotation



source : Eamonn Keogh ©

Sequence Learning Approaches

Regression or Classification Tasks:



Score	Sequence
290.5	AGTC**CACAA**GGCTAGGATAGCTA**TCCG**GATCGA
315.1	TATCCTGCAGTACAAG**TCCG**TAATT**CACAA**TCCA
805.6	AGTCCGC**TAGGCT**AGGATAGCTAGCCCGATCGA
799.7	AGCCAAGACCTGAAA**TAGGCT**CCTGAGATACAG
???	CGGGTCGTA**TCCG**CACTGAATATC**TAGGCT**TACG

Common Approaches: Pattern-based

- Identify predictive patterns (substrings)
- Transform each sequence into a feature vector
- Use classical learning algorithms

Sequence Learning Approaches

Other Approaches

Distance-Based Methods [1]

- **K-Nearest Neighbours, Support Vector Machines** with string kernels
- Generally outperformed by ensembles and deep learning

Generative Methods [2]

- **Hidden Markov Models**
- Generally outperformed by ensembles and deep learning

Deep Learning [3]

- **Convolutional Neural Networks, Long Short-Term Memory Networks**
- High accuracy, but need large amounts of training data
- Computationally intensive
- Hailed as end-to-end, but actually shift the problem of designing good features to designing good architectures

Sequence Learning Approaches

Pattern-based Methods: mine + learn [4]

Pattern Mining Approaches

Unsupervised Mining: (1) mine patterns, (2) feature selection, (3) classifier

Supervised Mining: (1) mine **discriminative** patterns (bound feature quality via Information Gain, Chi-square Score), (2) classifier

- Brute-force or add constraints when mining patterns (e.g., min support, closed patterns)
- Memory explodes and redundancy in the features

Integrated Approaches (1)

- Combine mining, selection and learning steps into one algorithm
- Start with empty model (no features selected)
- Iterative Algorithm:
  - Search for good features based on the current model
  - Update model
- Overcomes memory/redundancy challenge by selectively exploring features
- **High accuracy achieved with efficient linear models**

Sequence Learning Approaches

Integrated Approaches (2)

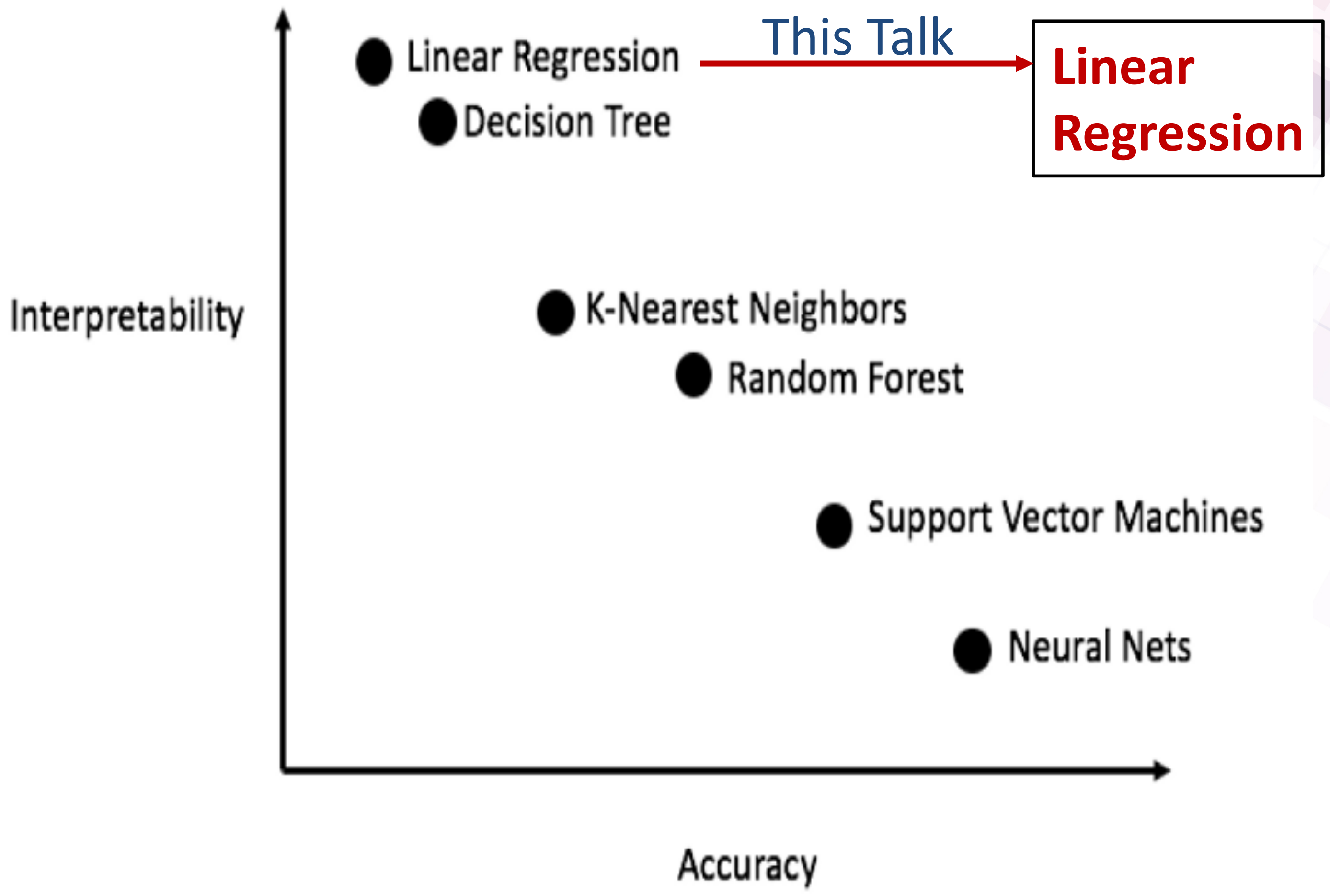
- Many algorithms developed for learning with graphs, trees, sequences, itemsets
- Key Idea: Efficient search for the next feature to update (branch-and-bound on feature quality measure)

