

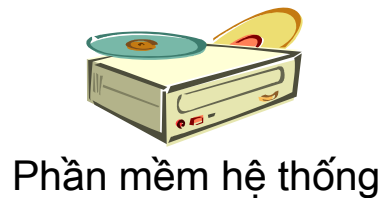
Về quan hệ giữa toán học và tin học

Hồ Tú Bảo

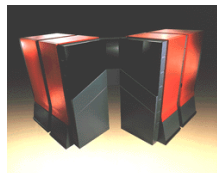
Viện Công nghệ Thông tin, Viện KH & CN Việt Nam
Japan Advanced Institute of Science and Technology

- Vài lĩnh vực tiêu biểu của toán học trong tin học
- Toán học trong xử lý ngôn ngữ tự nhiên
- Toán học và Web
- Toán học trong sinh học
- Toán học trong học máy (machine learning)

Đại thể về tin học (khoa học máy tính)

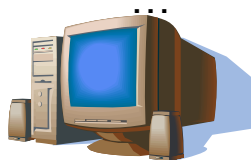


Cray Y-MP8E



Supercomputers, mainframes

Minicomputers, workstations



PC computers



An toàn thông tin



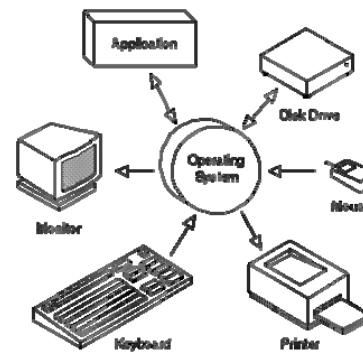
Tin sinh học



Nhận dạng



Trí tuệ nhân tạo



DOS, Windows, Mac OS, UNIX, Linux



Cơ sở dữ liệu

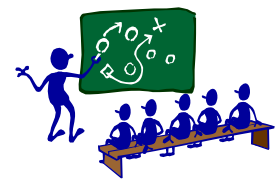
Pascal, C, C++, Java, ...



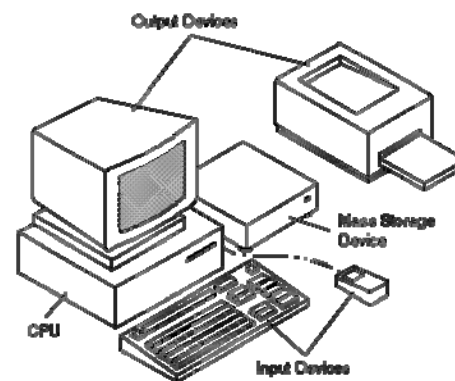
Kỹ nghệ phần mềm



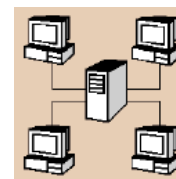
Thuật toán



Toán học cho tin học



Máy tính song song



LAN, WAN, ... Internet

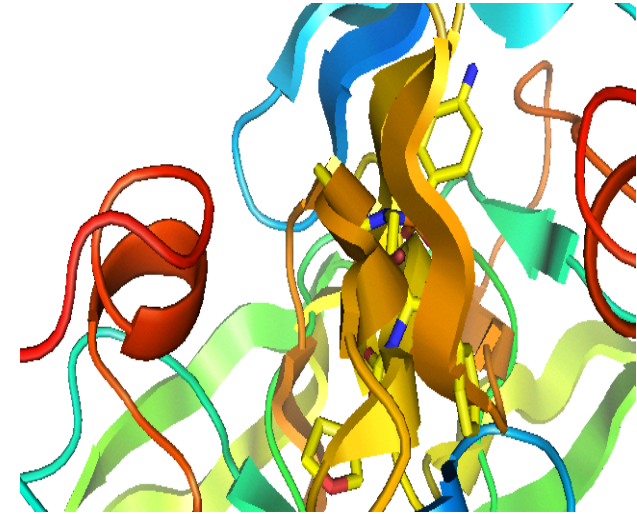
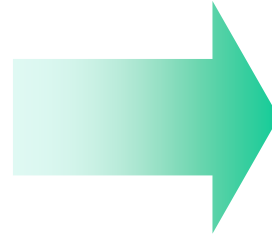
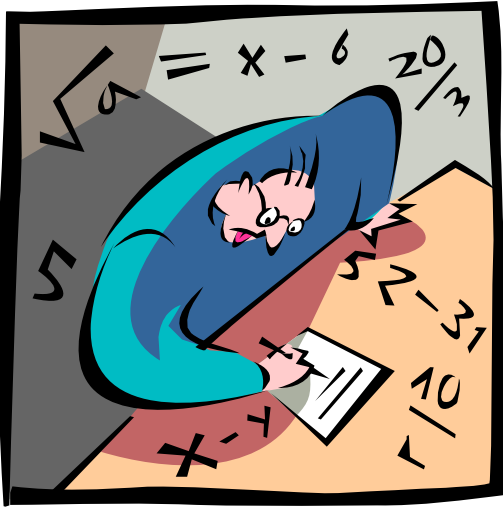
Mạng máy tính



Truyền thông

Toán học và Tin học

Hai con đường của toán trong tin học



Các lý thuyết, mô hình toán học làm cơ sở cho sự phát triển tin học.

Để giải quyết các vấn đề của tin học và ứng dụng tin học, tìm và dùng các lý thuyết và công cụ toán học.

Toán học rời rạc đối với tin học



- Logic (lập luận tự động, AI, hệ thống minh)
- Set theory (tập mờ, tập thô, ... tính toán với thông tin không đủ, không chính xác, ...)
- Number theory (bảo mật thông tin)
- Combinatorics (hình học tổ hợp, ...)
- Graph theory (mạng, sinh học, vật lý, ...)
- Digital geometry and digital topology (phân tích ảnh, etc.)
- Algorithmics (phương pháp tính toán)
- Computability (giới hạn lý thuyết và thực tế của thuật toán)
- Probability and Markov chains (xử lý tiếng nói, sinh học, ...)
- Linear algebra (phân tích dữ liệu, ...)
- Probability (ngẫu nhiên)
- Proofs (kỹ nghệ phần mềm, AI)
- etc.

Một số lĩnh vực toán học khác

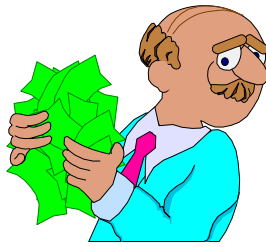


- Thống kê và phân tích dữ liệu (statistics, data analysis)
- Lý thuyết tối ưu (Optimization)

Logic trong tin học



=



Knowledge

(đại số, thống kê,
toán học rời rạc, ...)

+



Inference

(logic toán học, ...)

- Logic mệnh đề, logic tân từ, logic không chuẩn (modal, temporal, non-monotonic, fuzzy, high order, ... logic)
- Thí dụ: Máy tự động dùng tam đoạn luận (syllosism)

Elephants are bigger than dogs

Dogs are bigger than mice

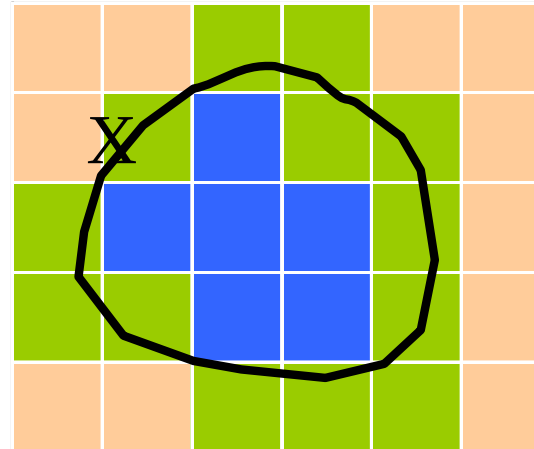
Therefore

Elephants are bigger than mice

Lý thuyết tập hợp (tập mờ, tập thô)



- Tập mờ (Zadeh 1965), tập thô (Pawlak, 1982)
- Rough fuzzy hybridization?
Rough sets + Fuzzy sets?



