CSE 847: Statistical Machine Learning

Kernel Methods

Jiayu Zhou
Computer Science & Engineering
Michigan State University

Outline

- Kernel Methods: Basic ideas
- Kernels and similarity
- How to choose kernels?
- Kernels and learning

Kernel Methods: Basic ideas

Given two vectors: $P = (p_1, \quad p_2, \quad \cdots \quad p_n) \in \Re^n$

$Q = (q_1, \quad q_2, \quad \cdots \quad q_n) \in \Re^n$

How to compute the similarity or distance between P and Q?

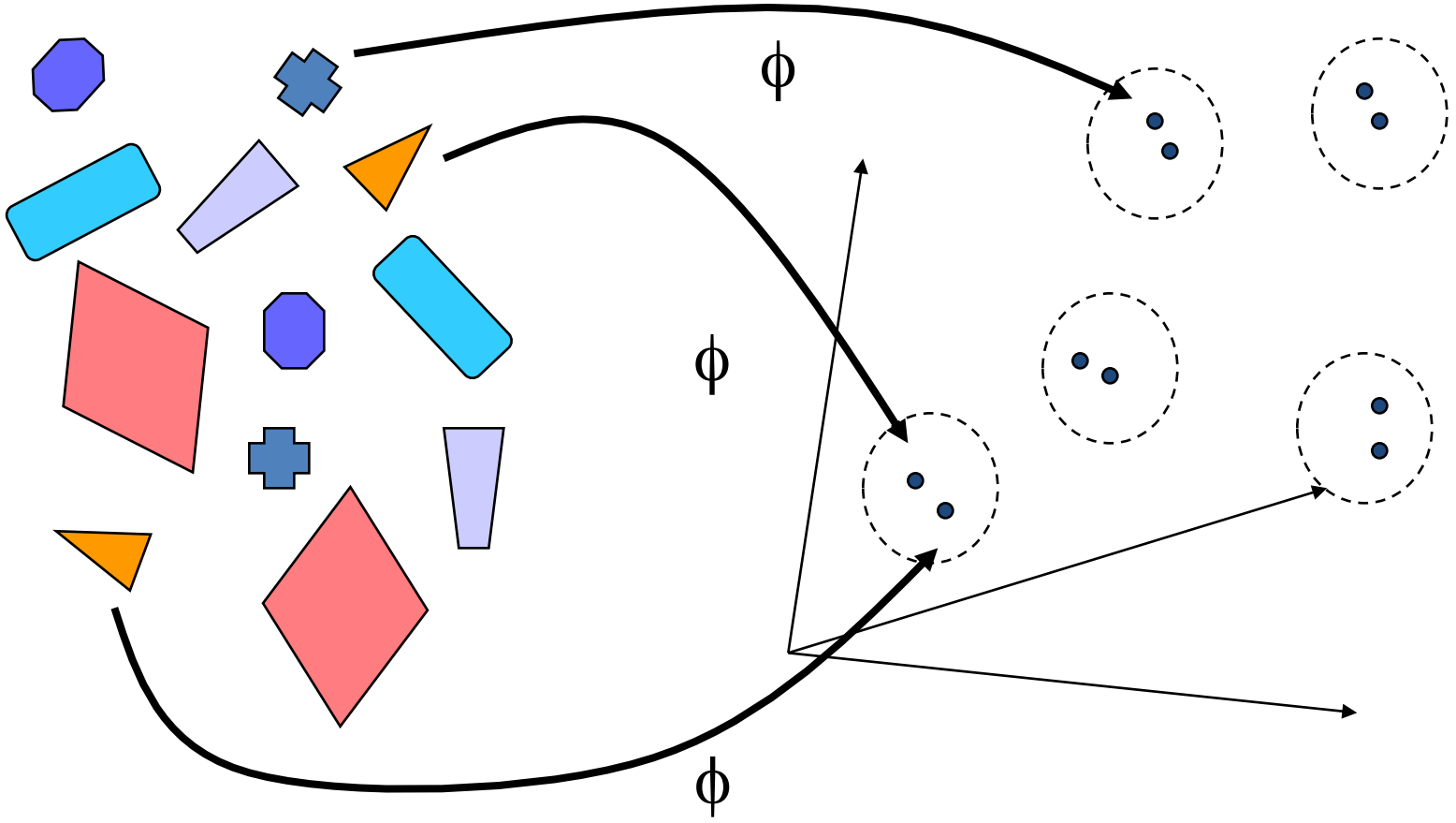
- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

- Minkowski Distance

$$\boldsymbol{dist = (\sum_{k=1}^{n} | p_k - q_k |^r)^{\frac{1}{r}}}$$

Kernel Methods: Basic ideas



Kernel Methods: Basic ideas

- Find a mapping ϕ such that, in the new feature space, problem solving is easier (e.g. linear).
- The *kernel* is defined as the inner product between data points in this new feature space.
  - Similarity measure
- But the mapping is left implicit.
  - Kernel trick
- Easy generalization of a lot of inner product (or distance) based pattern recognition algorithms.
  - SVM, PCA, LDA, CCA, K-Means, etc.