- **Two Types of Approaches:**
  - **Boosting-based optimization**: Kudo-Matsumoto bound on gain, e.g., AdaBoost [5], gboost [6]
  - **Direct optimization:** bounding the gradient of the loss function, e.g., Sequence Learner (SEQL) [7], glearn [8]

This talk: direct optimization to train sparse linear models with SEQL-based algorithms [7]

Key Take-Away

Insight
Centre for Data Analytics



Key Take-Away:
Linear Models with Rich Features are

- **Accurate**
- **Efficient**
- **Interpretable**

Key Take-Away

Linear models with rich features are strong competitors to very complex models (e.g., ensembles, deep learning)

Rich features = all subsequences as features; combinations of different representations

Strong competitors = as accurate as much more complex models (e.g., large ensembles, deep learning), but more efficient and easier to interpret

Hints:

Deep net: blow up the model, search for good architecture

Linear model: blow up the feature space, search for good features

Linear Models for Sequences: Methodology

Score	Sequence
290.5	AGTCCACAAGGCTAGGATAGCTATCCGGATCGA
315.1	TATCCTGCAGTACAAGTCCGTAATTCACAATCCA
805.6	AGTCCGCTAGGCTAGGATAGCTAGCCCGATCGA
799.7	AGCCAAGACCTGAAATAGGCTCCTGAGATACAG
???	CGGGTCGTATCCGCACTGAATATCTAGGCTTACG

Linear Model

Weight	*k*-mer
796.6	TAGGCT
402,5	CACAA
-125.3	TCCG

Goal is to learn a mapping:

$$f : S \rightarrow \mathbb{R}$$

Linear model: **f(x) = β^t x**, with **β** the feature weights (the model) and **x** the feature vector

k-mer Feature Space

***k*-mer:** Consecutive sequence of *k* symbols from an alphabet Σ

$<c_i, c_{i+1}, \ldots, c_{i+k-1}>$ with $c_i \in \Sigma$; e.g., for DNA, Σ={A,C,G,T}

All *k*-mers present in the training set are features

Example:

Sample sequence:	GTCCTAATCCTA	
1-mer:	A, C, G, T	(4 possible)
2-mer:	GT, TC, CC, CT, TA, . . .	($4^2 = 16$ possible)
3-mer:	GTC, TCC, CCT, CTA, TAA, . . .	($4^3 = 64$ possible)
⋮	⋮	
8-mer:	GTCCTAAT, TCCTAATC, . . .	($4^8 = 65536$ possible)
⋮	⋮	

Binary feature vector

k-mers	A	C	G	T	AA	AC	...	CA	CC	...	GTCCTA	GTCCTC	...	TTTTTT	...
Binary vector	1	1	1	1	1	0	...	0	1	...	1	0	...	0	...

$\boldsymbol{x}$ and $\boldsymbol{\beta}$ will become huge!

Learning Linear k-mer Models with SEQL

Insight
Centre for Data Analytics

Given:
Training set of labeled examples:

$$\{x_i, y_i\} \text{ for } i = 1, \dots, N$$

$$\mathbf{x_i} \in \mathbb{R}^d \quad \text{with } d = \text{number of features}$$

Goal:
Find $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_d)\,,\ \beta_i \in \mathbb{R}$ by optimizing:

$$\beta^* = \underset{\beta \in \mathbb{R}^d}{\arg\min} \sum_{i=0}^{N} L(y_i, \underline{\beta^T x_i})$$

