Support Vector Machines (SVM) in bioinformatics

Day 2:
Bioinformatics applications

Jean-Philippe Vert

Bioinformatics Center, Kyoto University, Japan
Jean-Philippe.Vert@mines.org

Human Genome Center, University of Tokyo, Japan, July 17-19, 2002.

3 days outline

- Day 1: Introduction to SVM
- Day 2: Applications in bioinformatics
- Day 3: Advanced topics and current research

Today's outline

1. Overview
2. Tissue classification
3. Gene function prediction
4. Protein localization
5. Secondary structure prediction
6. Protein superfamily prediction

Overview

Types of problems

- high dimensional (sequences, microarray data)
- very small or very large data sets
- heterogeneous but complementary data

Types of data

- sequences (of nucleotides or amino-acids)
- microarray expression data
- SNPs
- phylogenetic profiles
- networks (protein interaction network, biochemical pathways)

Tissue classification from microarray data

The problem

- Main goal : classification of tissue sample (e.g., type of cancer) based on microarray data (diagnosis)
- Secondary goal: Find genes potentially responsible for the classification (new insights for drug design)
- Few samples (20-30)
- Large dimensions (5,000 - 100,000)

References

- S. Mukherjee, P. Tamayo, J.P. Mesirov, D. Slonim, A. Verri, T. Poggio. **Support vector machine classification of microarray data.** A.I. Memo 1677, MIT Artificial Intelligence Laboratory, 1998.

- T.S. Furey, N. Duffy, N. Cristianini, D. Bednarski, M. Schummer, D. Haussler. **Support vector machine classification and validation of cancer tissue samples using microarray expression data.** *Bioinformatics*, 16(10):906-914, 2000.

- I. Guyon, J. Weston, S. Barnhill, V. Vapnik. **Gene selection for cancer classification using support vector machines.** *Machine Learning*, 46(1/3):389-422, Jan 2002.

Data

- Leukemia dataset: two types of leukemia (47 ALL and 25 ALL), 7,129 gene expression, 38 training and 34 test samples
- Ovarian cancer dataset: 16 normal vs 15 cancerous tissues, 97,802 cDNA expression data.
- Colon tumor dataset: 40 tumor and 22 normal colon tissues, 6,500 gene expressions data

Classification with SVM

- a linear kernel gives the best results.
- Almost perfect classification...
- ...but other algorithms perform well too (e.g., linear perceptron).

Gene selection with the Fisher-like score

- (Mukherjee 1998, Furey 2000): genes are ranked according to

$$F(g) = \left| \frac{\mu_1(g) - \mu_{-1}(g)}{\sigma_1 g + \sigma_{-1}(g)} \right|.$$

- Performance seems to increase (with 50-1,000 genes selected)
- Few biological relevance: only 5 of the 10 best cDNA selected in the ovarian cancer dataset are actually genes, 3 of which known to be cancer related
- Good classification is still possible by removing up to the 1,000 best genes.

Gene selection using SVM weights (Guyon et al. 2002)

- Genes are ranked based on their weight learned by a SVM
- Genes are removed one by one (or by chunks), and a SVM is re-run at each iteration
- Classification improves with this gene selection procedure
- Selected genes (top 7) are known to be cancer-related.

Conclusion

- diagnosis seems possible from microarray data but:
  - ⋆ larger-scale systematic experiments must be conducted (in the ovarian datasets, the origin of the cell is largely different between cancerous and normal cells..)
  - ⋆ SVM are "expected to have good performances when data increase", but currently perform at the same level as other methods
- Gene selection and biological interpretation is still a research topic. Encouraging results for SVM-based extraction methods.

Gene function prediction from microarray data

The problem

- Goal : prediction of the function of uncharacterized genes
- Unbalanced problem: each class contains few genes compared to the total number of genes (many negative examples)

References

- M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, Jr.M. Ares, D. Haussler. **Knowledge-based analysis of microarray gene expression data by using support vector machines**. *Proc. Natl. Acad. Sci. USA*, 97:262-267, 2000.

- P. Pavlidis, J. Weston, J. Cai, W.N. Grundy. **Gene functional classification from heterogeneous data**. *Proceedings of the Fifth Annual International Conference on Computational Biology*, 249-255. 2001.

