Improved Functional Prediction of Proteins by Learning Kernel Combinations in Multilabel Settings

Volker Roth, Bernd Fischer
ETH Zurich, Institute of Computational Science
Overview

- Genes/proteins characterized by different measurements → kernels for combining heterogeneous data.

- Proteins can have multiple functions → multilabel classifier which learns kernel combinations

- MIPS functional catalogue: hierarchy of protein functions → kernelHMM: hierarchical multilabel classifier.
Goal: Data-fusion, “Common Language”  \(\sim\) Mercer Kernels
Mercer Kernels

- **Support Vector Machines**: Non-linear learning machines

- “Classical View”: vectorial data.

Mapping from **input space** to **feature space**

\[ \phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \]

\[ (x_1, x_2) \mapsto (z_1, z_2, z_3) \]

\[ = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \]

\[ k(x, x') = \phi(x) \cdot \phi(x') \]

\[ = (x \cdot x')^2 \]

[Schölkopf, 2000]
Non-vectorial Data

• Biinformatics: many **non-vectorial** data-types:
  
  – Interaction graphs
  
  – Phylogenetic trees
  
  – Strings **GSAQVKGHGKKVADALTNAVAHV**

• **Observation:** each $n \times n$ kernel matrix contains dot products $\rightarrow$ pairwise similarities

• **Data fusion:** Data of each type are converted into a kernel $\Rightarrow$ “common language” for heterogeneous data.
RBF-Kernels for Expression Data

- **For each gene:** vector of expression values under different experimental conditions

- “classical” RBF-Kernel: \( k(x_1, x_2) = \exp\left(-\sigma \|x_1 - x_2\|^2\right) \)
Diffusion-Kernels for Interaction Graphs

- \( A \): Adjacency matrix, \( D \): node degrees, \( L = D - A \).

- \( K := \frac{1}{Z(\beta)} \exp(-\beta L) \) with transition probability \( \beta \).

- **Physical interpretation (random walk):** next node randomly picked among neighboring nodes.

- Self-transition probabilities: \( 1 - \frac{d_i}{\beta} \)

- \( K_{ij} \): Probability to walk from \( i \) to \( j \).  

(Kondor and Lafferty, ICML 2002)
Alignment kernels for Sequences

Sequence alignment via Pair HMMs ⇒ Mercer kernel (Watkins, 2000)
Combination of heterogeneous Data

Addition of kernels ⇒ new kernel:

\[ k_1(x, y) = \phi_1(x) \cdot \phi_1(y), \quad \Rightarrow k' = k_1 + k_2 = (\phi_1(x)) \cdot (\phi_1(y)) \]

\[ k_2(x, y) = \phi_2(x) \cdot \phi_2(y) \]

⇒ learn weighted combination of \( m \) kernels:

\[ K = \alpha_1 K_1 + \alpha_2 K_2 + \alpha_3 K_3 + \alpha_4 K_4 \]
Functional Classification of Yeast Proteins

- **Goal:** learn classifier that assigns each yeast protein to one or several functional classes
  \[\Rightarrow\] hierarchical learning problem with multiple labels

- Yeast has \(\approx 6000\) genes, for \(\approx 3500\) the function(s) is/are “known”.

### Functional Classification:

- **MIPS comprehensive yeast genome database** [www.mips.gsf.de](http://www.mips.gsf.de)

| Category | Category |
|----------|----------|
| 1        | metabolism | 10 | cellular transport |
| 2        | energy     | 11 | cellular communication |
| 3        | storage    | 12 | cell rescue, defense |
| 4        | cell cycle and DNA processing | 13 | interaction with cellular environment |
| 5        | transcription | 14 | interaction with environment(systemic) |
| 6        | protein synthesis | 15 | transposable elements |
| 7        | protein fate | 16 | cell fate |
| 8        | protein with binding function | 17 | development |
| 9        | protein activity regulation | 18 | biogenesis of cellular components |
Designing the Classifier: Wishlist

- **Data-fusion:** sparse combinations of kernel matrices.

- **Multiple protein functions:** consistent handling of multiple classes with **multilabels**;

- **Efficiency:** many objects, many kernels, out-of-core kernel matrices...