Applications in bioinformatics

Protein sequence

Protein structure

VMVKVGDKVAAEQSLITVE ------

Alphabet of 20 amino acids

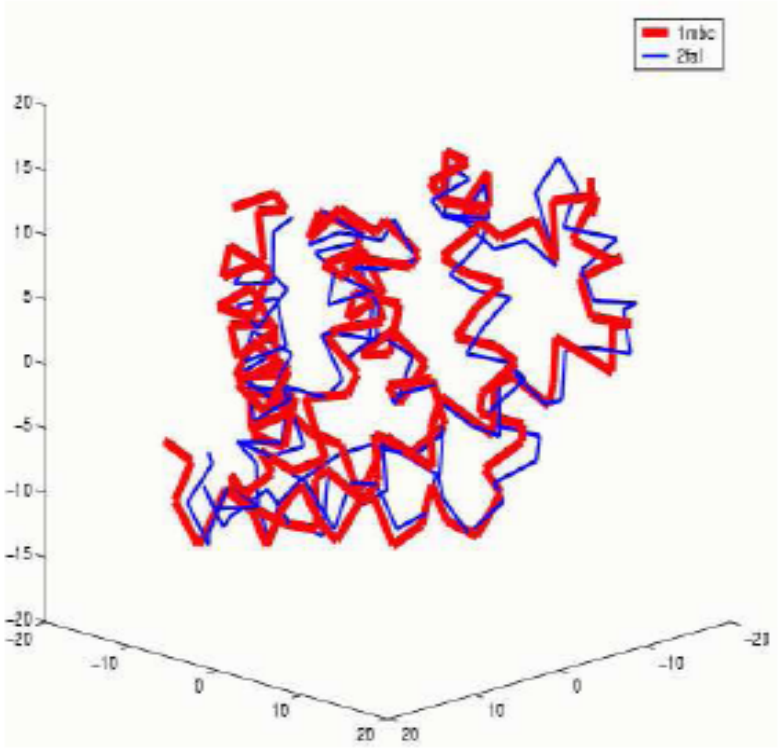
Applications in bioinformatics

Pairwise protein sequence alignment

Pairwise protein structure alignment

Myoglobin family

1mbc: VLSEGEWQLVLHVW A...
2fal: XSLSAAEADLAGKSW...



Kernel Methods: Basic ideas

A kernel k(x,y)

- is a similarity measure
- defined by an implicit mapping ϕ, from the original space to a feature space
  - $k(x,y)=\phi(x)\bullet\phi(y)$
- The feature space is possibly infinite dimensional, but still computational efficiency when computing $k(x,y)$

Kernel Methods: Basic ideas

- The function $k(x,y)$ is a valid kernel, if there exists a mapping ϕ into a vector space (with a inner product) such that k can be expressed as $k(x,y)=\phi(x)\bullet\phi(y)$.

- Theorem: $k(x,y)$ is a valid kernel if k is positive semi-definite and symmetric (Mercer Kernel)
  - A function is PSD if $\int K(\mathbf{x},\mathbf{y})f(\mathbf{x})f(\mathbf{y})d\mathbf{x}d\mathbf{y} \geq 0 \quad \forall f \in L_2$
  - In other words, the Gram matrix **K** (whose elements are $k(x_i,x_j)$) must be positive semi-definite for all x_i, x_j of the input space.

Kernel Methods: Basic ideas

In kernel methods, the **sole** information used from the training data set is the Kernel Gram Matrix

$$K_{training} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_m) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_m) \\ \dots & \dots & \dots & \dots \\ k(\mathbf{x}_m, \mathbf{x}_1) & k(\mathbf{x}_m, \mathbf{x}_2) & \dots & k(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$

If the kernel is valid, K is symmetric positive semi-definite.

Why positive semi-definite?

Support Vector Machines (SVM:)

Maximize $\sum_{k=1}^{R} \alpha_k - \frac{1}{2} \sum_{k=1}^{R} \sum_{l=1}^{R} \alpha_k \alpha_l Q_{kl}$ where $Q_{kl} = y_k y_l K(\mathbf{x}_k, \mathbf{x}_l)$

Subject to these constraints: $0 \le \alpha_k \le C \quad \forall k$ $\sum_{k=1}^{R} \alpha_k y_k = 0$

If K is not positive semi-definite, the optimization problem is not convex and the algorithm may not find the global optimal solution.