$f(x_i) = \beta^t x_i$

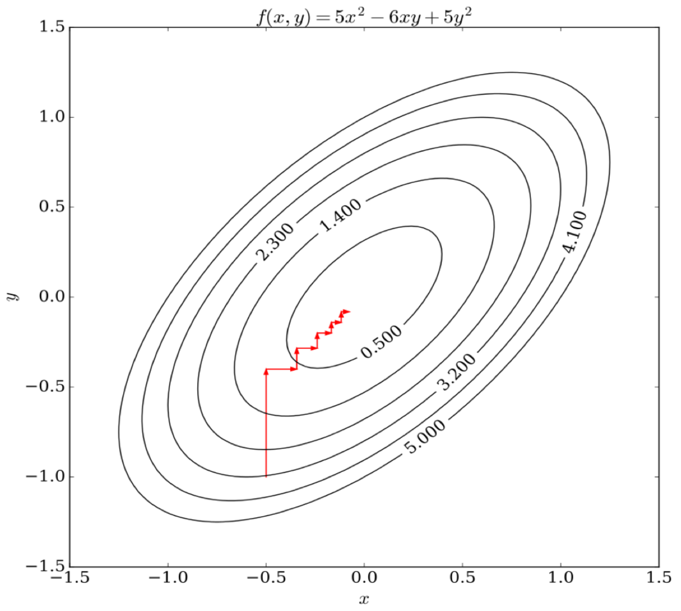
Efficiently learn linear model β with the SEQL algorithm: greedy coordinate descent (Gauss-Southwell selection) [19]

SEQL Algorithm

1. Directly optimize loss function in the feature space of all k-mers
2. Start with empty model and iteratively find/select best feature to update
3. Coordinate-wise bound on gradient restricts the search for best feature

Algorithm 1 Coordinate Descent with Gauss Southwell Selection

```
1: Set $\beta^{(0)} = 0$
2: while termination condition not met do
3:     Calculate objective function $L(\beta^{(t)})$
4:     Find coordinate $j_t$ with maximum gradient value
5:     Find optimal step size $\eta_{j_t}$
6:     Update $\beta^{(t)} = \beta^{(t-1)} - \eta_{j_t} \frac{\partial L}{\partial \beta_{j_t}}(\beta^{(t-1)}) e_{j_t}$
7:     Add corresponding feature to feature set
8: end while
```

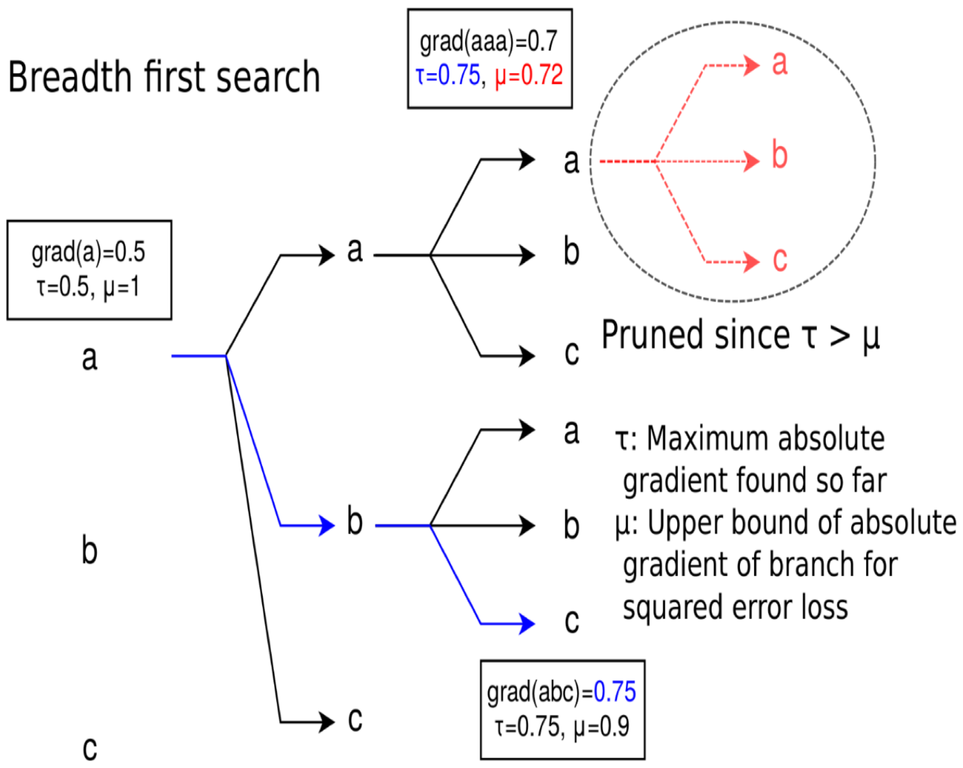


How do we find coordinate j_t efficiently?