Xấp xỉ dưới X_* :
Hợp của các lớp tương đương nằm trọn trong X

Xấp xỉ trên X^* :
Hợp của các lớp tương đương có giao khác rỗng với X



$$U = \{\text{red circle}, \text{red triangle}, \text{blue triangle}, \text{blue square}, \text{red circle}\} \quad \mathbf{R} = \{color, shape\}$$

$$Eq.class / color = \{ \{\text{red circle}, \text{red circle}, \text{red triangle}\}, \{\text{blue triangle}, \text{blue square}\} \}$$

$$Eq.class / shape = \{ \{\text{red circle}, \text{red circle}\}, \{\text{blue triangle}, \text{red triangle}\}, \{\text{blue square}\} \}$$

$$Eq.class = \{ \{\text{red circle}, \text{red circle}\}, \{\text{blue triangle}\}, \{\text{red triangle}\}, \{\text{blue square}\} \}$$

$$X = \{ \text{red circle}, \text{blue triangle} \}, X_* = \{ \text{blue triangle} \}, X^* = \{ \text{red circle}, \text{red circle}, \text{blue triangle} \}$$

23 bài toán của thế kỷ 20



Hilbert

- Tại Đại hội Toán học Thế giới lần thứ hai (Paris, tháng Năm 1900), Hilbert nêu ra 23 bài toán, thách thức các nhà toán học toàn thế giới giải trong thế kỷ 20.
- 12 bài toán đã được giải toàn bộ, 8 bài toán được giải từng phần, 3 bài vẫn chưa có lời giải.

7 bài toán của thế kỷ 21



- 4 giờ chiều Thứ tư ngày 24 tháng 5 năm 2000, Viện Toán học Clay công bố và thách thức 7 bài toán của thế kỷ 21 (1 triệu \$ cho mỗi lời giải).
- Bài toán số 1: P versus NP
- Sáu bài toán khác:
 1. The Hodge Conjecture
 2. The Poincaré Conjecture (solved 2006)
 3. The Riemann Hypothesis
 4. Yang-Mills Existence and Mass Gap
 5. Navier-Stokes Existence and Smoothness
 6. The Birch and Swinnerton-Dyer Conjecture.

Bài toán “P versus NP”



- Nếu ai đó hỏi rằng liệu 13.717.421 có là tích của hai số nhỏ hơn không, bạn sẽ cảm thấy rất khó trả lời là đúng hay sai.
- Nếu người đó bảo bạn rằng số này có thể là tích của 3607 và 3803, bạn có thể kiểm tra điều này thật dễ dàng.
- Xác định xem với một bài toán cho trước, liệu có tồn tại một lời giải có thể kiểm chứng nhanh (bằng máy tính chẳng hạn), nhưng lại cần rất nhiều thời gian để giải từ đầu (nếu không biết lời giải)?
- Có rất nhiều bài toán như vậy. Chưa ai có thể chứng minh được rằng, với bất kỳ bài toán nào như vậy, thực sự cần rất nhiều thời gian để giải. Có thể chỉ đơn giản là chúng ta vẫn chưa tìm ra được cách giải chúng nhanh chóng. Stephen Cook phát biểu bài toán **P versus NP** vào năm 1971.

Bài toán “P versus NP”



- Bài toán **SAT**: cho trước một mạch điện tử Boolean, liệu có các cách chọn inputs sao cho output là True hay không?
- Inputs của các mạch điện tử Boolean (với cổng AND, OR và NOT) hoặc là T (true) hoặc là F (false). Mỗi cổng nhận một số các inputs, và outputs giá trị logic tổng hợp được.



Bài toán “P versus NP”



Tôi không thể tìm được một thuật toán hiệu quả bởi vì không thể có một thuật toán nào như vậy.



nếu bạn chứng minh được SAT là intractable
(chứng minh intractability có thể khó như
việc tìm lời giải hiệu quả)

Tôi không thể tìm được một thuật toán hiệu quả bởi vì tất cả những người nổi tiếng này cũng không tìm được nó.



nếu bạn biết SAT là NP-complete

Độ phức tạp tính toán—Sự tồn tại các bài toán giải được nhưng vô cùng khó giải



- Độ phức tạp tính toán: **P** (thời gian đa thức) và **non-P** (thời gian hàm mũ). Bài toán kiểu **P** có thể giải dễ dàng (sắp xếp dãy số theo thứ tự), bài toán kiểu **non-P** rất khó giải (tìm các thừa số nguyên tố của một số nguyên cho trước).
- Người ta tin rằng có rất nhiều bài toán thuộc kiểu **non-P**, nhưng chưa bao giờ chứng minh được chính chúng là như vậy (hết sức khó).
- **NP** (Nondeterministic Polynomial) là một họ đặc biệt các bài toán kiểu **non-P**: nếu bất kỳ trong chúng có nghiệm thời gian đa thức thì tất cả sẽ có nghiệm thời gian đa thức.
- **P = NP?** Các bài toán kiểu **P** và **NP** là như nhau?

Thời gian đa thức và hàm mũ



Time complexity function	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$
n	0.0001 second	0.0002 second	0.0003 second	0.0004 second	0.0005 second	0.0006 second
n^2	0.001 second	0.002 second	0.003 second	0.004 second	0.005 second	0.006 second
n^3	0.01 second	0.02 second	0.03 second	0.04 second	0.05 second	0.06 second
n^5	1 second	3.2 second	24.3 second	1.7 minutes	5.2 minutes	13.0 minutes
2^n	0.01 second	1.0 second	17.9 second	12.7 days	35.7 years	336 centuries
3^n	0.059 second	58 minutes	6.5 years	3855 centuries	2×10^8 centuries	1.3×10^{13} centuries

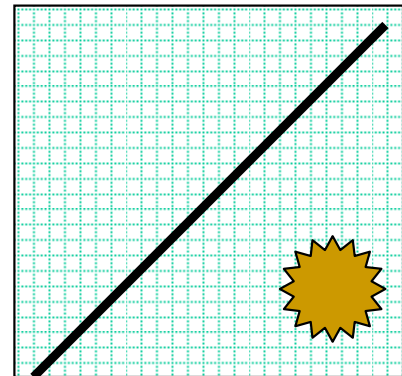
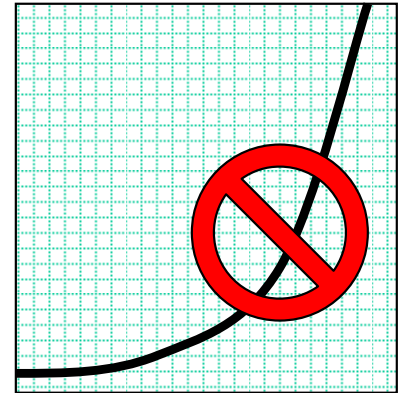
Scalable algorithms



- Điều gì xảy ra khi dữ liệu lớn, $N > 10^6$?
- Cần thuật toán heuristic nhanh, gần đúng
 - Fast CMB analysis by Szapudi et al (2001)

$O(N \log N)$ cần 1 ngày
 $O(N^3)$ cần 10 triệu năm

- Chú trọng đến giá của tính toán
 - Cần luôn điều khiển được độ chính xác
 - Đạt kết quả tốt nhất trong thời gian và tài nguyên cho phép
- Dùng máy tính song song