- **Structured outputs:** consistent handling of hierarchical multilabels.
Designing the Classifier: NKDA

- Method: multilabel version of **Nonlinear Kernel Discriminant Analysis** with built-in feature selection \( \mapsto \) subset of optimal kernel weights.
Designing the Classifier: multilabels

- Unsupervised LDA: **Mixture DA** [Hastie 96].
  
  **Observation:** LDA finds maximum likelihood solution for Gaussian mixture model with “pooled” covariance

- Unsupervised NKDA $\rightsquigarrow$ Gaussian mixtures in kernel space $\rightsquigarrow$ EM algorithm
Convex Kernel Combinations: Modified EM

E-step:
- DA as linear regression
- ARD priors: $p(\beta_k | \omega) = \prod_i \phi(0, \omega_i^{-1})$
- Inference: $\frac{1}{d} \sum_{i=1}^d \frac{1}{\omega_i} = \frac{1}{\lambda}, \quad \omega_i > 0$
- New variables:
  - $\gamma_{j,i} = \sqrt{\omega_i/\lambda} \beta_{j,i}$
  - $c_i = \sqrt{\lambda/\omega_i}$
- Weight sharing: $c = (c_1, \ldots, c_1, \ldots, c_J, \ldots, c_J)^T$

M-step:
- Relevance Parameters
- Ridge regression

Ridge regression:

$\hat{\gamma}_k = D_c^T X^T \alpha_k \Rightarrow K = XD_c^2 X^T = \sum_j c_j^2 X_{(j)} X_{(j)}^T = \sum_j c_j^2 K_j$

Optimize:

$$\sum_k \| y_k - XD_c \gamma_k \|^2 + \lambda \gamma_k^T \gamma_k \text{ s.t. } \sum_j c_j^2 = d$$
Nonlinear KDA with Feature Selection

• Core of the algorithm: **Ensemble of linear systems** with different right-hand sides:

\[(\sum_{j=1}^{m} c_j^2 K_j + \lambda I) a_k = y_k\]

• Approximation: **block conjugate gradients** (Dubrulle 01)

• For fixed number of iterations: \(\Rightarrow O(n^2)\) algorithm

• (Lanckriet et al. 2004): **quadratically constrained QP**: \(O(n^{4.5})\) algorithm and \(O(n^3)\) approximation.

• (Sonnenburg et al., 2006): **semi-infinite LP**: Complexity unclear, seems to work well.
Multilabel Classification as Special Case

- Object $x_i$ can have **multiple** labels $\leadsto$ label set $Y_i$

- **Probabilistic treatment**: object is generated from mixture of class-specific distributions.
  $\leadsto$ multilabels modeled as class-assignment probabilities

\[
p(C_j | x_i) = \frac{1}{|Y_i|}.
\]

- Run one single M-step.

- **Inference for $x_*$**: assign object to $k$ most probable classes such that

\[
\sum_{j=1}^k p_{\text{ordered}}(C_j | x_*) > \frac{1}{2}.
\]
Effect of Multiple Labels
Original 8 kernels

Extended kernel set (32)

• “correct” probabilistic handling of multiple labels.

• use of 32 kernels (instead of 8). Newest version: 120 kernel matrices: block-CG with optimized hard-disk access.
Learned Kernel Weights

• **Surprising observation:** genetic interactions are most important...but only RBF variants!
Hierarchical Extension: The Kernel HMM

Hierarchical classification: multiple layers

Tree-trellis variant of Viterbi algorithm:

$\Rightarrow k$ most probable paths.

Consistent handling of multiple hierarchical labels
**FunCat 2.0 hierarchy (MIPS).**

⇒ increased performance in deeper levels of hierarchy.
Summing Up

- Kernels as “common language” for heterogeneous data

- Aggregation of kernels ⇒ new kernel

- **Goal:** train classifier and find optimal kernel weights

- **Model:** Nonlinear Kernel Discriminant Analysis:
  - probabilistic treatment of multiple labels
  - efficient algorithm
  - excellent prediction of protein functions
  - “building block” in HMM ⇒ hierarchical classification
Document Classification: RCV1 corpus

BEGIN
CCAT
ECAT
GCAT
MCAT
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
C1151
END