SEQL Algorithm

Coordinate Selection in SEQL with Gradient Bound [7]

1. For linear models and many loss functions, the gradient at a k-mer depends on the k-mer occurrence in the training sequences
2. K-mer occurrence is anti-monotonic with k-mer length (i.e., occurrence decreases with increased length k)
3. Bound the gradient at any k-mer using only information about its sub-k-mer occurrence



SEQL-based Algorithms for Sequences

Task and Loss Function:

- **SEQL**: sequence classification; implements logistic loss and squared hinge loss [7]
- **SEQL-SqLoss**: linear regression in all-subsequence feature space [9]
- **SEQLGBM:** general gradient boosting algorithm with SEQL linear models as weak models; any differentiable loss functions; classification and regression [10]

Features:

- **NSEQL/NSEQLGBM:** integrates numeric and all k-mer features to improve the accuracy and speed of training linear models [10]
- **EmbSEQL:** exploits structure in the symbol space to group synonymous symbols, e.g., A(B|C)A; uses word/graph embeddings to define symbol groups and to help with model interpretability [11] (*Severin Gsponer presents at this workshop)

https://github.com/heerme/seql-sequence-learner

https://github.com/svgsponer/seqlgbm

Evaluation on Sequence Classification

Protein Classification Benchmark [12]

Classification Task: Predict if protein is soluble/insoluble given its primary structure (sequence of 21 amino acids).

- Pharma applications: production of insulin

Size:
Training: 69k sequences **(**29k soluble, 40k insoluble)
Test: 2k sequences **(**1k soluble, 1k insoluble)

Sequence length:
Training: avg length: 298.93, max length: 1696, min length: 19
Test: avg length: 296.75; max length: 1697; min length: 34

Evaluation on Sequence Classification

Direct vs Boosting-based Methods

Direct optimization:

- **SEQL:** greedy coordinate-descent [7]
- **glearn:** greedy block-coordinate descent [8]

Boosting-based optimization:

- **SEQLGBM:** gradient boosting machine with SEQL weak models [10]
- **AdaBoost:** boosting with decision stumps and bound on gain [5]
- **gboost:** LPboost with decision stumps and bound on constraint violation [6]

All above algorithms train linear models in all k-mer space.

Method	Accuracy	AUROC	Training time
SEQL	0.675	0.75	5 min
SGBM	0.675	0.75	5 min
AdaBoost	0.672	0.75	15 min
glearn	0.661	0.74	664 min
gboost	0.670	0.74	312 min

- glearn and gboost have scalability issues; lots of fiddling with parameters to get any model at all
- AdaBoost more robust, scalability issue
- SEQL and SGBM fast to train, can run on full benchmark
- All methods have comparable accuracy

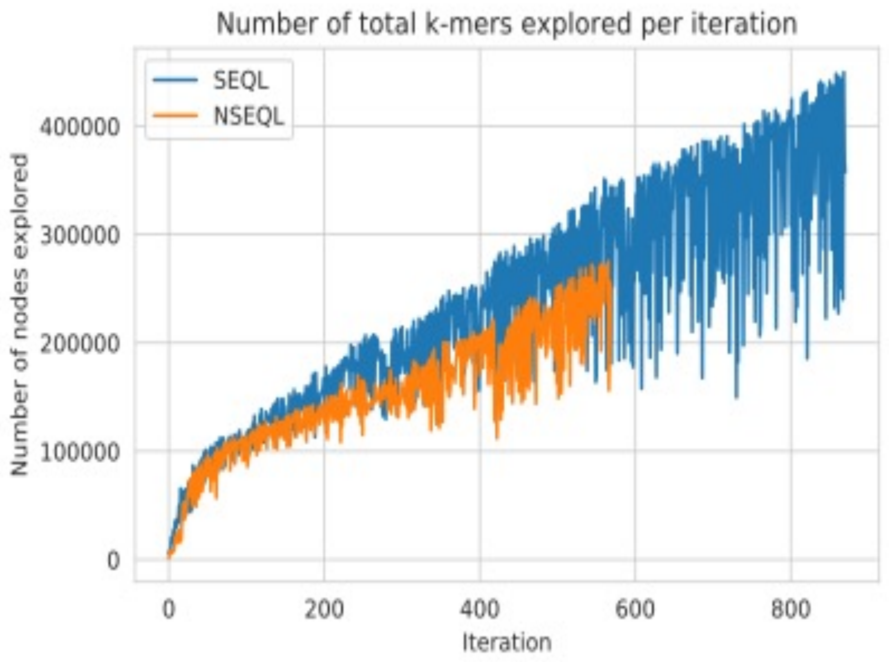
Evaluation on Sequence Classification

Combining Numeric and k-mer Features

- Many tasks have additional domain knowledge captured in numeric features
- NSEQL/NSGBM: First compute the gradient at numeric features to seed the greedy search for the best k-mer
- Faster training, higher accuracy