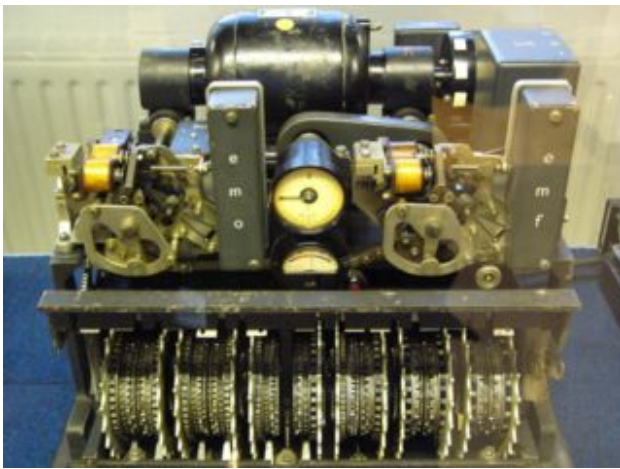


**Polynomial time
algorithms
do not always work!**

Mật mã và an toàn thông tin



- Là nghiên cứu về bí mật của truyền tin (truyền tin trong điều kiện có kẻ địch).
- An toàn mạng và máy tính: quản lý sự truy nhập máy tính và tin cậy của thông tin, và các ứng dụng như: ATM cards, computer passwords, e-commerce, ...



Germany Lorenz cipher machine



Mật mã và an toàn thông tin



- Tạo mã (encryption: plaintext $\rightarrow$ ciphertext) và giải mã (decryption)
- Khái niệm: mật mã (cipher: letter, bit), mã (code: word or phrase meaning), khóa (key). Rất nhiều kỹ thuật:
 - Symmetric-key cryptography (cùng một key cho nhận/gửi)
 - Public-key cryptography 1976 (public/private key) dựa trên number theory (integer factorization)
 - Cryptanalysis (tìm điểm yếu, không an toàn trong một hệ dùng mật mã)
 - Cryptographic primitives
 - Cryptographic protocols

(Khóa là một đoạn tin (piece of information) kiểm soát hoạt động của một thuật toán tạo mật mã, chữ ký điện tử, định quyền hạn (encryption, digital signature, authentication)).

Public-key cryptography

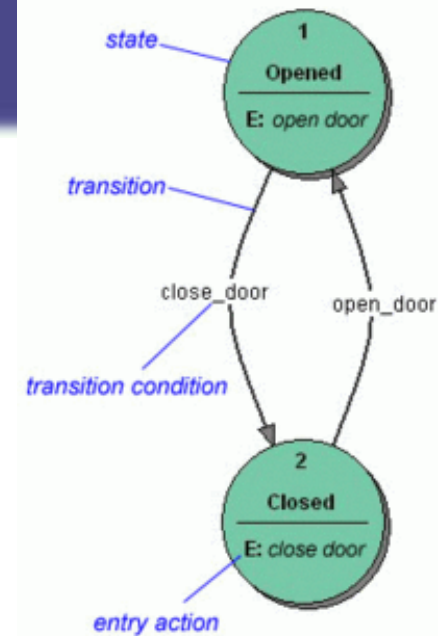


- **integer factorization** là quá trình phân tích một chữ số đa hợp thành tích của các ước số nhỏ hơn sao cho khi nhân các ước số này ta được số ban đầu.
- BSI team: Nov. 2005, RSA-640 (193 decimal digits, 640 bits) on 80 AMD Opteron CPUs computer.
- Khó nhất trong các bài toán này là trường hợp ước số là các số nguyên tố được chọn ngẫu nhiên với cùng độ lớn. Vẫn chưa có thuật toán thời gian đa thức để phân tích một số lớn b -bit thành tích của hai số nguyên tố có cùng kích thước.
- Một trong các bài toán lớn chưa giải được trong KHMT và lý thuyết số là việc tìm một thuật toán để nhân tử hóa các số trong thời gian đa thức.

Finite state machines

Máy hữu hạn trạng thái

- Là mô hình của các hành vi tạo nên bởi một số hữu hạn các trạng thái (states), các chuyển đổi trạng thái (transitions), và các hành động.
- Mô hình toán học: $(\Sigma, S, s_0, \delta, F)$,
- Việc thực hiện một phần cứng máy tính đòi hỏi một công-tơ (register) để chứa các biến trạng thái, một khối (block) mạch logic xác định sự chuyển trạng thái, và một khối thứ hai các mạch logic xác định output của một FSM.



Machine learning and data mining

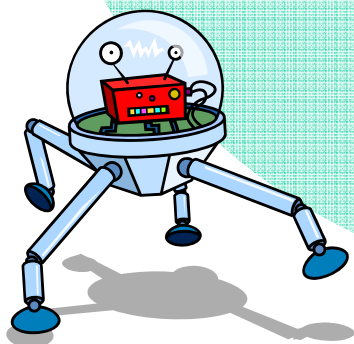
Học máy và Khai phá dữ liệu



■ Machine learning

Xây các hệ máy tính có thể học như con người (một số khả năng).

- ICML since 1982
ECML since 1989
- ECML/PKDD since 2001



■ Data mining

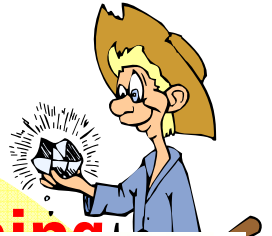
Tìm tri thức mới và hữu ích từ những cơ sở dữ liệu lớn.

- ACM KDD since 1995, PKDD and PAKDD since 1997, IEEE ICDM and SIAM DM since 2000, etc.



Machine learning and data mining

Học máy và Khai phá dữ liệu

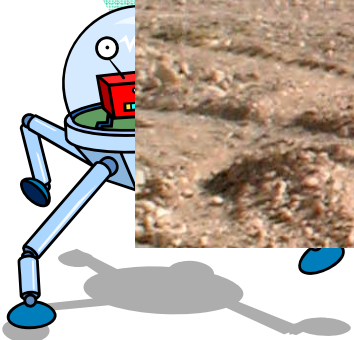


■ Machine learning

■ Data mining



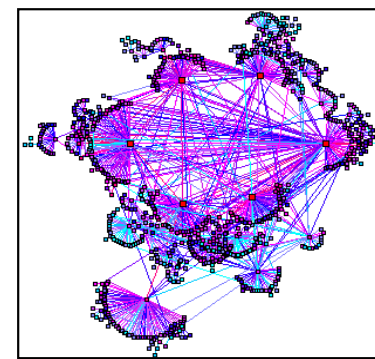
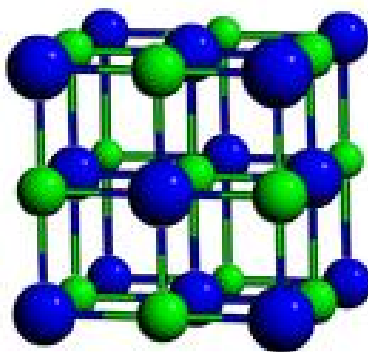
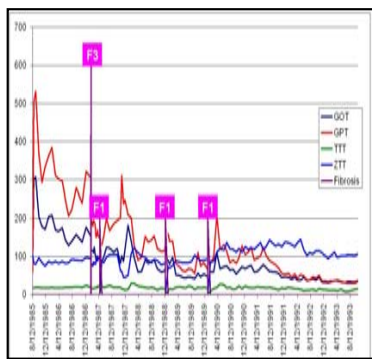
- 1997: Deep Blue beats Kasparov (12.2006: Deep Fritz beats Kramnik)
- 1997: Robot cup
- 2005: DARPA Grand Challenge



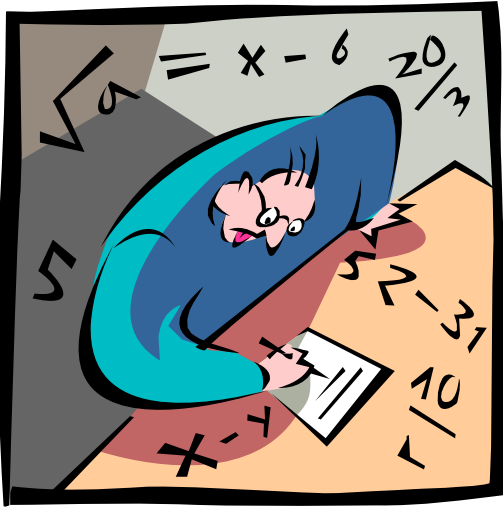
Complexly structured data

- Rất nhiều dữ liệu được thu thập và tích lũy trong các cơ sở dữ liệu lớn (hàng triệu bản ghi, hàng nghìn thuộc tính).
- Rất nhiều dữ liệu là loại được cấu trúc phức tạp (không ở dạng vectơ).

