Kernel Methods and Algorithms for General Sequence Analysis

Pavel P. Kuksa

Department of Computer Science
Rutgers University

Qualifying Exam, 2008

Outline

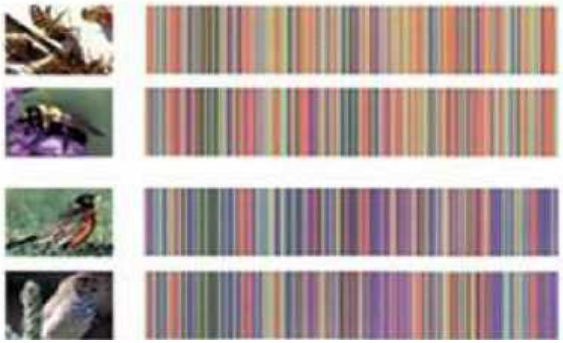
Outline

Classification of Sequential Data using Kernel Methods

- Focus on classification of sequences (text, biological sequences, audio, etc.): inferring functions of proteins, species labels from DNA, music classes or text topic
- Problem Solving framework: Kernel Methods
  - similarity-induced inference
  - problem specific kernel $k(x, y)$
  - learning algorithm that works in a kernel space
- Infer class labels via similarity measures (kernels) $K(x, y)$:
  - measures similarity between two objects (e.g. documents, bio-sequences, etc)
  - has a corresponding vector (dot-product) feature (kernel) space $\phi(x)$
  - the problem is mapped from the original space (may be arbitrary, non-vector) to *vector feature* space
- Main focus is on *string* kernels

DNA Barcoding: Species Identification

- Species are described using short fragment of a DNA



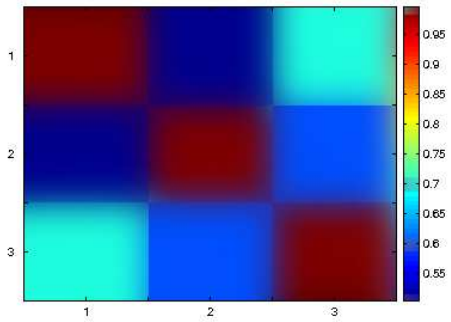
  - track species, biodiversity monitoring, taxonomic research, handheld barcoding devices

- Species Identification problem: predict a taxonomic group (e.g. species) of a given unidentified sample of a DNA

Data

#	Sequence	Species
1	AATTTGAGCAGGACTAATTGGAACTTCTTTAAGATTACTTATTCGAACTGAATTAG GAACCCCAGGATCTTTAATTGGAGATGATCAAATT	A
2	NNNNTTTGAGCAGGACTAATTGGAACTTCTTTAAGATTACTTATTCGAACTGAATT AGGAACCCCAGGATCTTTAATTGGAGATGA	A
3	NNGGAATTTGAGCAGGACTAATTGGAACCTCCTTAAGATTACTTATTCGAACTGAA TTAGGAACCCCAGGATCTTTAATTGGAGATGATCAAATTTATAATACAATTGT	B

Kernel

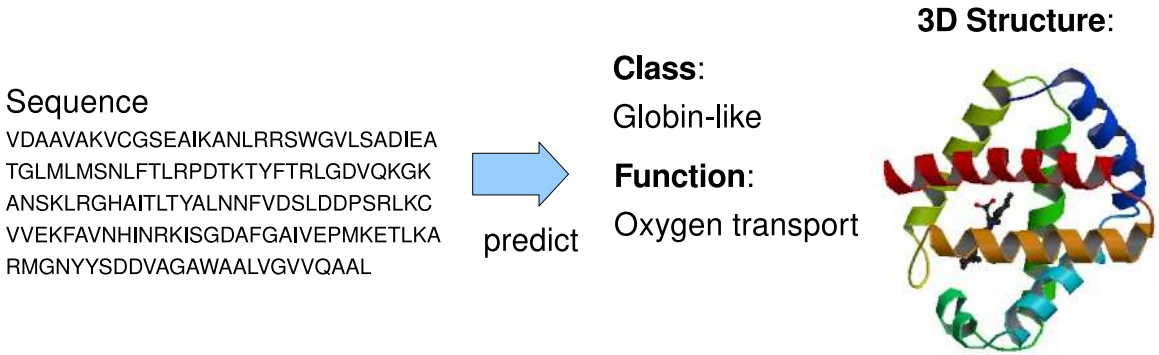


Predictor

Predict (AATTGGAACCTCCTTAAGATTACTTATTCGAA CTGA ATTAGGAACCCCAGGATCTTTAATTGGAGATGATCA AATTTATAATACAATTGT) = { Is **A** / Is **B**

Classification of Protein Sequences

- Proteins: strings of amino acids (primary sequence)
- Primary sequence information easy to obtain
- Structural/Functional class information difficult to obtain
- Classification problem: predict structural/functional class using primary sequence information



Text Classification

- Texts:
  - Bag of words: dot product kernels, polynomial , etc.
  - String of characters: string kernels
  - *String of words*: word sequence kernels
  - Trees: tree kernels
  - etc.
- Classification problem: predict topic (e.g. sport, politics, etc) of a given text

Bio-Sequence Classification

- Bio-Sequences
  - Bag of characters: dot-product kernels, polynomial, etc.
  - String of characters: *string feature-based kernels*
  - Profile (probabilistic generalization of a string): profile-based kernels
  - Realization of a probability distribution: generative models
- Classification problem: predict structural or functional class of a given sequence

Main Results: Summary

- state-of-the-art performance for protein classification
  - supervised
  - semi-supervised
- state-of-the-art performance on the text categorization
- substantially improved accuracy for music classification
- order-of-magnitude faster than many existing approaches
- can work with very large datasets