Algorithm	Accuracy	MCC	Selectivity soluble	Selectivity insoluble	Sensitivity soluble	Sensitivity insoluble
NSGBM	0.735	0.483	0.613	0.689	0.614	0.854
NSEQL	0.735	0.483	0.614	0.690	0.615	0.855
SGBM	0.678	0.365	0.559	0.644	0.560	0.795
SEQL	0.673	0.357	0.544	0.638	0.545	0.800
LOGREG	0.651	0.356	0.399	0.601	0.390	0.916

Method	SEQL	SGBM	NSEQL	NSGBM
Runtime (s)	338	318	284	281
# Features	537	539	(43) 461	(39) 462
# Iterations	871	890	811	750



Evaluation on Sequence Classification

Insight Centre for Data Analytics

Comparing to SOTA for this Benchmark

- **PROSOII:** Stacked logistic regression models; 2-mers as features [13]
- **PaRSnIP:** Gradient Boosted Trees; numeric and up to 3-mer features [14]

- **DeepSol:** Convolutional Neural Network [15]
  - **DeepSol1:** k-mer features (up to k=15)
  - **DeepSol3:** numeric and k-mer features

- **NSEQL and NSGBM:** linear models with numeric and all k-mer features

- Linear models are comparable in accuracy to complex (non-linear) models

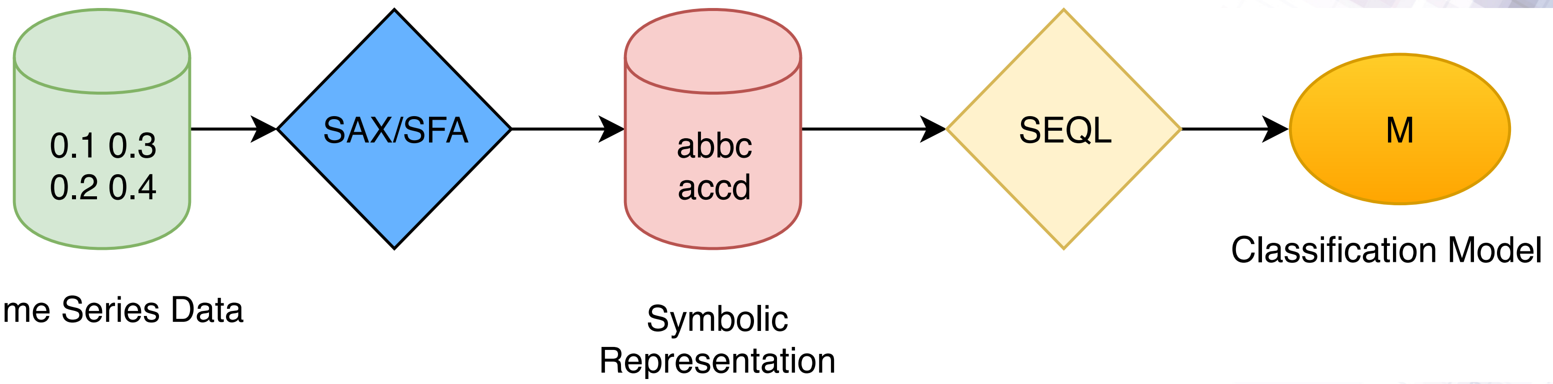
- NSEQL/NSGBM: Train in 5mins on full benchmark

Algorithm	Accuracy	MCC	Selectivity soluble	Selectivity insoluble	Sensitivity soluble	Sensitivity insoluble
DeepSol S3	0.77	0.54	0.81	0.73	0.69	0.84
PaRSnIP	0.74	0.48	0.76	0.72	0.70	0.78
NSGBM	0.73	0.48	0.61	0.69	0.61	0.86
NSEQL	0.73	0.48	0.61	0.69	0.61	0.85
DeepSol S1	0.73	0.46	0.75	0.71	0.69	0.77
PROSO II	0.64	0.34	0.67	0.68	0.69	0.66

Linear Models for Times Series

Time Series Classification with SAXSEQL [16]