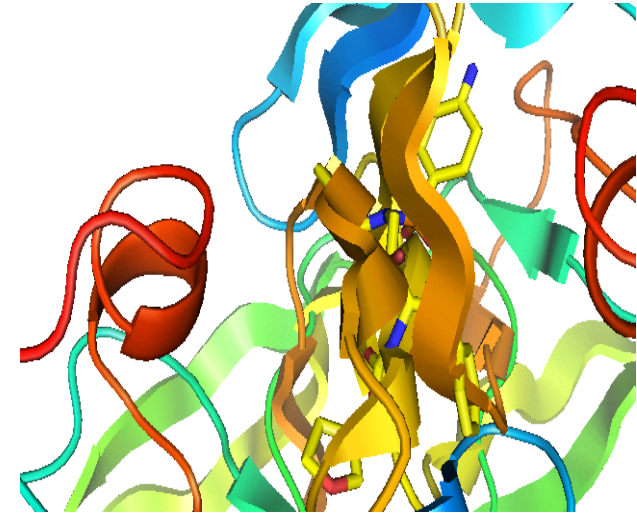
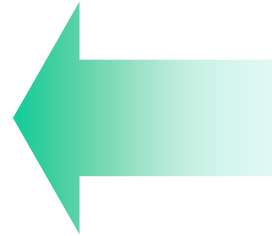
Thách thức: Tìm thuật toán khả cỡ để xử lý dữ liệu lớn và cấu trúc phức tạp?



Hai con đường của toán trong tin học



Các lý thuyết, mô hình toán học làm cơ sở cho sự phát triển tin học.



Để giải quyết các vấn đề của tin học và ứng dụng tin học, tìm và dùng các lý thuyết và công cụ toán học.

- Vài lĩnh vực tiêu biểu của toán học trong tin học
- Toán học trong xử lý ngôn ngữ tự nhiên
- Toán học và Web
- Toán học trong sinh học
- Toán học trong học máy (machine learning)

Natural language processing (NLP)



Lexical / Morphological Analysis

Tagging (gán nhãn từ loại)

Chunking (phân cụm từ)

Syntactic Analysis

Grammatical Relation Finding

Named Entity Recognition

Word Sense Disambiguation

Semantic Analysis

Reference Resolution

Discourse Analysis

text

Shallow parsing

The woman will give Mary a book

POS tagging

The/Det woman/NN will/MD give/VB
Mary/NNP a/Det book/NN

chunking

[The/Det woman/NN]_{NP} [will/MD give/VB]_{VP}
[Mary/NNP]_{NP} [a/Det book/NN]_{NP}

relation finding

subject

[The woman] [will give] [Mary] [a book]

i-object

object

meaning

Ông già đi nhanh quá

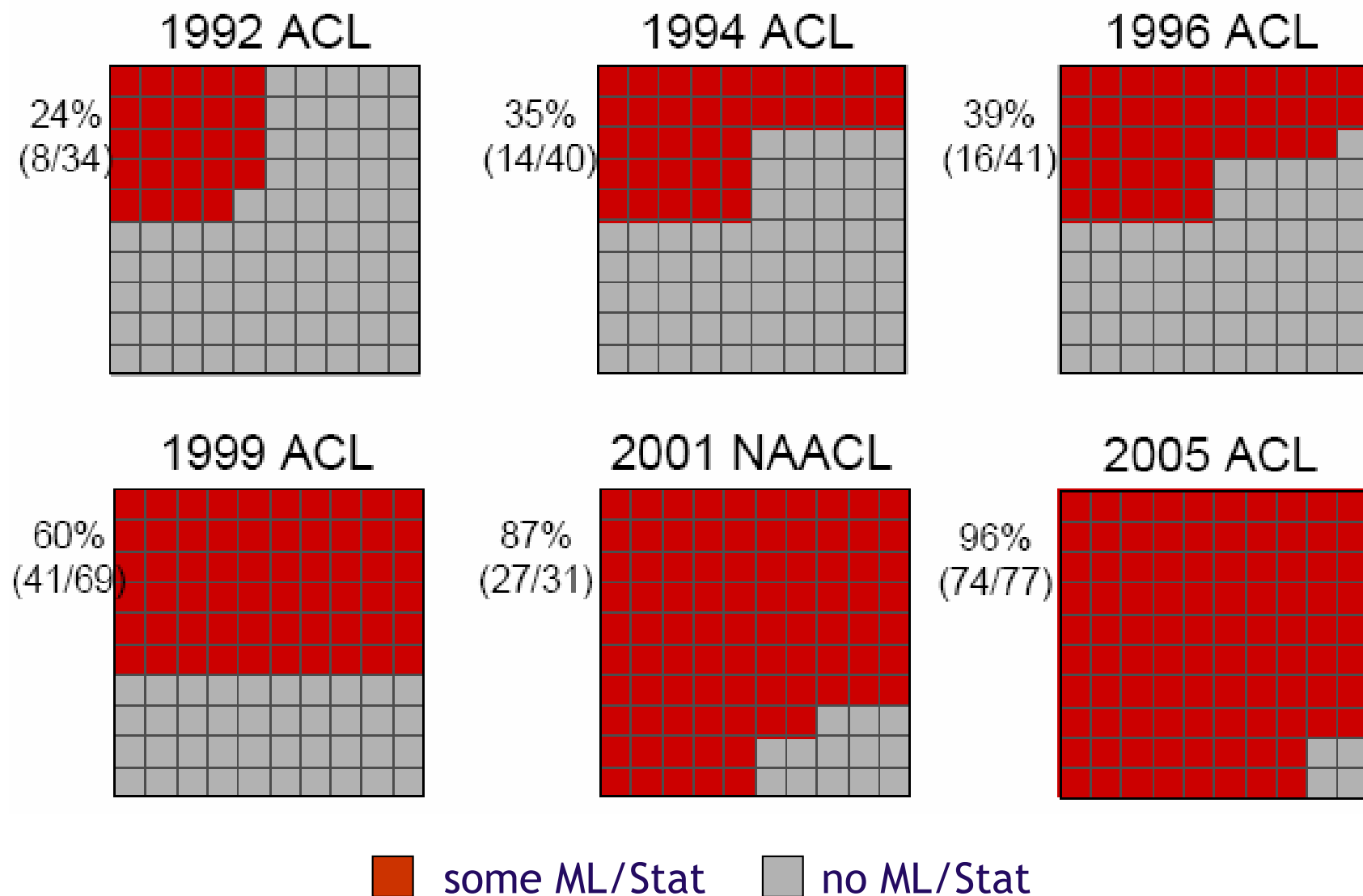
l o a n h o c v a T i n h o c

Xu thế trong NLP



- 1990s–2000s: Statistical learning
 - algorithms, evaluation, corpora
 - 1980s: Standard resources and tasks
 - Penn Treebank, WordNet, MUC
 - 1970s: Kernel (vector) spaces
 - clustering, information retrieval (IR)
 - 1960s: Representation Transformation
 - Finite state machines (FSM) and Augmented transition networks (ATNs)
 - 1960s: Representation—beyond the word level
 - lexical features, tree structures, networks
- Trainable parsers
- Trainable FSMs
-