Data

- 2,467 genes of the yeast *S. Cerevisiae* with known function (in the MIPS functional catalog)
- 79 expression measurement (Spellman et al., 1998).
- A small number of functional classes supposed to be correlated with gene expression are selected (TCA cycle, cytoplasmic histones...).

Results

- Linear, polynomial and Gaussian kernels
- Compared with Parzen windows, Fisher's linear discriminant, decision trees
- For all classes, SVM with Gaussian kernel performs best.

Heterogeneous information (Pavlidis et al. 2001)

- Combining microarray data with phylogenetic profiles to improve gene function prediction
- The phylogenetic profile of a gene is a vector.
  - ⋆ Each dimension corresponds to one fully sequenced organism
  - ⋆ Each value is $-\log E$, where E is the lowest E-value reported by BLAST in a search against complete genome
- Use 24 genomes
- Use polynomial kernels of degree 3.

Integration of heterogeneous data

- Early: concatenate expression $\vec{e}$ and profile $\vec{p}$ into a single vector
- Intermediate: form a kernel by adding the microarray kernel and the profile kernel:

$$K(g, g') = K(\vec{e}, \vec{e}') + K(\vec{p}, \vec{p}').$$

- Late: train two separate SVM, and add together the discriminant functions

Results

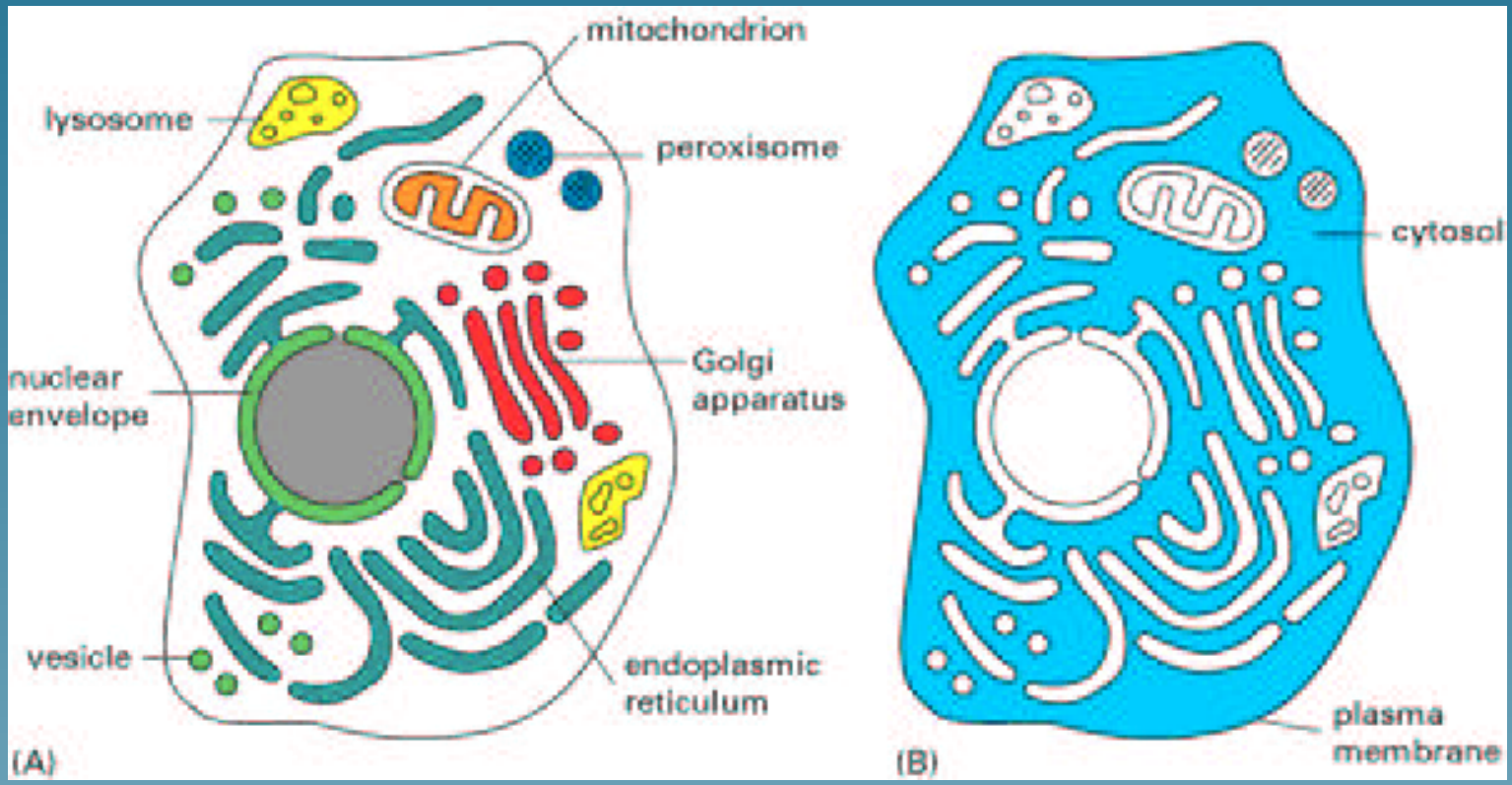
- Intermediate integration gives the best results...
- but integration of both data types fails to improve performance on 4 out of 27 classes (when one data set performs poorly compared to the other).
- feature selection (Fisher score based or SVM based) does not solve this problem.
- The contribution of phylogenetic profiles seems to be their ability to summarize sequence similarity (and not function conservation during evolution).