Outline

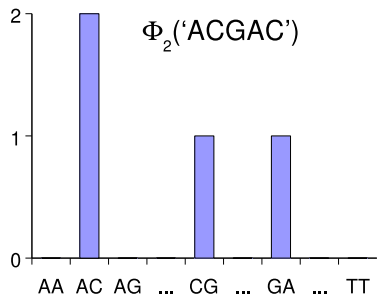
Original String Kernels

- Original string kernels (Lodhi, Haussler) require quadratic $O(n^2)$ in sequence length n time
- Spectrum-like kernel family (Leslie): require only linear in sequence length time
- Best performing kernels have large multiplicative complexity factors (e.g. exponential dependency on the alphabet size and kernel parameters)

Spectrum-like kernels

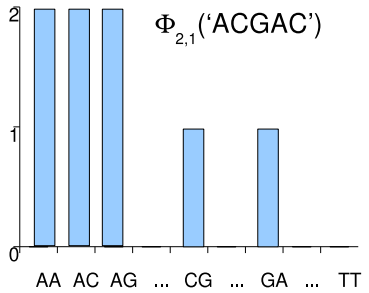
- Measure similarity based on common contiguous fixed-length substrings (k-mers)
- Feature map for *exact spectrum* kernel (no mismatches):

$$\Phi(X) = \left(\sum_{\substack{\alpha \in X \\ |\alpha| = k}} I(\alpha, \gamma) \right)_{\gamma \in \Sigma^k}$$



  - sparse representation: e.g. HKY $\rightarrow$ (00...01(HKY)0...0)
- Feature map for the *mismatch* kernel (m mismatches):

$$\Phi(X) = \left(\sum_{\substack{\alpha \in X \\ |\alpha| = k}} I_m(\alpha, \gamma) \right)_{\gamma \in \Sigma^k}$$



  - less sparse representation (up to $k^m|\Sigma|^m$ more dense): e.g. HKY $\rightarrow$ (01(AKY)...1(HAY)...1(HKA)0...1(HKY)0...0)

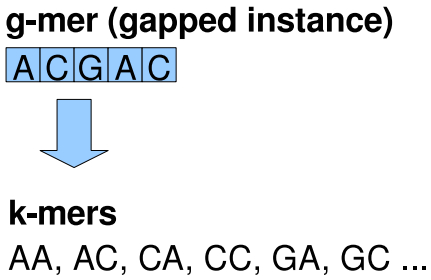
String Spectrum kernels

- Mismatch (k, m)-kernel
  - best performing string kernel for bio-sequences
  - has strong dependency on the alphabet size $O(k^{m+1}|\Sigma|^m n)$
  - high computational complexity
  - the number of mismatches/substitutions m has to be limited to small values (1-2) to control the complexity of the algorithm
  - larger number of allowed mismatches is needed in case of multiple insertions/deletions/substitutions (high computational cost)

Gapped/Subsequence kernels

- Measure similarity based on common subsequences or gapped instances g, $|g| > k$, of k-mers
- Feature map for Gapped kernels:

$$\Phi(X) = \left(\sum_{\substack{g \in X \\ |g|=k+m}} I(\gamma, g) \right)_{\gamma \in \Sigma^k}$$



  - e.g. HHKHHH $\rightarrow$ $(1(HH)0...01(HK)0...01(KH)0...0)$
- Feature map for Subsequence kernels:

$$\Phi(X) = \left(\sum_{\substack{\mathbf{i}=i_1 i_2 ... i_k \\ 1 \leq i_1 < i_2 < ... < i_k \leq n}} I(X(\mathbf{i}), \gamma) \right)_{\gamma \in \Sigma^k}$$

  - e.g. HHKHHH $\rightarrow$ $(10(HH)0...0\ 2(HK)0...0\ 3(KH)0...0)$

Profile kernel: probabilistic string representation

- Replaces string representation (nx1) with a profile (nxΣ)
- Each character in a string is replaced with a probability distribution over alphabet characters
- Profile can be estimated using PSI-BLAST, etc.
- Measure similarity based on common (in probabilistic sense) k-mers
- Profile feature map:

$$\Phi(X) = \left(\sum_{M_\sigma \in X} I(\gamma \in M_\sigma) \right)_{\gamma \in \Sigma^k}$$

$$M_\sigma = \{a : P(a) > \sigma\} \text{(mutation neighborhood)}$$

Summary: String Kernels

- All kernels have the same feature space indexed by all possible contiguous k-mers
- Differences in kernels are mostly in the feature extraction phase, i.e. in the embedding of the sequence:
  - exact contiguous: spectrum kernels
  - inexact contiguous: mismatch, profile kernel
  - exact with gaps: gapped/subsequence kernels
- The position or spatial configuration or correlation of the features in the sequence is not retained
- Performance of the best available methods is still too low for reliable sequence annotation.

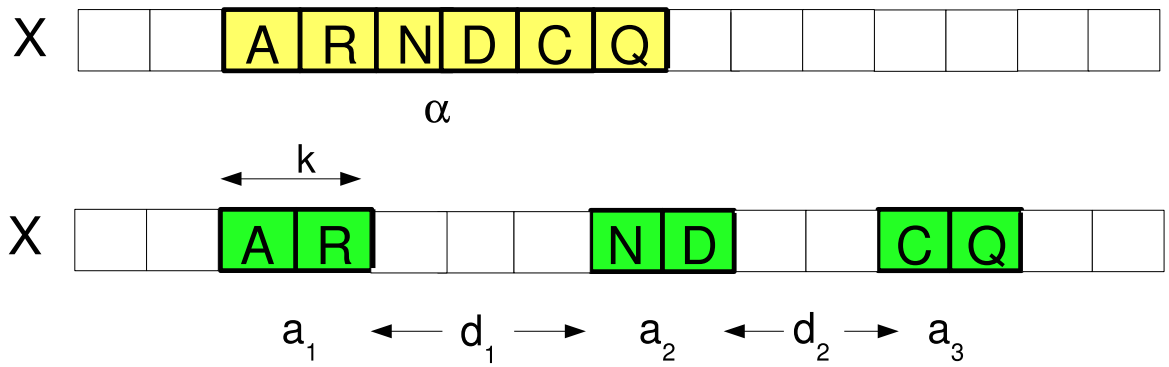
Outline

New Sequence Kernels

- We will define new spatial sample kernels for sequences
  - Inspired by problem of modeling complex processes of sequence transformations (such as evolution)
  - Have time complexity $O(cn)$ linear in the sequence length and alphabet-independent constants
  - Display state-of-the-art performance for protein classification and text categorization tasks