Outline of lecture

- ❑ Kernel Methods: Basic ideas
- ❑ Kernels and similarity
- ❑ How to choose kernels?
- ❑ Kernels and learning

Kernels and similarity

- Intuition of kernels as similarity measures:

$$k(\mathbf{x}, \mathbf{x}') = \frac{\|\phi(\mathbf{x})\|^2 + \|\phi(\mathbf{x}')\|^2 - d(\phi(\mathbf{x}'), \phi(\mathbf{x}'))^2}{2}$$

- When the diagonal entries of the Kernel Gram Matrix are constant, kernels are directly related to similarities.
  - For example Gaussian Kernel

$$k_G(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{d(\mathbf{x}, \mathbf{x}')^2}{2\sigma^2}\right)$$

- In general, it is useful to think of a kernel as a similarity measure.

From similarity scores to kernels

Empirical Kernel Map

- Choose a finite set of template samples and compute the similarity of x with all these samples:

$$\mathbf{x} \in \mathcal{X} \rightarrow \phi(\mathbf{x}) = (s(x, t_1), \ldots, s(x, t_r))^\top \in \mathbb{R}^p .$$

- Construct the kernel based on the similarity to the template samples:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \quad k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}) = \sum_{i=1}^{r} s(\mathbf{x}, t_i)\, s(\mathbf{x}', t_i) .$$

From similarity scores to kernels

Removal of negative eigenvalues

- Form the similarity matrix S, where the (i,j)-th entry of S denotes the similarity between the i-th and j-th data points. S is symmetric, but is in general not positive semi-definite, i.e., S has negative eigenvalues.

$$S = U\Sigma U^T, \text{ where } \Sigma = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n), \text{ and}$$

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq 0 > \lambda_{r+1} \geq \cdots \geq \lambda_n.$$

$$K = U\hat{\Sigma}U^T, \text{ where } \hat{\Sigma} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_r, 0, \cdots, 0).$$

Outline of lecture

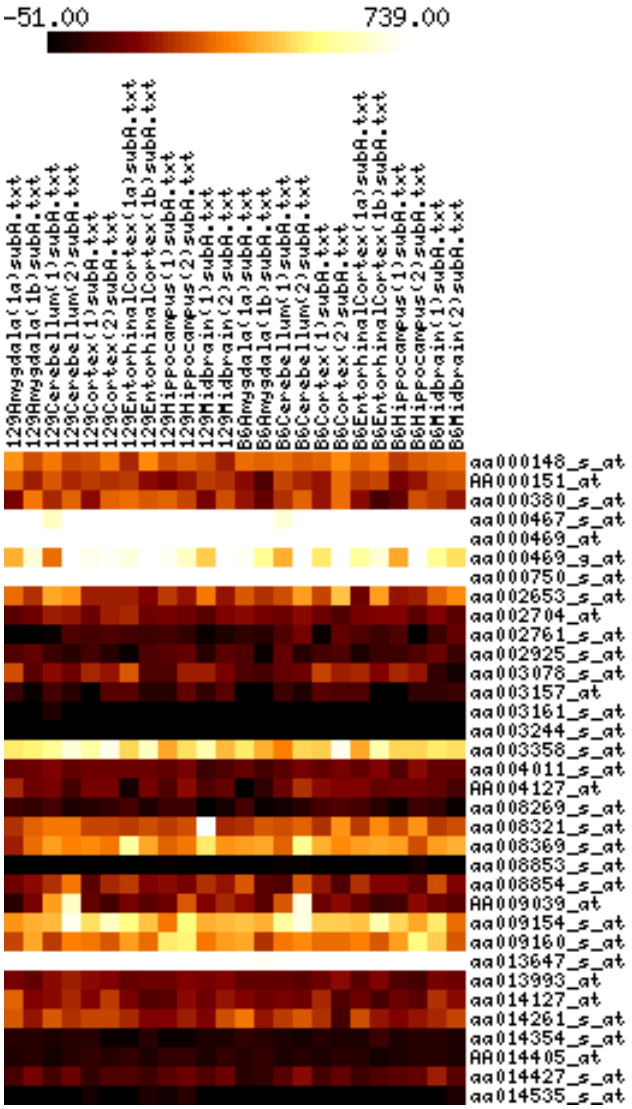
- ❑ Kernel Methods: Basic ideas
- ❑ Kernels and similarity
- ❑ How to choose kernels?
- ❑ Kernels and learning

How to choose kernels?

- There is no absolute rule for choosing the right kernel, adapted to a particular problem.

- Kernel should capture the desired similarity.
  - Kernels for vectors: Polynomial and Gaussian kernel
  - String kernel (text documents)
  - Diffusion kernel (graphs)
  - Sequence kernel (protein, DNA, RNA)

Kernel Design: expression kernel



- Each matrix entry is an mRNA expression measurement.
- Each column is an experiment.
- Each row corresponds to a gene.

Vectorial data

Kernel Design: linear expression kernel

- Normalized scalar product

$$K(X,Y) = \frac{\sum_i X_i Y_i}{\sqrt{\sum_i X_i X_i}\sqrt{\sum_i Y_i Y_i}}$$

- Similar vectors receive high values, and vice versa.