Conclusion

- SVM performs better than other classical learning algorithms.
- Interesting data integration by summing up two kernels (based on the prior knowledge that correlations within each data set are more relevant than between them).

Protein subcellular localization prediction

The problem



Predict protein localization from sequence.

References

- S. Hua and Z. Sun. **Support vector machine approach for protein subcellular localization prediction**. *Bioinformatics*, 17(8):721-728, 2001.

- K. Park, *in preparation*.

Data

- 997 prokaryotic sequences divided in 3 classes: cytoplasmic (688), periplasmic (202) and extracellular (107).
- 2427 eukaryotic sequences divided into 4 classes : nuclear (1097), cytoplasmic (684), mitochondrial (321) and extracellular (325).

SVM approach

- Each sequence is transformed into a 20-dimensional vector (amino-acid composition)
- Multiclass problem: 1-versus-all approach
  - ⋆ One SVM is trained for each class
  - ⋆ predict the localization with highest output value

Results

- Best performance with Gaussian kernel.
- Comparison with other composition-based methods: (accuracy with a Jackknife test):

Sequence	NN	Cov. Disc	Markov	SVM
Prok	81	86.5	89.1	91.4
Euk	66		73.0	79.4

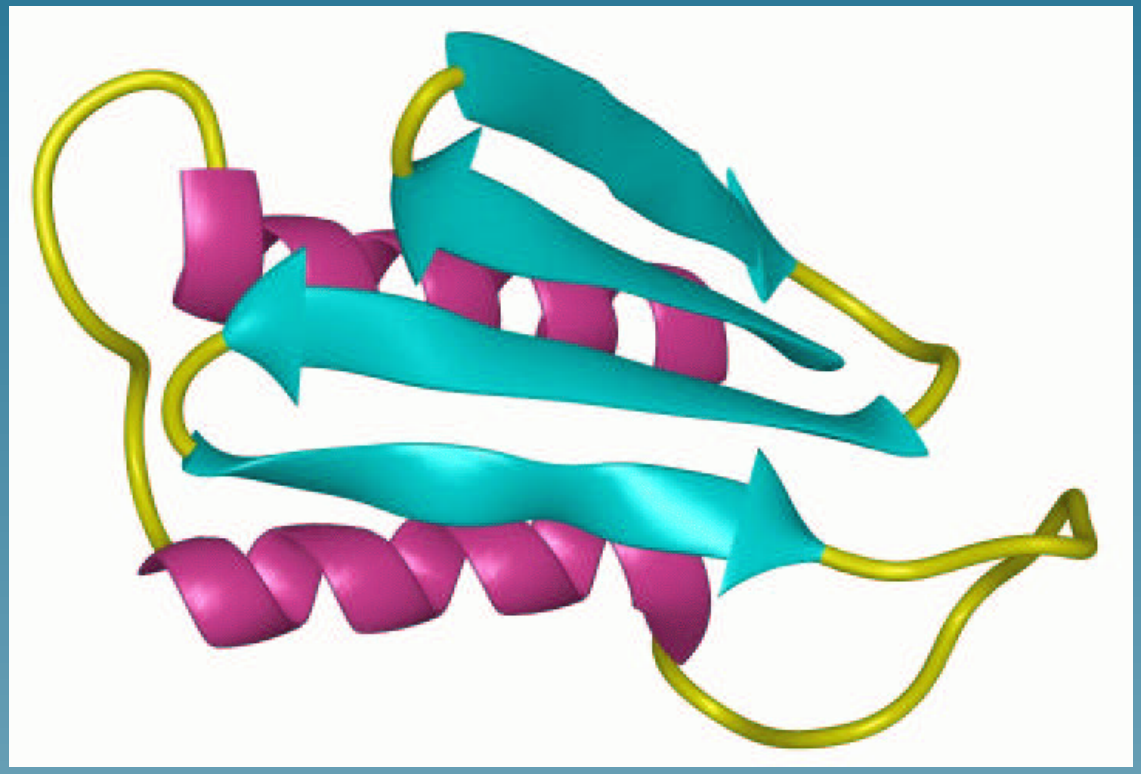
- No comparison with signal recognition-based approach (PSort, TargetP)

Conclusion

- SVM perform better than other composition-based methods.
- Integration with signal recognition is possible and might increase performance (K. Park, personal communication)

Protein secondary structure prediction

The problem



Predict local structure from sequence (ex: prion)

References

- S. Hua and Z. Sun. **A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach**. *Journal of Molecular Biology*, 308(2):397-407, April 2001.

Data

- Non-redundant sets of proteins (RS126 and CB513) with known 3D structure.
- 3 states: helix, sheet and coil.