Spatial Sample Kernels

$$K(X, Y|k, t, d) = \sum_{\substack{(a_1,d_1,a_2,\ldots,d_{t-1},a_t),\\ a_i \in \Sigma^k, 0 \leq d_i \leq d-1}} C(a_1, d_1, \ldots, d_{t-1}, a_t|X) C(a_1, d_1, \ldots, d_{t-1}, a_t|Y)$$



Contiguous k-mer feature α of a traditional spectrum/mismatch kernel (top) contrasted with the spatial sample features (bottom).

- Notation:
  - $t = 2$: double-(k,d)
  - $t = 3$: triple-(k,d)

Features Induced by Kernels

S = HKYNQLIM

spectrum-5
HKYNQ
KYNQL
YNQLI
NQLIM

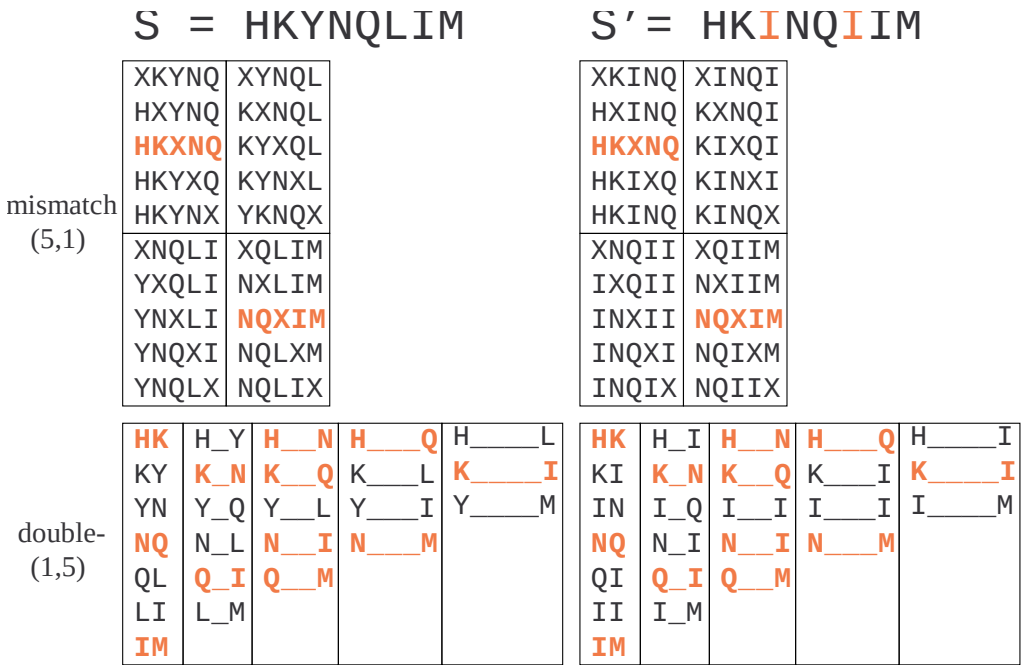
mismatch(5,1)	
XKYNQ	XYNQL
HXYNQ	KXNQL
HKXNQ	KYXQL
HKYXQ	KYNXL
HKYNX	YKNQX
XNQLI	XQLIM
YXQLI	NXLIM
YNXLI	NQXIM
YNQXI	NQLXM
YNQLX	NQLIX

double-(1,5)				
HK	H_Y	H__N	H___Q	H____L
KY	K_N	K__Q	K___L	K____I
YN	Y_Q	Y__L	Y___I	Y____M
NQ	N_L	N__I	N___M	
QL	Q_I	Q__M		
LI	L_M			
IM				

Comparison of features extracted by the spectrum-like and spatial kernels. In the mismatch representation, each feature basis with 'X' corresponds to $|\Sigma|$ features.

- spatial kernels: low-dimensional, sparse, few features
- mismatch kernels: highly-dimensional, very many features

Sequence Modeling



Differences in handling substitutions by spectrum-like and spatial kernels

- Mismatch: very few features retained
- Spatial kernels: still significant overlap in sequence features
- For the spectrum-like kernel, the number of allowed mismatches needs to be increased (at a high computational cost)

Computing Spatial Sample Kernels

- Can be efficiently computed using Sorting and Counting

Algorithm 1. Spatial Sample Kernel

Input: set of strings $S = \{s_1, \ldots, s_N\}$, parameters t (number of sequence elements sampled), and d (maximum distance)
Repeat for every $(d_1, ..., d_t), 0 \leq d_i \leq d - 1$
1: **Extract** t-tuples of features from each string $s \in S$
2: **Sort** t-tuples using t rounds of counting sort
3: **Scan** the sorted list and for each distinct feature f
update the kernel matrix K for all strings containing f
Output: kernel matrix K

Semi-Supervised Spatial Sample Kernel

- Semi-Supervised Learning
  - Small amount of training labeled sequences under supervised learning: low performance
  - Can improve by enlarging training set with unlabeled sequences that are similar (homologous) to the labeled sequences
- Semi-Supervised Spatial Sample Kernel
  - Define a kernel over **string sets** $N(X)$ (instead of strings X)

$$K(X, Y) = \sum_{x \in N(X)} \sum_{y \in N(Y)} K(x, y)$$

  - String set $N(X)$ is formed by homologous (similar in the sense of some measure) sequences