Similar

Dissimilar

Kernel Design: Gaussian expression kernel

- Use general similarity measurement for vector data: Gaussian kernel

$$K(X,Y) = \exp\left(\frac{-\|X-Y\|^2}{2\sigma^2}\right)$$

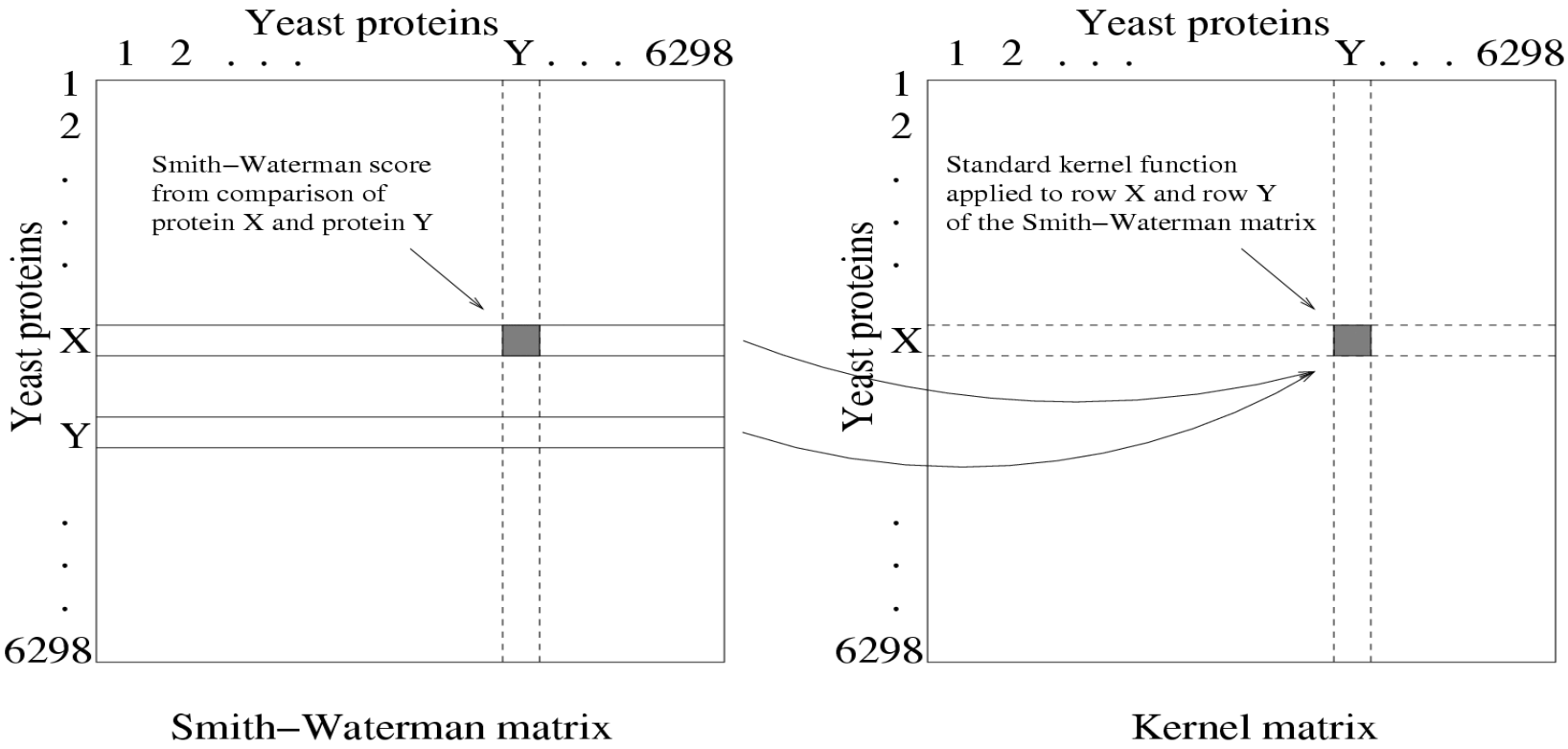
Kernel Design: sequence kernel

- Scalar product on a pair of variable-length, discrete strings ??