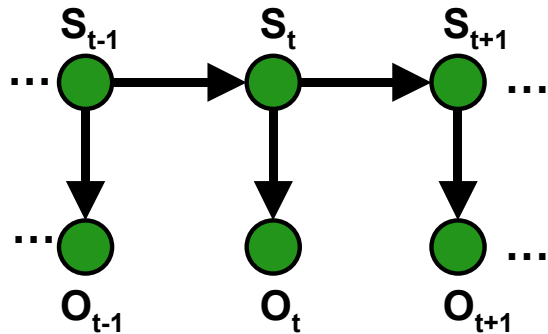
Machine learning and statistics in NLP



(Marie Claire, ECML/PKDD 2005)

Toán học và Tin học

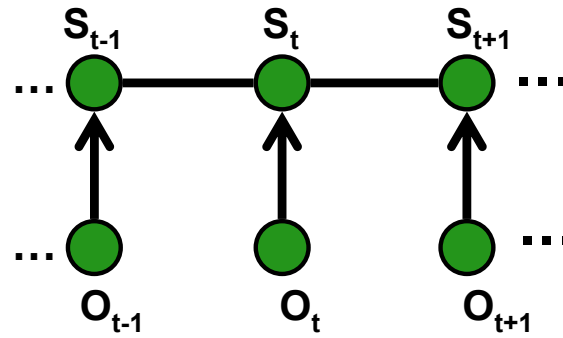
Trainable finite state machines



Hidden Markov Models (HMMs)

[Baum et al., 1970]

- Generative
- Need independence assumption
- Local optimum
- Local normalization

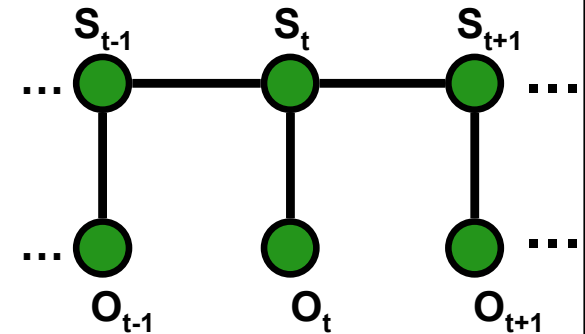


Maximum Entropy Markov Models (MEMMs)

[McCallum et al., 2000]

- Discriminative
- No independence assumption
- Global optimum
- Local normalization

More accurate than HMMs



Conditional Random Fields (CRFs)

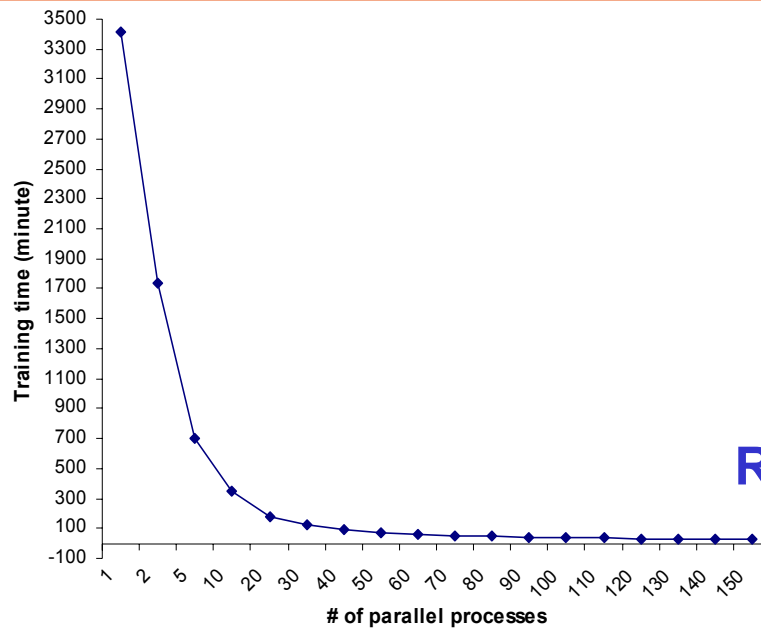
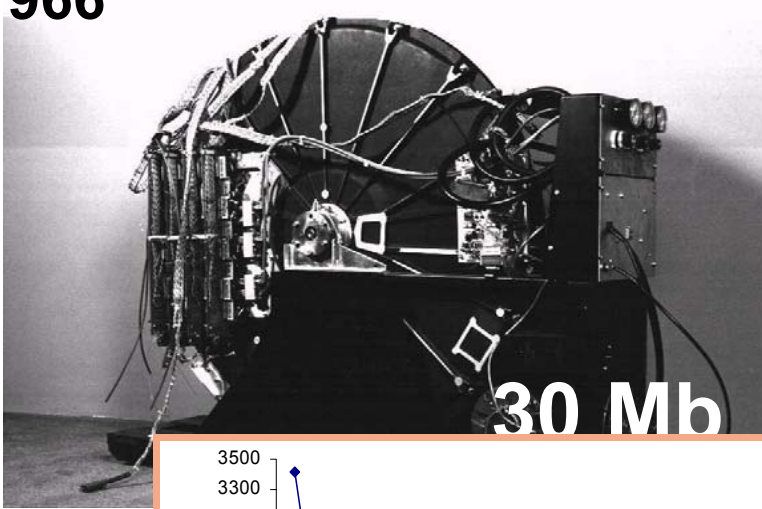
[Lafferty et al., 2001]

- Discriminative
- No independence assumption
- Global optimum
- Global normalization

More accurate than MEMMs

Increasing computing power

1966



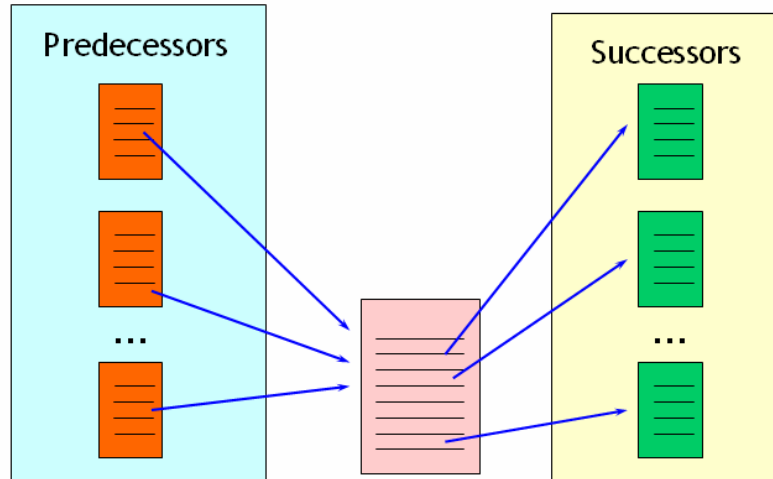
JAIST's CRAY XT3
Linux OS, 180 AMD Opteron 2.4GHz CPUs,
8Gb RAM/CPU, total: **1.44Tb** RAM

Second order of Conditional
Random Fields (Text chunking):