Computing Spatial Sample Kernels over String Sets

Algorithm 2. Spatial Sample Neighborhood Kernel

Input: set of *string sets* $N(S) = \{N(s_1), ...N(s_N)\}$, where $N(s_i)$ is a set of neighbors for s_i, parameters t, d

Repeat for every $d_1, ..., d_t \in \{1, ..., d\}$

1: *Extract* t-tuples of features from each input set $N(s) \in N(S)$

2: *Sort* t-tuples using t rounds of counting sort

3: *Scan* the sorted list and for each distinct feature f update the kernel matrix K for all string groups containing f

Output: kernel matrix K

Complexity Comparison

Method	Time complexity	Running time (s)
Triple kernel	$O(d^2nN + d^2\|\Sigma\|^3N^2)$	112
Double kernel	$O(dnN + d\|\Sigma\|^2N^2)$	54
Mismatch	$O(k^{m+1}\|\Sigma\|^m nN + \|\Sigma\|^k N^2)$	948
Gapped kernel	$O(\binom{g}{k} gnN + \|\Sigma\|^k N^2)$	-
Neighborhood kernels		
Triple kernel	$O(d^2HnN + d^2\|\Sigma\|^3N^2)$	327
Double kernel	$O(dHnN + d\|\Sigma\|^2N^2)$	67
Profile kernel	$O(kM_\sigma nN + \|\Sigma\|^k N^2)$	10 hours
Mismatch	$O(k^{m+1}\|\Sigma\|^m HnN + \|\Sigma^k\|N^2)$	74540

Notations used in the table: N-number of sequences, n-sequence length,
H is the sequence neighborhood size, $|\Sigma|$ is the alphabet size
k, m are mismatch kernel parameters ($k = 5, 6$ and $m = 1, 2$ in most cases)
M_σ is the profile neighborhood size, $k^m|\Sigma|^m \leq M_\sigma \leq |\Sigma^k|$

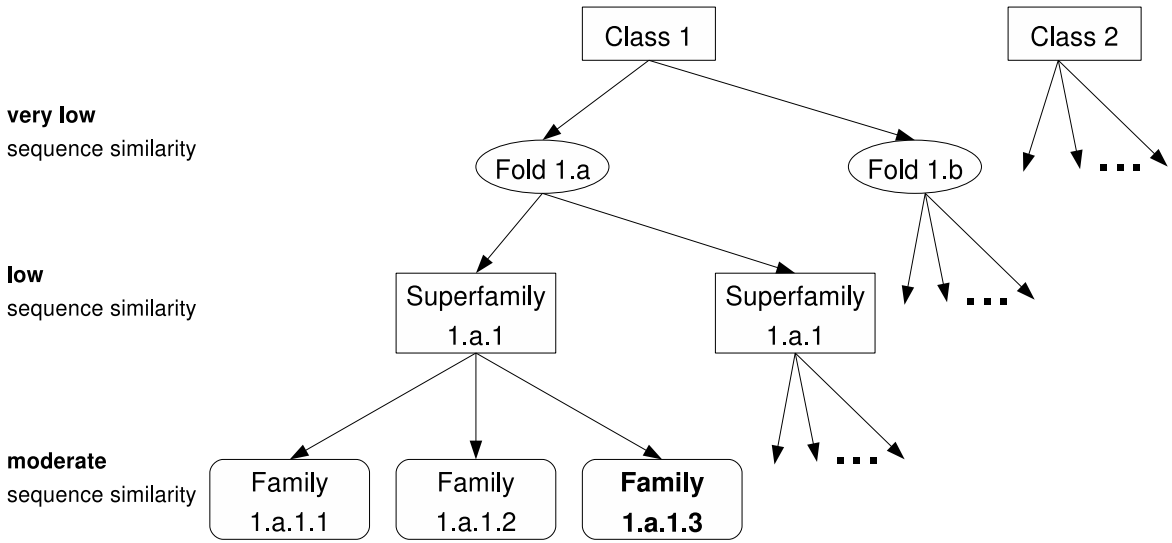
Spatial Kernels: Summary

- variable length pattern
- explicitly model spatial configuration of features in the sequence
- efficiently model complex sequence transformations (mutations, insertions, deletions, etc)
- captures sequence content at multiple scales
- fast $O(cn)$ evaluation
- alphabet-free matching (alphabet size independent)

Outline

Experiments: Remote Protein Homology Prediction

- Protein *remote* homology prediction (recognition of previously unseen families)
  - We address this problem in the context of Structural Classification of Proteins (SCOP)



  - Challenges:
    - limited labeled examples
    - highly dissimilar sequences
  - Experiments are designed to test whether the method can detect a new (unseen) family from a given superfamily

SCOP

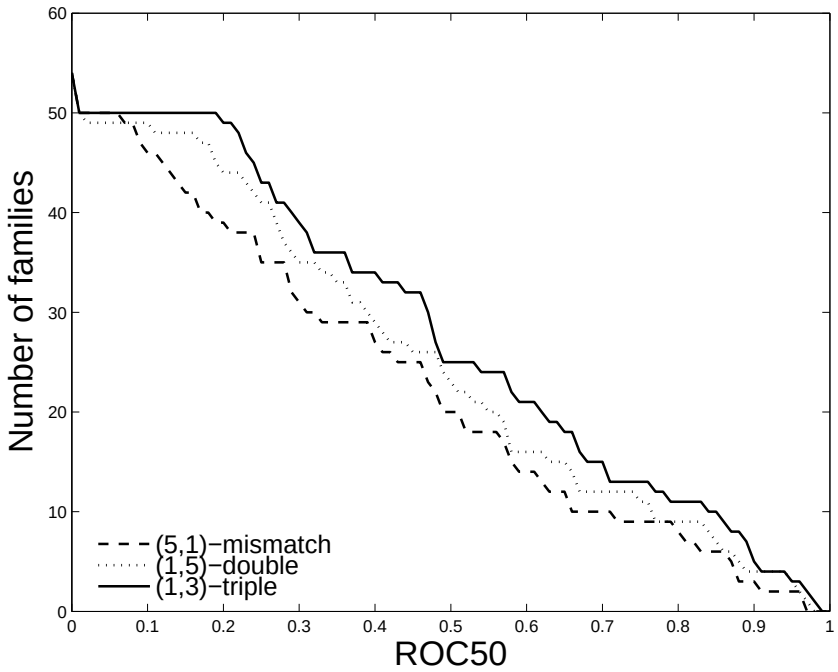
- Supervised Setting (2862 labeled sequences):