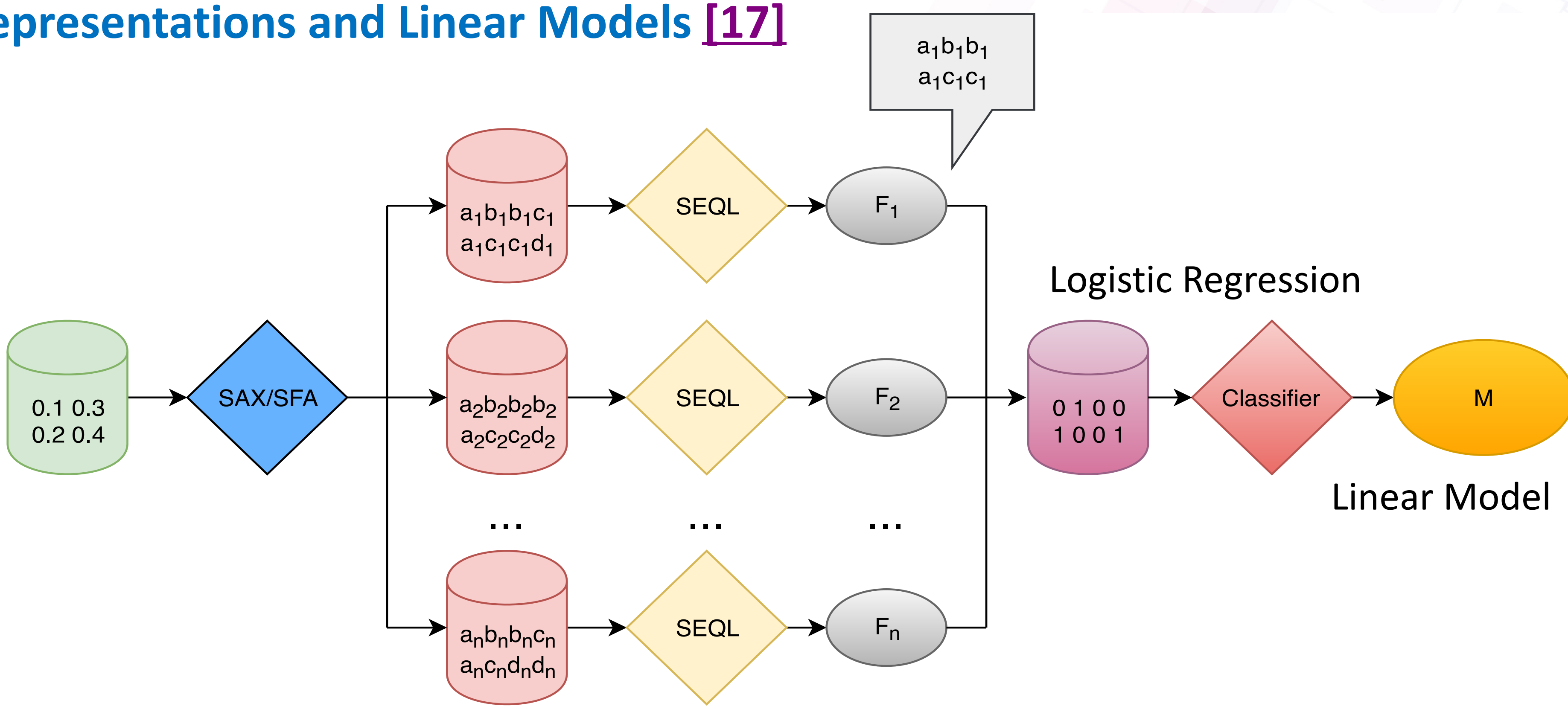
Time Series → Discretisation (SAX, SFA) → Symbolic Sequence → SEQL algorithm



https://github.com/lnthach/SAX-SEQL

MrSEQL: Linear Model with Multiple Representations

Interpretable Time Series Classification with Multiple Symbolic Representations and Linear Models [17]



https://github.com/lnthach/Mr-SEQL

Evaluation on Time Series Classification

Insight
Centre for Data Analytics

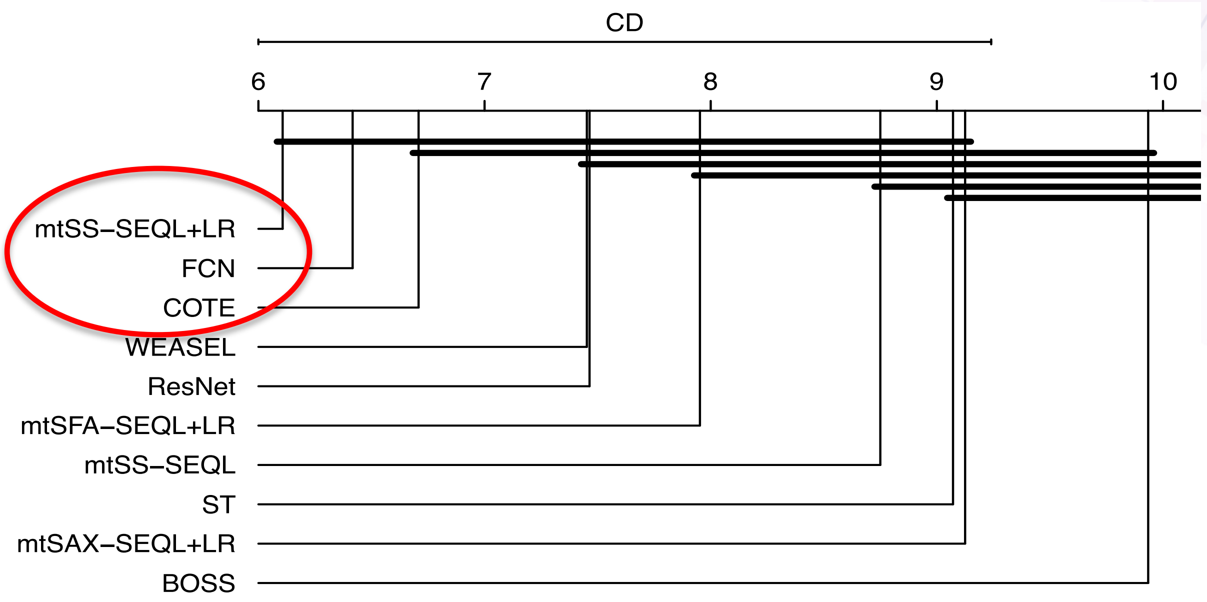
TSC Benchmark: UCR Archive (85 TSC datasets: images, sensors, video)