SVM approach

- Multiple sequence alignment is performed for each sequence (available in the HSSP database)
- Each position is encoded into a vector using a sliding window of size l. Dimension: $21 \times l$.
- Multiclass problem:
  - ⋆ 1-vs-all with maximum score
  - ⋆ all-vs-all + decision trees (hand-designed)
  - ⋆ all-vs-all + vote
  - ⋆ all-vs-all + NN

Results

- 1-vs-all with max score gives the best result (can be improved by a jury decision from all methods)
- SOV index on the RS126 dataset (sevenfold cross-validation):

Method	PREDATOR	DSC	NNSSP	PHD	SVM
SVO (%)	70.3	71.1	72.7	73.5	74.6

Protein fold prediction

The problem

- Fold = common 3D pattern with the same major secondary structure elements in the same arrangement and with the same topological connections.
- Virtually no sequence similarity
- SCOP, CATH: more than 600 folds are known.
- Goal: predict the fold of a protein from its amino-acid sequence

References

- C. Ding and I Dubchak. **Multi-class protein fold recognition using support vector machines and neural networks**. *Bioinformatics*, 17:349-358, 2001.

Approach

- Multiclass problem:
  - ⋆ 1-vs-all: predict all folds with positive scores (*multiclass prediction*)
  - ⋆ Unique 1-vs-all: perform 1-vs-all, keep only folds with positive score, and perform all-vs-all + vote on these folds only
  - ⋆ All-vs-all + vote.
- Compare NN and SVM.

Protein vectorization

Each protein is transformed into a 125-dimensional vector by extracting features from the amino-acid sequence:

- AA composition
- secondary structure
- hydrophobiciy
- Van der Waals volume
- Polarity
- Polarizability

Results

- Prediction on 27 SCOP folds
- Both unique 1-vs-all and all-vs-all are significantly better than 1-vs-all (because false positive are removed).
- No clear difference between unique 1-vs-all and all-vs-all
- SVM significantly better than NN, and 10-100 times faster.