Method	ROC	ROC50	# dim.
(5, 1)-mismatch	0.8749	0.4167	3200000
SVM-pairwise	0.8930	0.4340	-
gapped(6,2)	0.8296	0.3316	400
gapped(7,3)	0.8540	0.3953	8000
(1,5) double	0.8901	0.4629	2000
(1,3) triple	**0.9148**	**0.5118**	72000

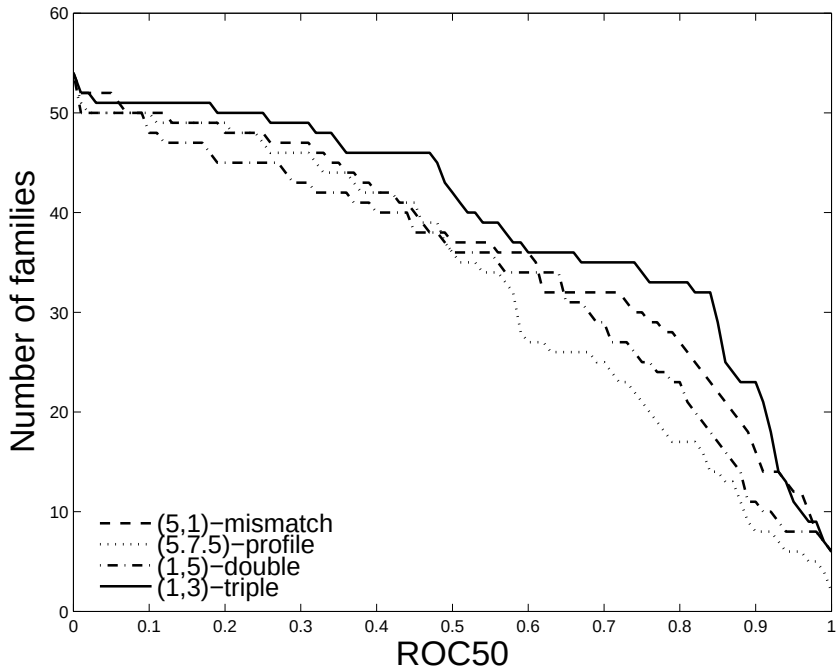
- Semi-Supervised Setting (labeled+unlabeled, 7329 total)

Method	ROC	ROC50
(5, 1)-mismatch neighborhood	0.9093	0.6745
(5,7.5)-profile	0.9190	0.6069
(1,5)-double neighborhood	0.9282	0.6383
(1,3)-triple neighborhood	**0.9382**	**0.7262**

SCOP: ROC



(a) SCOP dataset. Supervised setting.

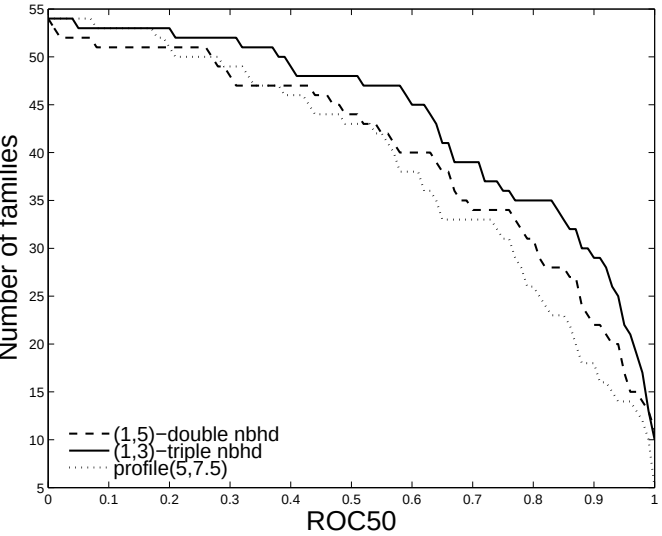


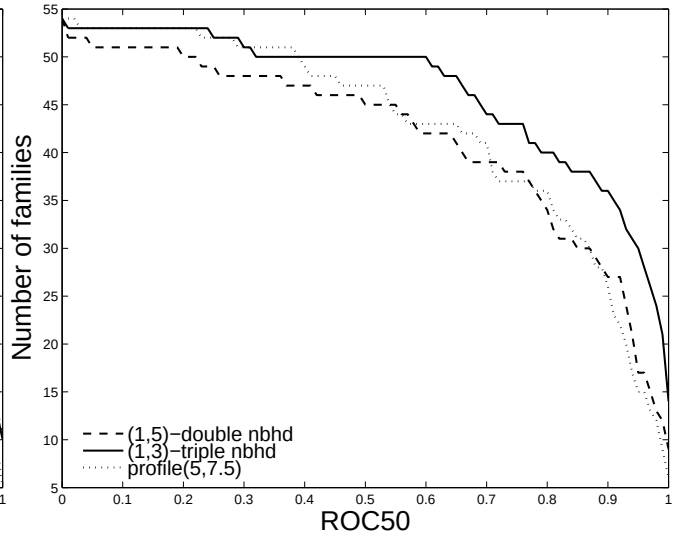
(b) SCOP dataset. Semi-supervised setting