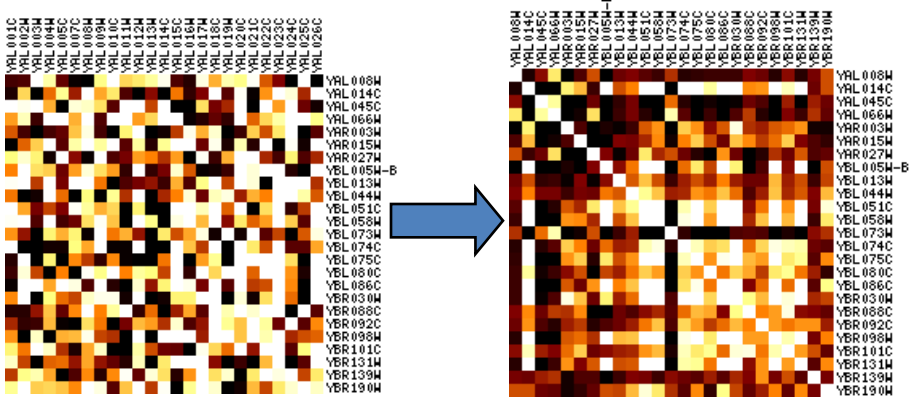
```
>ICYA_MANSE
GDIFYPGYCPDVKPVNDFDLSAFAGAWHEIAKLPLENENQGKCTIAEYKY
DGKKASVYNSFVSNGVKEYMEGDLEIAPDAKYTKQGKYVMTFKFGQRVVN
LVPWVLATDYKNYAINYMENSHPDKKAHSIHAWILSKSKVLEGNTKEVVD
NVLKTFSHLIDASKFISNDFSEAACQYSTTYSLTGPDRH

>LACB_BOVIN
MKCLLLALALTCGAQALIVTQTMKGLDIQKVAGTWYSLAMAASDISLLDA
QSAPLRVYVEELKPTPEGDLEILLQKWENGECAQKKIIAEKTKIPAVFKI
DALNENKVLVLDTDYKKYLLFCMENSAEPEQSLACQCLVRTPEVDDEALE
KFDKALKALPMHIRLSFNPTQLEEQCHI
```

Non-vectorial data

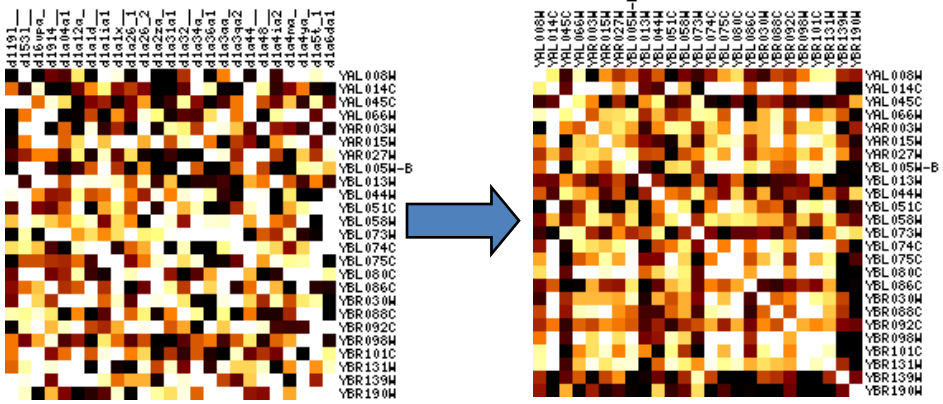
Kernel Design: sequence comparison kernel



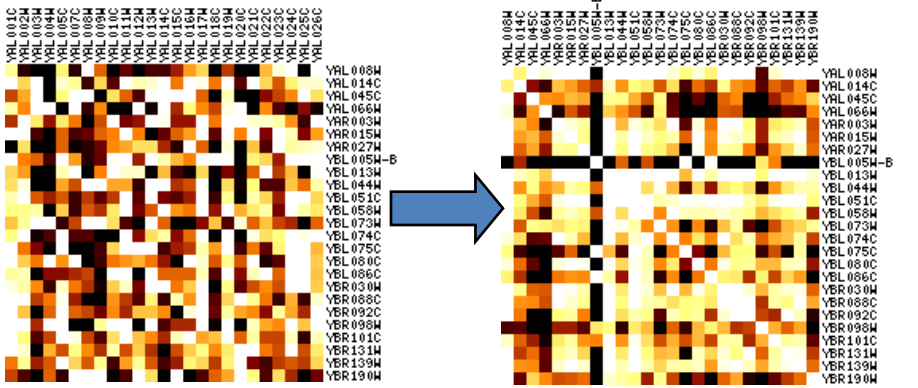
Kernel Design: sequence comparison kernel - variants