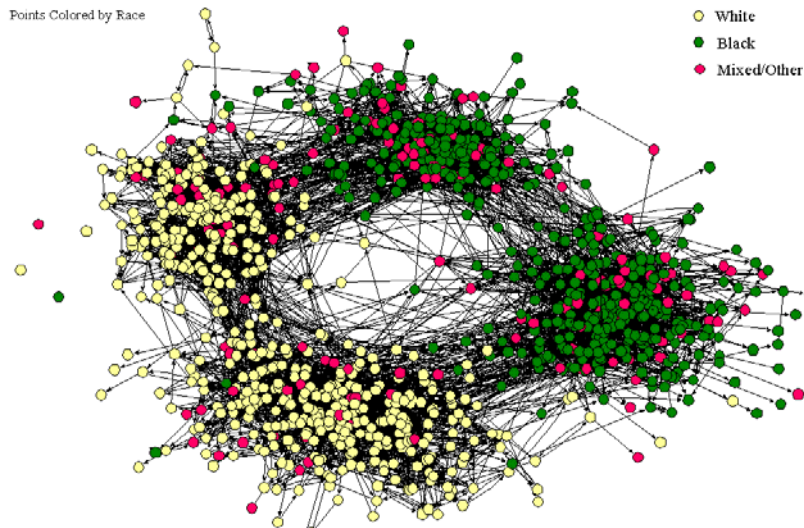
All phrase chunking, 17h46' on
90 processes vs. 1348h

- Vài lĩnh vực tiêu biểu của toán học trong tin học
- Toán học trong xử lý ngôn ngữ tự nhiên
- **Toán học và Web**
- Toán học trong sinh học
- Toán học trong học máy (machine learning)

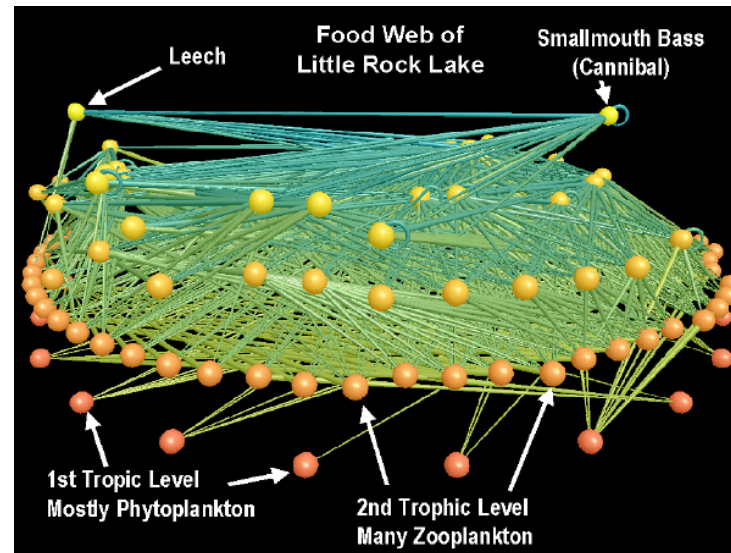
Web link data



The Social Structure of "Countryside" School District

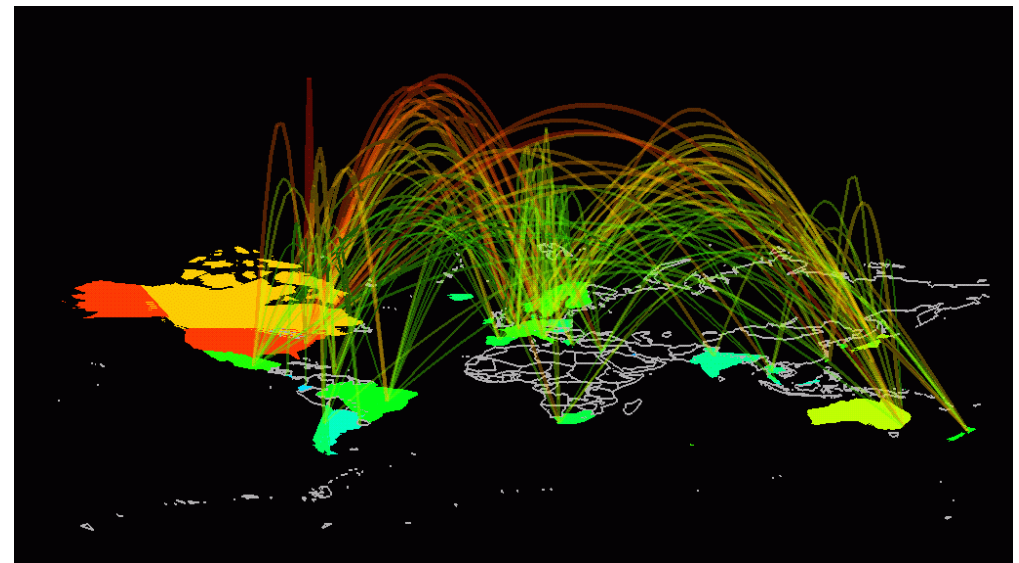


Friendship Network
[Moody '01]



Food Web
[Martinez '91]

Over 3 billion
documents



Hai cách tiêu biểu (1998):

1. **Page rank**: tìm các trang Web quan trọng và liên quan nhất trên Web (như trong Google)
2. **Hubs and authorities**: đánh giá chi tiết hơn tầm quan trọng của các trang Web.



Larry Page, Sergey Brin



Jon Kleinberg
awarded
Nevanlinna Prize

PageRank: key idea



- Google thường đưa ra một số rất lớn các trang Web ('weather forecast' → 5.5 million pages). Các trang liên quan nhất thường nằm trong một, hai chục trang đầu.
- Làm sao search engine biết được các trang Web nào là quan trọng nhất?
- Google gán cho mỗi trang Web một con số đặc trưng tầm quan trọng. Con số này được gọi là số PageRank và được tính như giá trị riêng (eigenvalue) của bài toán

$$Pw = \lambda w$$

ở đây P được tính dựa trên link structure của Internet. Vấn đề cơ bản là làm sao thiết lập xác đáng được link structure của toàn bộ Web, i.e., ma trận P .

PageRank: key idea



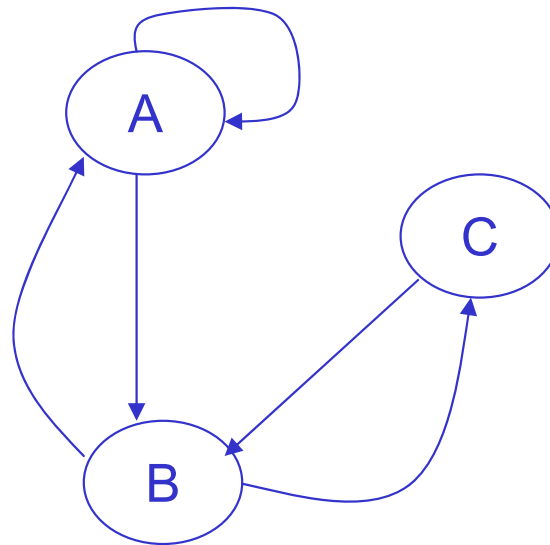
- Mô hình cơ sở của thuật toán PageRank là random walk thực hiện trên toàn bộ các trang Web trên Internet.
- Ký hiệu $p_t(x)$ khả năng ta ở tại trang Web x ở thời điểm t . Số PageRank của x được các định như $\lim(p_t(x))$ khi $t \rightarrow \infty$.
- Ma trận P có tính chất không rút gọn được (irreducible) và ngẫu nhiên (stochastic), nên random walk có thể biểu diễn qua chuỗi Markov, và PageRank của mọi trang có thể tính được qua các vectors riêng của P .

PageRank: Stochastic matrix



- Mỗi trang i tương ứng với dòng i và cột i của P
- Nếu trang j có n successors (links) và trang i là một trong số n successors của j thì ô ij^{th} của ma trận được là to $1/n$, nếu không là 0.

Assume the Web consists of only three pages A, B, and C. The links among these pages are shown as in the graph.