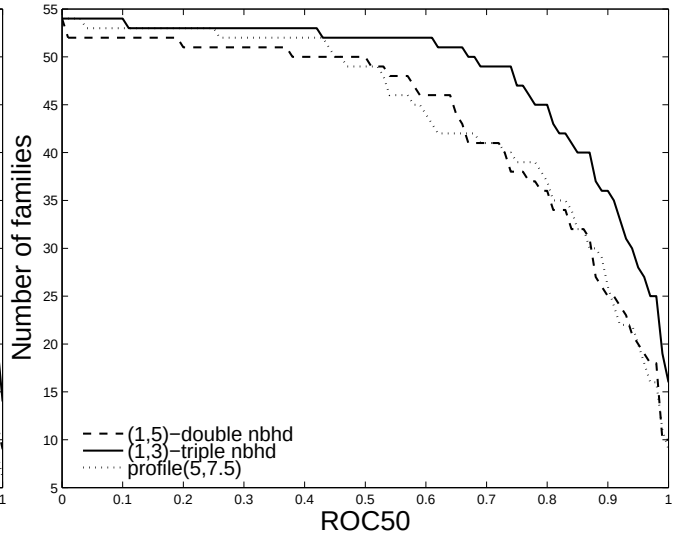
Large-Scale Protein Remote Homology Detection

Dataset	# Seq	# Neighbors (mean/median/max)
Scop	4,467	11/2/465
Swiss-Prot	101,602	56/28.5/385
PDB	17,232	16/5/334
NR	534,936	114/86/490

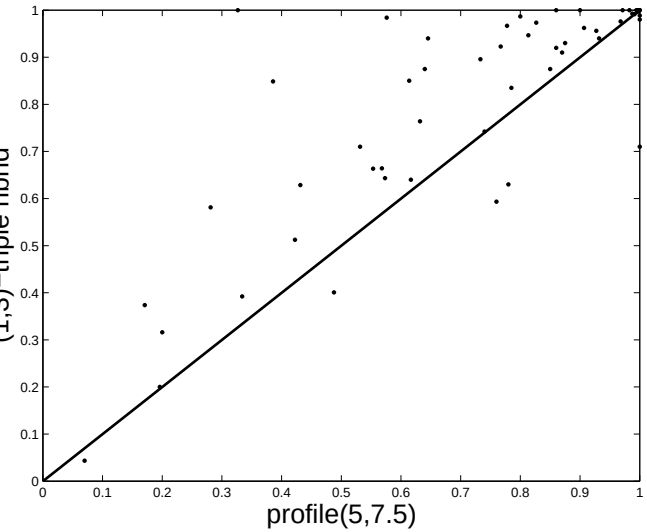
Large-Scale Protein Remote Homology Prediction