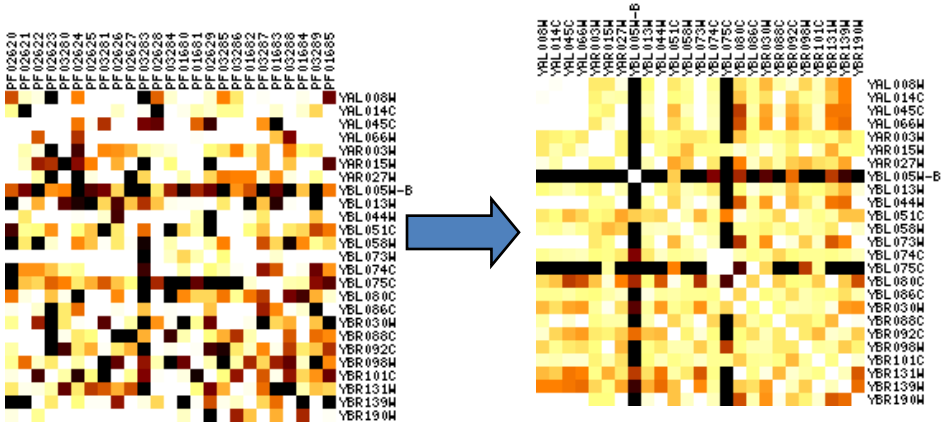
Smith-Waterman all-vs-all



Smith-Waterman w.r.t. SCOP db



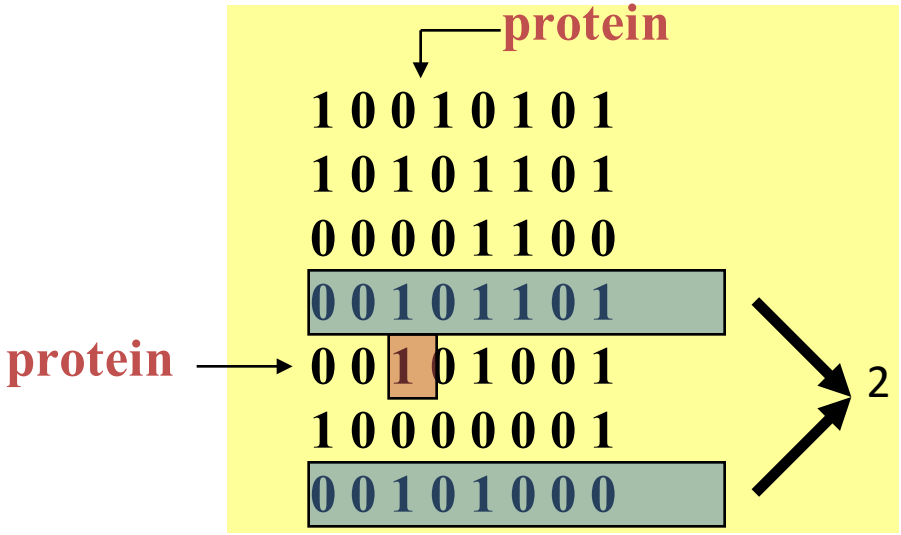
BLAST all-vs-all



E-values from Pfam database

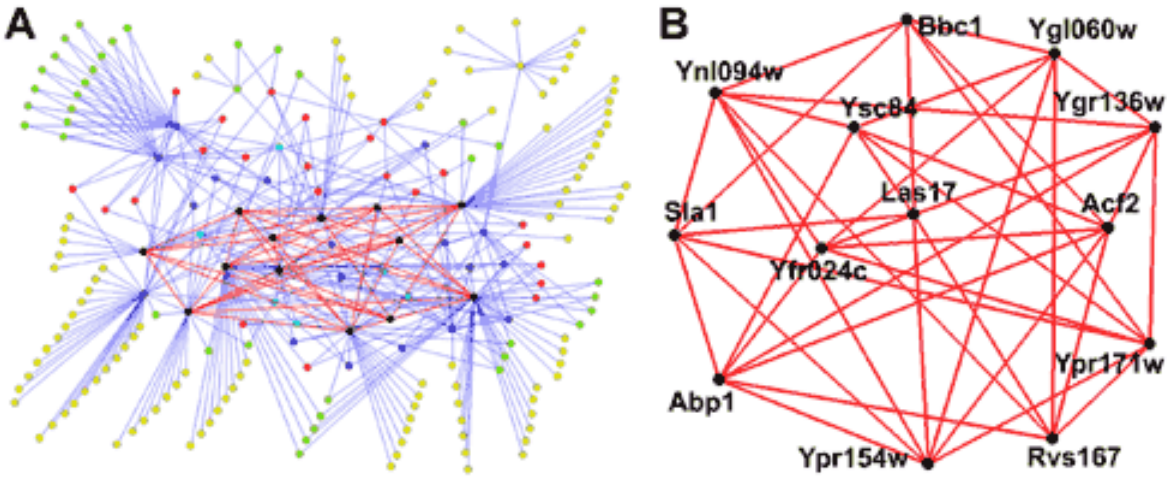
Kernel Design: linear interaction kernel

- Pairwise interactions can be represented as a graph or a matrix.
- The simplest kernel counts the number of shared interactions between each pair.



Kernel Design: diffusion kernel

- A general method for establishing similarities between nodes of a graph.
- Based upon a random walk.
- Efficiently accounts for all paths connecting two nodes, weighted by path lengths.