Protein superfamily prediction with the Fisher kernel

The problem

- Goal: detecting remote protein homology (sequence similarities that direct methods like BLAST don't detect)
- Use the SCOP classification. A superfamily is a set of proteins with a common 3D structure believed to have evolutionary relationship (family ¡ superfamily ¡ fold)
- Other methods: BLAST, Fasta, PROBE, profiles, PFAM, HMMs, PSI-BLAST
- None of this method is discriminative

References

- T. Jaakkola, M. Diekhans and D. Haussler. **A discriminative framework for detecting remote protein homologies**. *Journal of Computational Biology*, 7(1,2):95-114, 2000.

- L. Liao and W.S. Noble. **Combining pairwise sequence similarity and support vector machines for remote protein homology detection**. Proceedings of the Sixth International Conference on Computational Molecular Biology. *to appear in Bioinformatics*, 2002.

How to represent protein sequences as vectors?

- The Fisher score (Jaakkola et al.)
- Vector of pairwise similarities (Liao and Noble)

The Fisher score of HMM

- A set of HMM for various families (e.g., immunoglobulins) is given (using existing databases).
- Any HMM $\mathcal{H}$ defines the probability $P(x|, \mathcal{H}, \theta)$ of any sequence x (θ is the parameter vector which contains emission and transitions probabilities).
- The Fisher score vector of a given sequence x is:

$$U(x) = \nabla_\theta \log P(x|, \mathcal{H}, \theta).$$

Computing the Fisher score

For a classical HMM, let

- $P(x|s,\theta) = \theta_{x|s}$ the probability of emitting a residue x while in state s.
- $P(s'|s.\tau) = \tau_{s'|s}$ the transition probability from the current state s to the next state s'.

The probability of a sequence $X = x_1 \dots x_n$ is:

$$P(X|\theta,\tau) = \sum_{s_1,\dots,s_n} \prod_{i=1}^{n} \theta_{x_i|s_i} \tau_{s_i|s_{i-1}}.$$

Computing the Fisher score (ctd.)

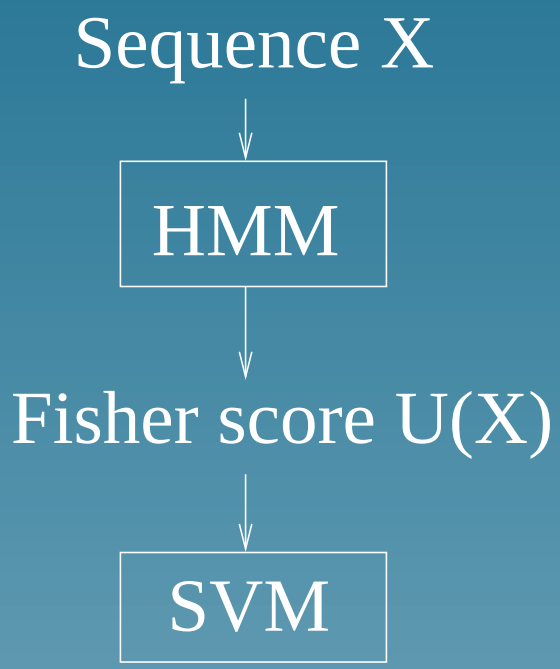
This probability can be computed by the classical forward-backward algorithm, which also gives the posterior expectations $\xi(x, s)$ of visiting state x and generating residue x.

Derivating the preceding equation gives:

$$\frac{\partial}{\partial \theta_{x|\theta}} \log P(X|\theta, \tau) = \frac{\xi(x, s)}{\theta_{x|\theta}} - \xi(s).$$

The Fisher score vector can be computed as a by-product of the forward-backward algorithm