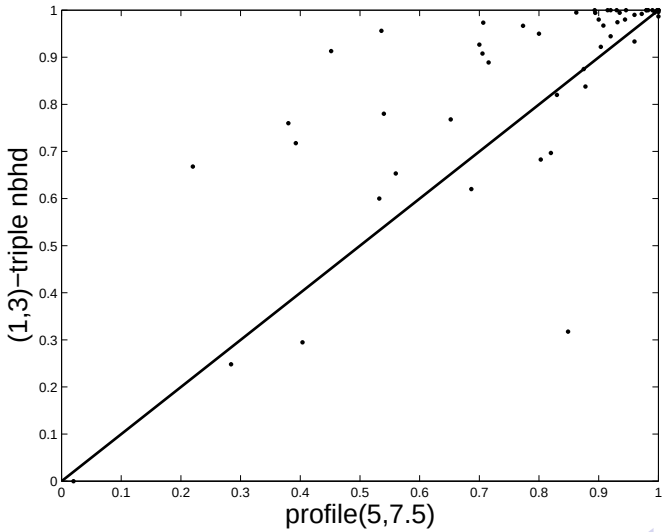




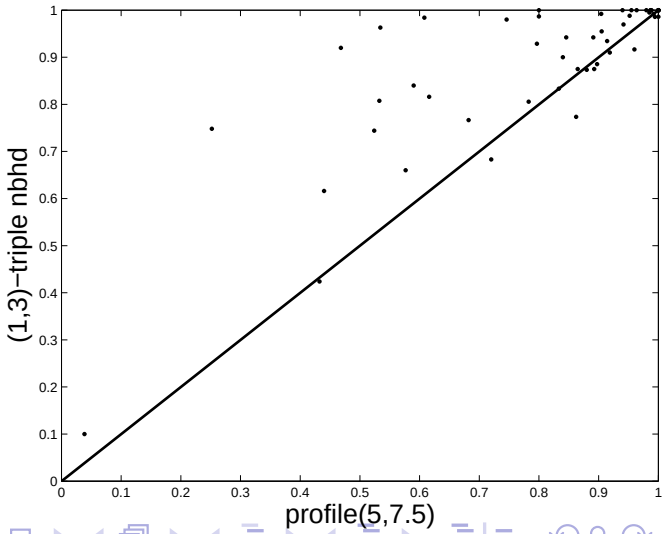
(c) PDB



(d) Swiss-Prot



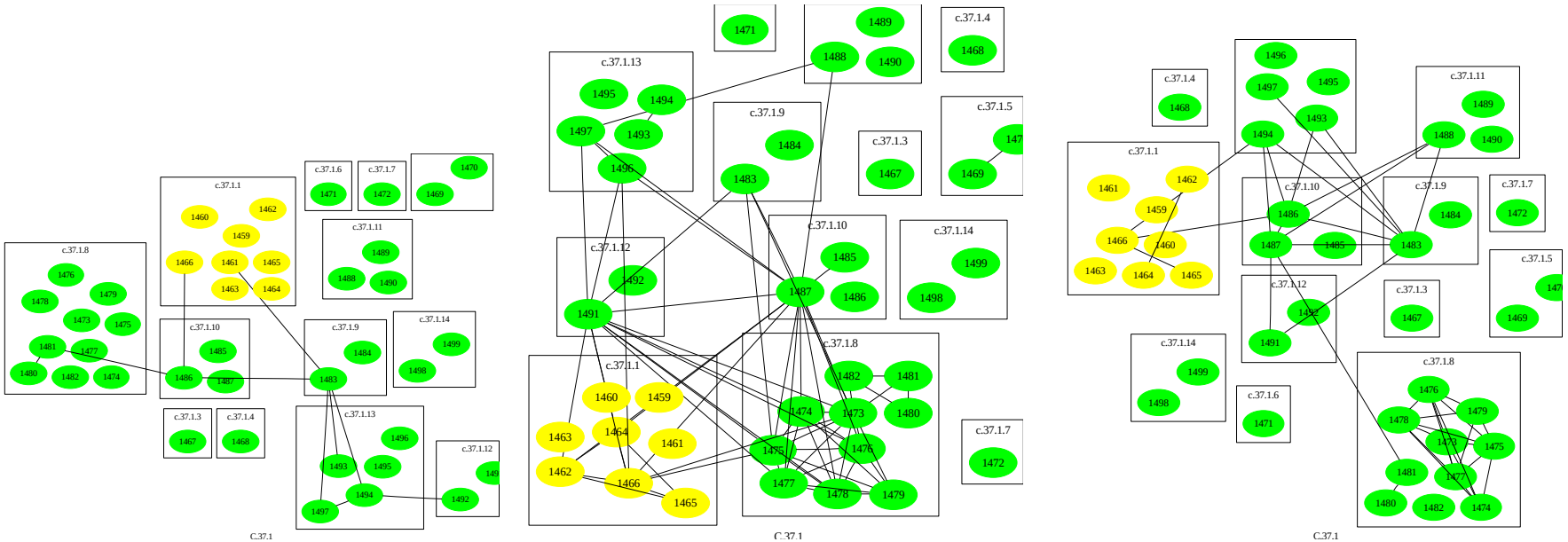
(e) NR



Large-Scale Protein Remote Homology Detection

PDB	ROC	ROC50
double-(1,5) neighborhood	.9599	.7466
triple-(1,3) neighborhood	**.9691**	**.8083**
profile(5,7.5)	.9511	.7205
Swiss-Prot		
double-(1,5) neighborhood	.9582	.7701
triple-(1,3) neighborhood	**.9732**	**.8605**
profile(5,7.5)	.9709	.7914
mismatch neighborhood†	.955	.810
NR		
double-(1,5) neighborhood	.9720	.8076
triple-(1,3) neighborhood	**.9861**	**.8944**
profile(5,7.5)-2 iterations	.9734	.8151
profile(5,7.5)-5 iterations	.984	.874
profile(5,7.5)-5 iter. with secondary structure	.989	.883

Kernel-Induced Manifolds



(i) Supervised Triple (j) Semi-Supervised Triple (k) Semi-Supervised Profile

- Each cluster (box) represents a family in the superfamily (c.37.1)
- Green (darker) and yellow (lighter) nodes are train and test sequences, respectively
- Stronger connectivity in case of the triple kernel

Text Classification

- Reuters Dataset (benchmark dataset for text categorization)
- 8983 documents (top categories)
- Alphabet Size: **29224**
- all documents are converted to strings over $\{1...29224\}$ alphabet after removing stop-words and stemming
- Text examples:

Computer Terminal Systems Inc said it has completed the sale of 200,000 shares of its common stock, and warrants ...

AM International Inc, reporting an operating loss for the January 31 second quarter, said prospects for the balance of the fiscal year remain good.

Text Classification (F1-scores)

Class	TF-IDF	KSG	Double(1,5)
Acq	97.11	96.8	**97.72**
Crude	87.71	89.4	**90.11**
Earn	98.70	98.3	**99.02**
Grain	93.29	92.5	90.18
Interest	71.80	81.5	**81.63**
Money	77.61	84.0	82.66
Ship	72.97	81.9	**85.56**
Trade	84.26	90.2	**93.47**
Mean (macro-average)	85.43	89.33	**90.05**
Mean (micro-average)	93.18	93.9	**94.51**

KSG=key-substring-group features [4] (state-of-the-art method)

Text Classification (Double-(1,5))

Class	Accuracy	Precision	Recall	Specificity
Acq	98.62	98.72	96.38	99.50
Crude	98.58	92.14	88.17	99.40
Earn	98.97	98.53	99.07	98.90
Grain	99.33	97.64	83.78	99.87
Interest	97.91	91.49	65.65	99.87
Money	97.63	85.63	79.89	98.98
Ship	98.97	82.80	88.51	99.35
Trade	99.09	89.08	91.38	99.46
Mean (macro-average)	98.54	92.00	86.60	99.39
Mean (micro-average)	98.59	95.98	93.08	99.42

Music Genre Classification

- 1000 audio sequences (each $\approx$ 6500 long)
- 10 Genres
- Alphabet size: **1024** (MFCC features)
- 10-fold cross-validation
- Binary and Multi-Class settings

Music Sample 1

Music Sample 2

Music Genre Classification: Binary

#	Genre	DWCH†	Double (MFCC)
1	Blues	95.49 (1.27)	93.6 (4.77)
2	Classical	98.89 (1.1)	95.6 (1.35)
3	Country	94.29 (2.49)	94.3 (2.21)
4	Disco	92.69 (2.54)	**94.3 (1.41)**
5	Jazz	97.9 (0.99)	95.5 (2.27)
6	Metal	95.29 (2.18)	94.7 (1.42)
7	Pop	95.8 (1.69)	**96.2 (1.75)**
8	Hiphop	96.49 (1.28)	**97.1 (0.99)**
9	Reggae	92.3 (2.49)	**95.5 (1.58)**
10	Rock	91.29 (2.96)	**95.1 (1.66)**