Avg Ranking of classification algorithms by Accuracy (left is best)

Top-3 models: 1. MrSEQL: **mtSS-SEQL+LR** (our method, a linear model)

2. FCN (deep neural network)

3. COTE (ensemble of 35 classifiers)



Linear Models are Interpretable

Case Study: Human Motion Classification [18]

- Collaboration with Personal Sensing Group@Insight
- Athlete performance testing (**Jump motion**)
- Data collected in Insight lab with wearable sensor (Shimmer3 on foot)

- 3 classes of jumps: Correct technique (**Normal**), Aberrant technique (**Bending**, **Stumble**)
- **Prediction Task:** given the athlete's motion, predict jump class
- **Explanation:** Highlight the time series segments important for the prediction

Interpretability for Time Series Classification

Insight Centre for Data Analytics

Case Study: Human Motion Classification

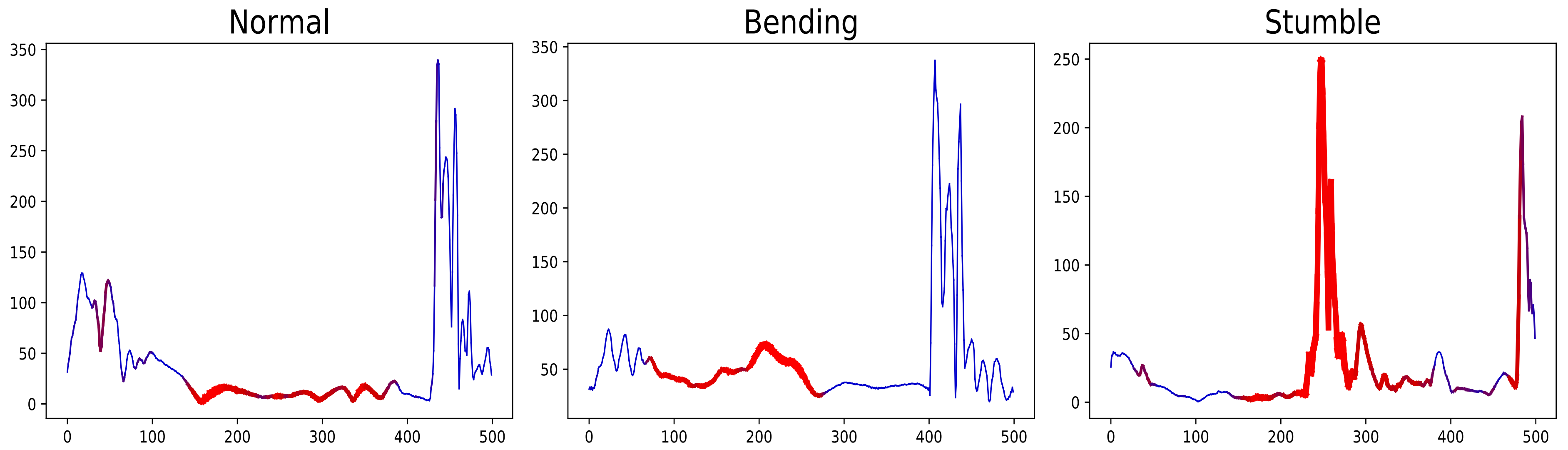
Accuracy: MrSEQL: 97.2% (2min)**,** FCN: 95.5% (2h)

Plot MrSEQL features back to time series

(red segment important for classification)

MrSEQL linear model

l	w	α	Coefficients	Subsequences
42	16	4	0.066	cbaab
53	16	4	0.062	db
53	16	4	0.062	ddddb
42	16	4	0.062	da
31	16	4	0.060	bbbbbbbbbcdddd
53	16	4	−0.054	aaaaaabbbb
20	16	4	−0.054	bbbbaaaaaa
53	16	4	−0.055	bbbcddddd
53	16	4	−0.056	bbbbbbbaaa
53	16	4	−0.061	bbbbbbaaa



Conclusion

Insight
Centre for Data Analytics

Linear Models for Sequences and Time Series are strong contenders vs more complex models (ensembles, deep learning)