Using the Fisher score with SVM

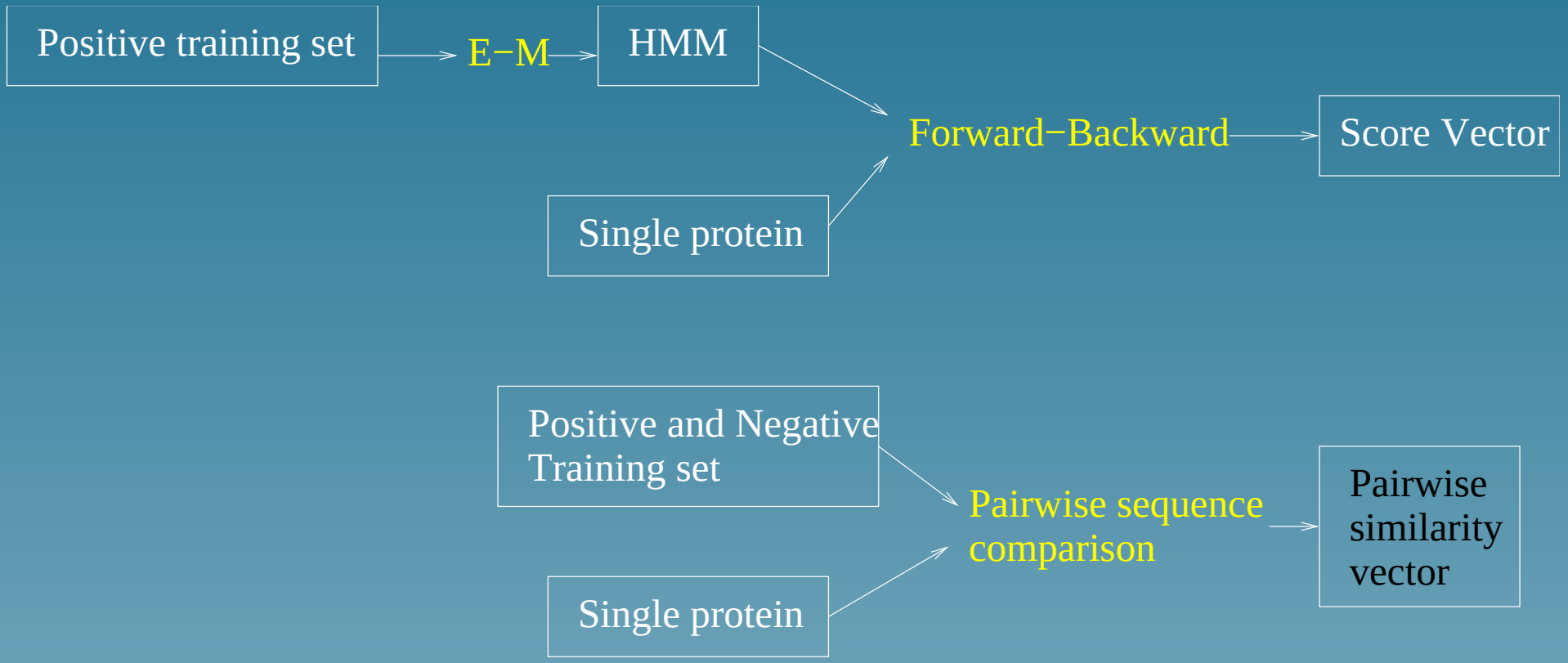


Vector of similarities

- The similarity between any two protein sequences can be computed using the Smith-Waterman algorithm.
- Let $f(x, y)$ the log of the P-value of the Smith-Waterman score between two sequence x and y.
- For a training set $\{x_1, \ldots, x_n\}$, one can represent each sequence x by the n-dimensional vector:

$$\vec{\Phi}(x_i) = \begin{pmatrix} f(x, x_1) \\ \vdots \\ f(x, x_n) \end{pmatrix}$$

Comparison of both approaches



Results

Experiment: for 33 SCOP family, recognize the superfamily by only using sequences in other families (simulate a remote protein homology problem).

1. Pairwise similarity + SVM
2. Fisher score + SVM
3. PSI-Blast and SAM
4. Other direct homology-based methods enditemize

Discussion and conclusion

Summary

The examples we saw today all involve 4 important steps subject to discussion:

- Expressing the problem as a binary classification problem
- Finding a vector representation of the objects to be classified
- Using an appropriate algorithm to learn the classification
- Evaluating the performance

Conclusion

- SVM have been tested on many bioinformatics problems in recent years
- In many cases SVM outperform other classification methods
- However comparison is sometimes difficult because not all problems are stated as a clean machine learning problem
- Handling multiclass is still not trivial (not only for SVM)
- In today's examples, SVM were only used as replacement for NN or Fisher discriminant. See tomorrow for examples where SVM provide more than that.