How to build new kernels

- Kernel combinations, preserving *validity*:

$$K(\mathbf{x},\mathbf{y}) = \lambda K_1(\mathbf{x},\mathbf{y}) + (1-\lambda)K_2(\mathbf{x},\mathbf{y}) \quad 0 \le \lambda \le 1$$

$$K(\mathbf{x},\mathbf{y}) = a \bullet K_1(\mathbf{x},\mathbf{y}) \quad a > 0$$

$$K(\mathbf{x},\mathbf{y}) = K_1(\mathbf{x},\mathbf{y}) \bullet K_2(\mathbf{x},\mathbf{y})$$

How to build new kernels

$$K(\mathbf{x},\mathbf{y}) = f(x) \bullet f(y) \quad f \text{ is real-valued function}$$

$$K(\mathbf{x},\mathbf{y}) = K_3(\boldsymbol{\varphi}(\mathbf{x}),\boldsymbol{\varphi}(\mathbf{y}))$$

$$K(\mathbf{x},\mathbf{y}) = \mathbf{x}'P\mathbf{y} \quad P \text{ symmetric definite positive}$$

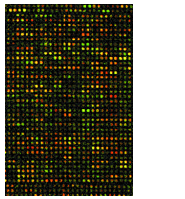
$$K(\mathbf{x},\mathbf{y}) = \frac{K_1(\mathbf{x},\mathbf{y})}{\sqrt{K_1(\mathbf{x},\mathbf{x})}\sqrt{K_1(\mathbf{y},\mathbf{y})}}$$

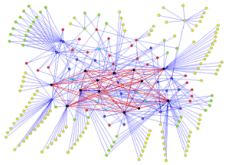
Outline of lecture

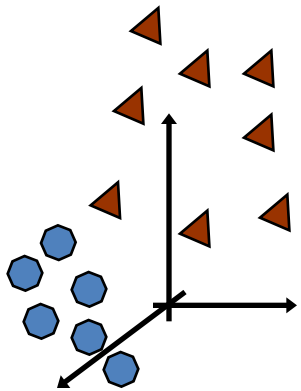
- ❑ Kernel Methods: Basic ideas
- ❑ Kernels and similarity
- ❑ How to choose kernels?
- ❑ Kernels and learning