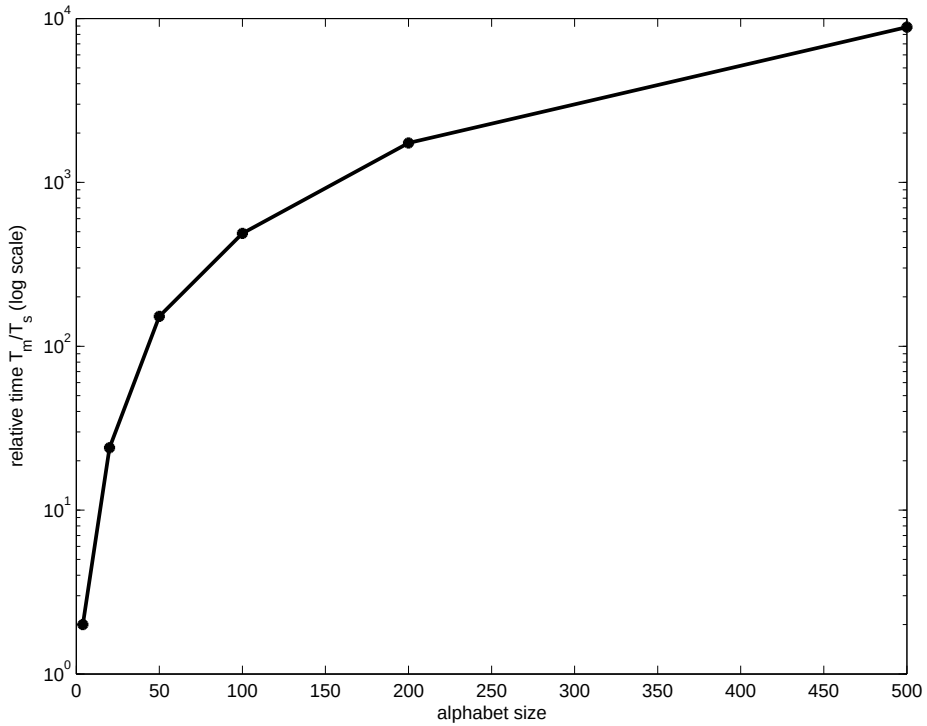
†: DWCH=Daubechies Wavelet Coefficient Histograms [2]

Music Genre Classification: Multi-Class

Method	Accuracy	Top 3 Accuracy
MFCC	53.7 (4.11)	
Double-(1,5) (MFCC)	67.6 (4.17)	86.6 (1.45)

Running Time Analysis: Synthetic data ($n = 10,000$)

alphabet size	4	20	50	100	200	500	1000
mismatch	0.1	1	5	17	58	295	1029
triple	0.03						



Relative running time ratio ($T_{\text{mismatch}}/T_{\text{triple}}$) as a function of the alphabet size.

Experiments: DNA Barcoding

- Classification of DNA barcodes
  - multi-class
  - 10-fold cross-validation

Dataset	Mismatch		Triple	
	Error(%)	F1	Error(%)	F1
Astraptes (12 classes)	1.27 ± 1.46	82.94	1.06 ± 1.52	**90.33**
Bats of Guyana (96 classes)	1.62 ± 1.19	82.94	1.77 ± 1.26	**88.92**
Fish larvae (7 classes)	25.00 ± 30.43	56.46	4.00 ± 8.43	**64.99**

Running Time Analysis

Comparison with other methods, time (s)

method	supervised	semi-supervised
triple	112s	327s
double	54s	67s
mismatch	948s	74540s
profile	-	36000s

Spatial kernels, running time (s)

Dataset	Alphabet size	# Seq.	Seq. length	Double, (s)	Triple, (s)
Reuters	29,224	8983	70	76	97
Music genre	1024	1000	6895	99	141
SCOP	20	7329	172	54	112
Swiss-Prot	20	101,602	332	64	276
NR	20	534,936	367	81	587

Summary

- Fast, Alphabet-independent matching
- Significantly improved classification performance and sensitivity for protein remote homology prediction
- State-of-the-art performance in text, music classification
- General approach, can be readily extended to other domains
- Outlook
  - Beyond classification: algorithms for ranking, clustering
  - Extension to Non-sequential domains: e.g. images
  - Efficient Feature selection Algorithms
  - Approximate Matching Algorithms (e.g. approximate distance matching)
  - New efficient string algorithms
  - New applications: fold prediction, using higher order information (e.g. secondary structure), semi-supervised methods for text, etc.

References

R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie.

Profile-based string kernels for remote homology detection and motif extraction.
J Bioinform Comput Biol, 3(3):527–550, June 2005.

T. Li, M. Ogihara, and Q. Li.

A comparative study on content-based music genre classification.
In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 282–289, New York, NY, USA, 2003. ACM.

J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble.

Semi-supervised protein classification using cluster kernels.
Bioinformatics, 21(15):3241–3247, 2005.

D. Zhang and W. S. Lee.

Extracting key-substring-group features for text classification.
In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 474–483, New York, NY, USA, 2006. ACM.

Publications I


Pavel Kuksa, Vladimir Pavlovic
Fast Kernel Methods for SVM Sequence classifiers
Algorithms in Bioinformatics, 2007


Pavel Kuksa, Pai-Hsi Huang, Vladimir Pavlovic
Fast and Accurate Multiclass Fold Prediction with Spatial Sample Kernels
Computational Systems Bioinformatics, 2008


Pavel Kuksa, Pai-Hsi Huang, Vladimir Pavlovic
Spatially-constrained Sample Kernel for Sequence Classification
Learning Workshop, 2008


Pavel Kuksa, Pai-Hsi Huang, Vladimir Pavlovic
Fast Protein Homology Detection with Sparse Spatial Sample Kernels
journal article in preparation