	A	B	C
A	$1/2$	$1/2$	0
B	$1/2$	0	1
C	0	$1/2$	0

The PageRank algorithm



- Ma trận Google P hiện có kích thước 4.2×10^9 và do đó việc tìm giá trị riêng là không tầm thường.

```
w0 = initial guess
For k = 1 to 50
    wk = P*wk-1
End
Return w50
```

- Tìm xấp xỉ các véc tơ riêng của P (power method)
- Việc tính toán (matrix-vector multiplications) phải thực hiện trên các hệ máy tính song song lớn.

PageRank: Example



$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

w_0 = initial guess
For $k = 1$ to 50
 $w_k = P * w_{k-1}$
End
Return w_{50}

Bắt đầu cho $w_1 = w_2 = w_3 = 1$, và tính đệ quy phép nhân ma trận với các giá trị này ...

a	$=$	1	1	$\frac{5}{4}$	$\frac{9}{8}$	$\frac{5}{4}$	...	$\frac{6}{5}$
b	$=$	1	$\frac{3}{2}$	1	$\frac{11}{8}$	$\frac{17}{16}$	...	$\frac{6}{5}$
c	$=$	1	$\frac{1}{2}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{11}{16}$	...	$\frac{3}{5}$

Finding “things” but not “pages”

Information Extraction vs. Information Retrieval



Information extraction: the process of extracting text segments of semi-structured or free text to fill data slots in a predefined template

foodscience.com-Job2

JobTitle: Ice Cream Guru

Employer: foodscience.com

JobCategory: Travel/Hospitality

JobFunction: Food Services

JobLocation: Upper Midwest

Contact Phone: 800-488-2611

DateExtracted: January 8, 2004

Source: www.foodscience.com/jobs_midwest.html

OtherCompanyJobs: foodscience.com-Job1

Ice Cream Guru

If you dream of cold cream, chocolate or cookie cookie, there's a great opportunity for you to maintain and expand this major corporation's high-end ice cream brand. Will be based in the Upper Midwest for about a year. After that, California here I come! Requires a BS in Food Science or dairy, plus ice cream formulation experience. Will consider entry level with an MS and an internship. Contact Susan: e-mail 1-800-488-2611

nomists will be hosted by AEAweb at the URL below:

インターネット

- Vài lĩnh vực tiêu biểu của toán học trong tin học
- Toán học trong xử lý ngôn ngữ tự nhiên
- Toán học và Web
- **Toán học trong sinh học**
- Toán học trong học máy (machine learning)

Problems in bioinformatics

Metabolomics

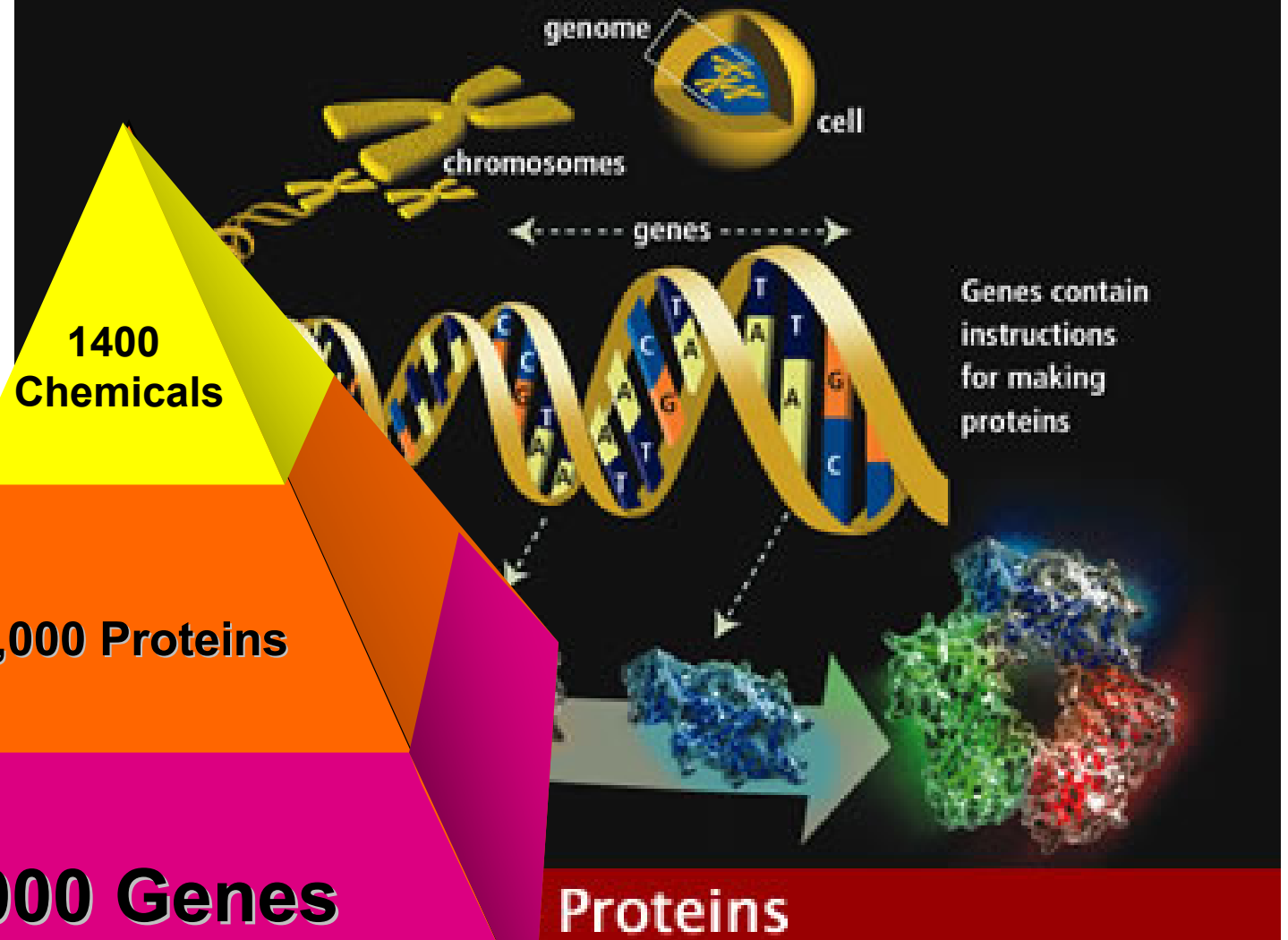
**1400
Chemicals**

Proteomics

10,000 Proteins

Genomics

30,000 Genes



Proteins

How biological data look like?



Một mẫu 350 chữ cái của dãy DNA 1.6 triệu chữ (nhỏ hơn 4570 lần):

```
...TACATTAGTTATTACATTGAGAACTTTATAATTA AAAAAGATTTCATG
TAAATTTCTTATTTGTTTATTTAGAGGTTTAAATTTAATTTCTAAGGGT
TTGCTGGTTTCATTGTTAGAATATTTAACTTAATCAAATTATTTGAATTT
TTGAAAATTAGGATTAATTAGGTAAGTAATAAATTTCTAAGAAATA
AGTTAAATTTTAAATTTAAGGA
GGAAAGAAAGATTTAAATACTA
CCTTAGAAAAATATGGGTATAGA
ATATTATGT...
```

```
function(ecoli962,5,1,4,'insB_4','IS1 protein InsB').
function(ecoli963,7,0,0,'b0989','cold shock-like protein').
function(ecoli964,7,0,0,'cspG','low-temperature-responsive gene analog of CspA and CspB homolog of Salmonella
cold shock protein').
function(ecoli965,7,0,0,'sfa','suppresses fabA and ts growth mutation').
function(ecoli966,0,0,0,'b0992','orf').
function(ecoli967,3,5,2,'torS','sensor protein tors (3rd module transmitter domain (?kinase) interacts with torr)').
function(ecoli968,3,5,2,'torT','part of regulation of tor operon periplasmic(1st module)').
function(ecoli969,3,5,2,'torR','response transcriptional regulator for torA (sensor TorS)(1st module)').
function(ecoli970,3,5,2,'torC','trimethylamine N-oxide reductase cytochrome c-type subunit also has activity as
negativer regulator of tor operon(1st module)').
function(ecoli971,3,5,2,'torA','trimethylamine N-oxide reductase subunit').
function(ecoli972,3,5,2,'torD','part of trimethylamine-N-oxide oxidoreductase').
function(ecoli973,0,0,0,'yccD','orf').
```

Nhiều loại dữ liệu khác

(Approximate) String Matching

Input: Text **T** , Pattern **P**

Question(s):

P xuất hiện trong **T**?