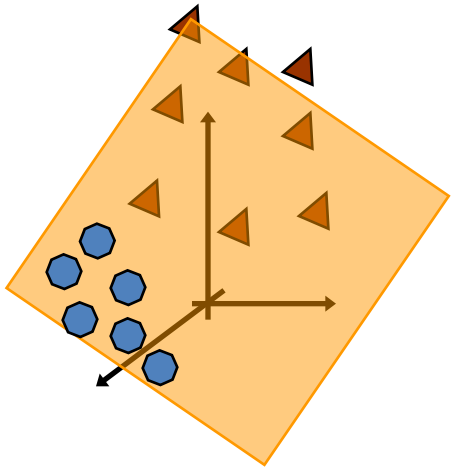
Kernel-based Learning

Data ***Embed data*** ***Linear algorithm***





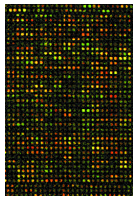


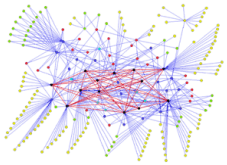


SVM, PCA, CCA, LDA…

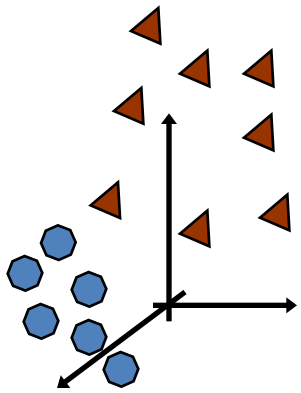
Kernel-based Learning

Data

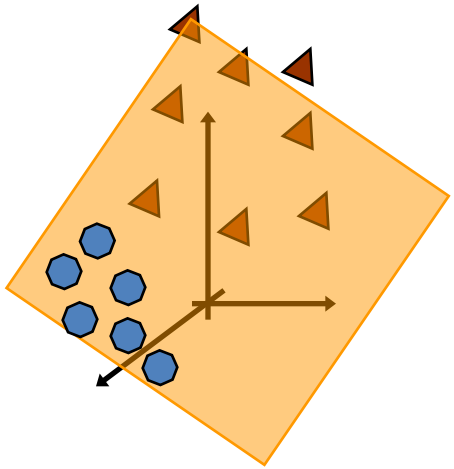




Embed data



Linear algorithm



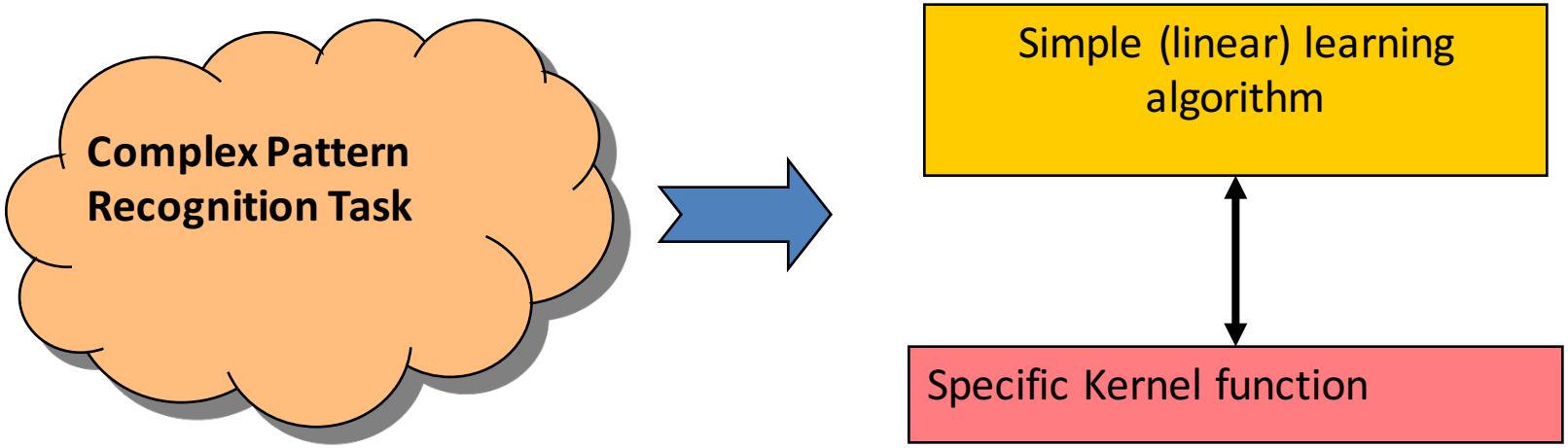
SVM, PCA, CCA, FDA…

Kernel design

Kernel algorithm

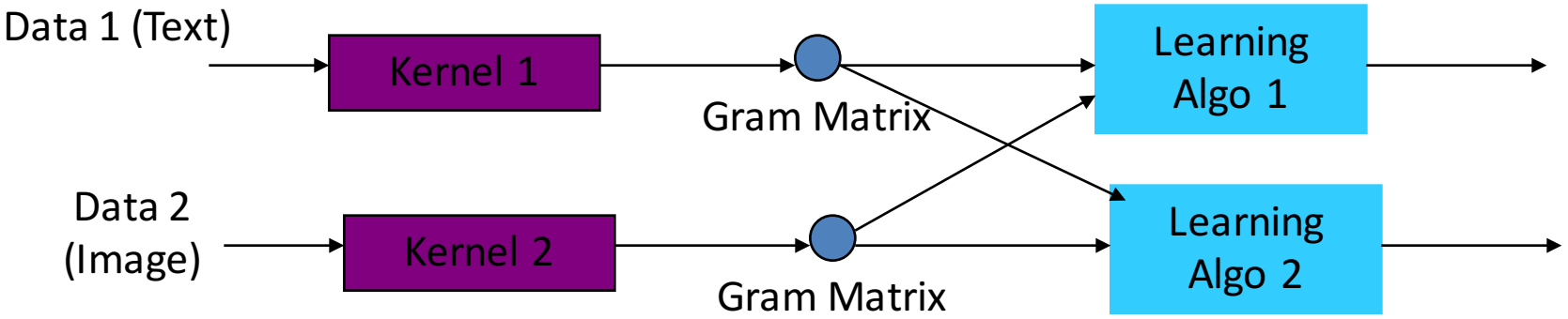
Kernels and Learning

- In Kernel-based learning algorithms, problem solving is now decoupled into:
  - A general purpose learning algorithm (e.g. SVM, PCA, LDA, CCA, etc); and
  - A problem specific kernel



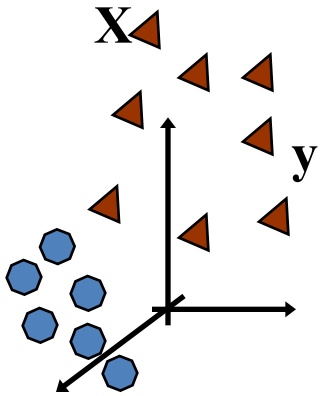
Kernels and Learning

- Modularity and re-usability
  - Same kernel,different learning algorithms
  - Different kernels, same learning algorithms



Summary

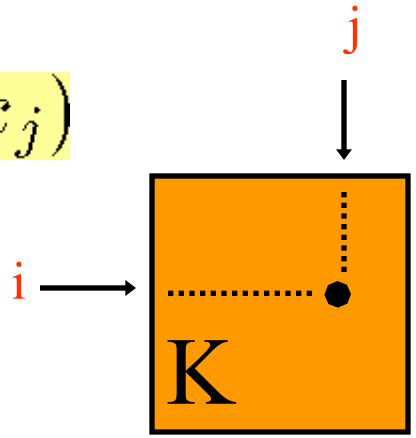
Embed data



IMPLICITLY: Inner product measures similarity

$$k(x, y)$$

$$K_{ij} = k(x_i, x_j)$$



Add domain-specific knowledge to measure similarity

Property: Any symmetric positive definite matrix specifies a kernel matrix & every kernel matrix is symmetric positive definite

Reference

- A primer on kernel methods

  Vert, Tsuda, and Scholkopf

- http://www.kernel-machines.org/

  Papers, software, workshops, conferences, etc.