- **Accuracy:** Can create rich feature space by combining representations
- **Efficiency:** Can efficiently navigate huge feature space by exploiting its structure to bound feature quality (e.g., nested k-mers)
- **Interpretation:** the learned model is a sparse linear model (a list of weighted features); explicit formula for prediction

Future Work:

- Explore more representations for sequences and time series
- Multivariate sequences and time series
- Resource-constraint learning (on device training/prediction)
- From interpretation to explanation: natural language explanation, focus on discriminative video segment, user studies

Open Resources: Papers, Code, Data

Check out the group's research work here:

Google Scholar:
https://scholar.google.com/citations?hl=en&user=MHs3X9YAAAAJ&view_op=list_works&sortby=pubdate

Research Gate:
https://www.researchgate.net/profile/Georgiana_Ifrim/research

Github:
Github code/data for our work:
https://github.com/heerme?tab=repositories

Acknowledgements

Insight
Centre for Data Analytics

Thank you to co-authors: Severin Gsponer, Thach Le Nguyen, Iulia Ilie, Martin O'Reilly, Luca Costabello, Freddy Lecue

Funding Source: Insight Centre for Data Analytics, Ireland, under Science Foundation Ireland grant number 12/RC/2289 and 12/RC/2289_2

Thank You! Questions?

Insight
Centre for Data Analytics

References

1. M. Ghandi, M. Mohammad-Noori, N. Ghareghani, D. Lee, L. Garraway, and M. A. Beer. gkmSVM: an R package for gapped-kmer SVM. *Bioinformatics*, 32(14):2205–2207, Apr 2016.
2. P. K. Srivastava, D. K. Desai, S. Nandi, and A. M. Lynn. HMM-ModE–improved classification using profile hidden markov models by optimising the discrim-ination threshold and modifying emission probabilities with negative training sequences. *BMC Bioinformatics*, 8(1):104, 2007.
3. B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Bio- technology*, 33(8):831–838, 2015.
4. B. Bringmann, S. Nijssen, and A. Zimmermann. Pattern-based classification: a unifying perspective. *arXiv preprint arXiv:1111.6191*, 2011.
5. T. Kudo and Y. Matsumoto. A boosting algorithm for classification of semi-structured text. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 301–308, 2004.
6. S.Nowozin. Learning with structured data: applications to computer vision. PhD thesis, Berlin Institute of Technology, 2009.

References

7. G. Ifrim and C. Wiuf. Bounded coordinate-descent for biological sequence classification in high dimensional predictor space. *Proceedings of the 17th ACM SIGKDD*, 2011.
8. I. Takigawa and H. Mamitsuka. Generalized sparse learning of linear models over the complete subgraph feature set. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):617–624, 2017.
9. S. Gsponer, B. Smyth, and G. Ifrim. Efficient sequence regression by learning linear models in all-subsequence space. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017.
10. S. Gsponer. Effective Algorithms for Sequence Learning in All-Substring Feature Spaces, PhD Thesis, University College Dublin, 2019.
11. S. Gsponer, L. Costabello, C. Le Van, S. Pai, C. Gueret, G. Ifrim, and F. Lecue. Background knowledge injection for interpretable sequence classification. In *In-ternational Workshop on New Frontiers in Mining Complex Patterns 2019*, 2019.
12. M. T. Weirauch, A. Cote, R. Norel, and M. Annala. Evaluation of methods for modeling transcription factor sequence specificity. *Nature Biotechnology*, 31(2):126–134, 2013.
13. P. Smialowski, G. Doose, P. Torkler, S. Kaufmann, and D. Frishman. PROSO II - A new method for protein solubility prediction. *FEBS Journal*, 279(12), 2012.

References

14. R. Rawi, R. Mall, K. Kunji, C. H. Shen, P. D. Kwong, and G. Y. Chuang. PaRSnIP: Sequence-based protein solubility prediction using gradient boosting machine. *Bioinformatics*, 34(7), 2018.
15. S. Khurana, R. Rawi, K. Kunji, G.-y. Chuang, H. Bensmail, and R. Mall. Deep- sol: A deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 22(16), 2018.
16. T. L. Nguyen, S. Gsponer, and G. Ifrim. Time series classification by sequence learning in all-subsequence space. In *Proceedings of 33rd IEEE International Conference on Data Engeneering (ICDE)*, pages 947–958, 2017.
17. T. L. Nguyen, S. Gsponer, I. Ilie, M. O'Reilly, and G. Ifrim. Interpretable time series classification using linear models and multi-resolution multi-domain sym- bolic representations. *Data Mining and Knowledge Discovery*, 33(4):1183–1222, Jul 2019.
18. T. L. Nguyen, Interpretable Time Series Classification with Symbolic Representations and Efficient Sequence Classifiers, PhD Thesis, University College Dublin, 2019.
19. J. Nutini, M. Schmidt, I. H. Laradji, M. Friedlander, and H. Koepke. Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. *International Conference on Machine Learning*, 37(Section 5):1–34, 2015.