Tìm một xuất hiện của **P** trong **T**.

Tìm mọi xuất hiện của **P** trong **T**.

Tính số xuất hiện của **P** trong **T**.

Tìm dãy con dài nhất của **P** trong **T**.

Tìm dãy con gần nhất của **P** trong **T**.

Xác định các lặp trực tiếp của **P**
trong **T**.

và nhiều biến dạng khác

Applications:

Liệu **P** đã có trong cơ sở dữ liệu **T**?

Xác định vị trí của **P** trong **T**.

Liệu có thể dùng **P** như một nguyên
tố của **T**?

P có tương đồng với gì đó trong **T**?

P có bị hỏng bởi **T**?

Liệu prefix(**P**) = suffix(**T**)?

Xác định các lặp sau trước (tandem)
của **P** trong **T**.

Pairwise Sequence Alignment

Sắp dãy từng cặp



■ Input

- Hai dãy chữ cái
- Một cách cho điểm

■ Output

- Cách sắp thẳng dãy tối ưu

- ❖ Bài toán cơ bản nhất của tin sinh học
- ❖ Các dãy được sắp thẳng $\Rightarrow$ có dùng cấu trúc hoặc chức năng
- ❖ Cho nhiều gợi ý nếu cấu trúc và chức năng của một trong các dãy được sắp thẳng đã biết

ATTGCGC

ATTC~~C~~GC

→ ATTGCGC

→ ATCCGC

→

ATTGCGC

AT - CCGC

→

ATTGCGC

ATC - CGC

→

ATTGCGC

ATCCG - C

Pairwise Sequence Alignment

Sắp dãy từng cặp



- Phổ biến dùng HMM: Các cặp dãy được sắp do dùng các thuật toán vét cạn bởi quy hoạch động.
 - Độ phức tạp của các phương pháp vét cạn là $O(2^n m^n)$
 - $n = \text{số các dãy}$
- Sắp dãy bội xử dụng các phương pháp heuristic

#Rat	ATGGTGCACCTGACTGATGCTGAGAAGGCTGCTGT
#Mouse	ATGGTGCACCTGACTGATGCTGAGAAGGCTGCTGT
#Rabbit	ATGGTGCATCTGTCCAGT - - - GAGGAGAAGTCTGC
#Human	ATGGTGCACCTGACTCCT - - - GAGGAGAAGTCTGC
#Opposum	ATGGTGCACCTTGACTTTT - - - GAGGAGAAGAACTG
#Chicken	ATGGTGCACCTGGACTGCT - - - GAGGAGAAGCAGCT
#Frog	- - - ATGGGTTTGACAGCACATGATCGT - - - CAGCT

Đoán nhận gene Gene prediction



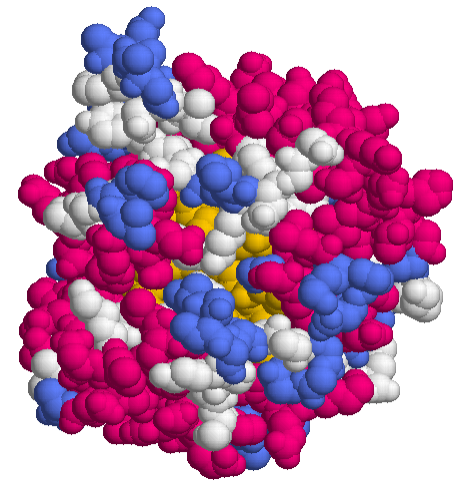
Là bài toán quan trọng của tin sinh học và hiện có nhiều thuật toán cho đoán nhận gene dựa trên các gene đã biết như dữ liệu huấn luyện. Một kỹ thuật toán nhận gene phổ biến là Hidden Markov Models (HMMs).

(given the genomic DNA sequence, can we tell where the genes are?)

Bài toán đoán nhận cấu trúc protein



- Có khoảng 15,000 cấu trúc protein trong các cơ sở dữ liệu công cộng, và trong số này rất nhiều cấu trúc giống nhau. Con người mới **biết** chừng 1,500 cấu trúc protein khác nhau.
- **Dự đoán cấu trúc protein** từ các dãy amino-acid là một trong các bài toán quan trọng nhất của tin sinh học, và con người còn đang cách lời giải rất xa.



Dự đoán tương tác proteins



DNA → RNA → protein

Sequence → Structure → Function

Interaction → Network → Function

DNA Sequence

Gene

Protein Structure

Function

7000 Yeast interactions
among 3000 proteins

Hầu hết các quá trình sinh học
đều liên quan đến tương tác
nào đó của proteins.

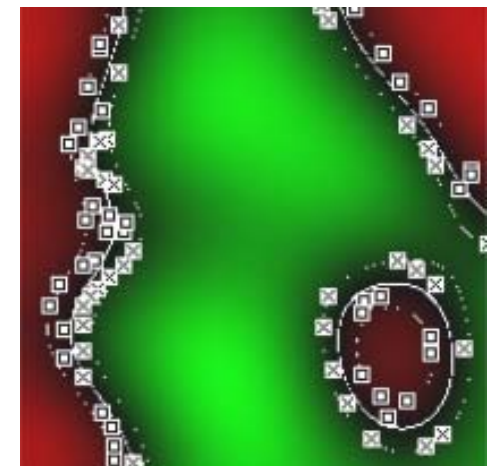
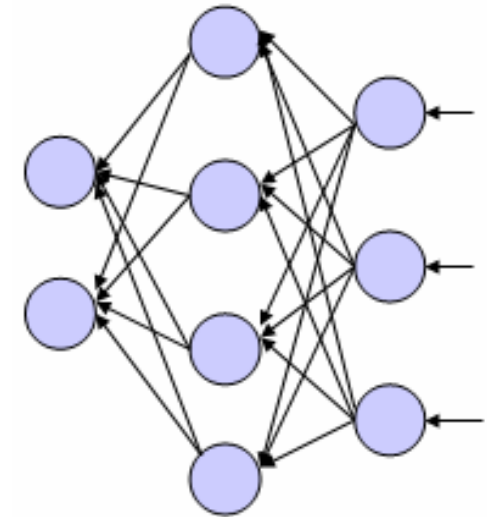
- Neural Networks
 - Sequence Encoding and Output Interpretation
 - Prediction of Protein Secondary Structure
 - Prediction of Signal Peptides and Their Cleavage Sites
 - Applications for DNA and RNA Nucleotide Sequences
- Hidden Markov Models
 - Protein Applications
 - DNA and RNA Applications
- Probabilistic Graph Models
- Probabilistic Models of Evolution
- Stochastic Grammars and Linguistics
- Kernel methods

- Vài lĩnh vực tiêu biểu của toán học trong tin học
- Toán học trong xử lý ngôn ngữ tự nhiên
- Toán học và Web
- Toán học trong sinh học
- Toán học trong học máy (machine learning)

Kernel methods: a bit of history



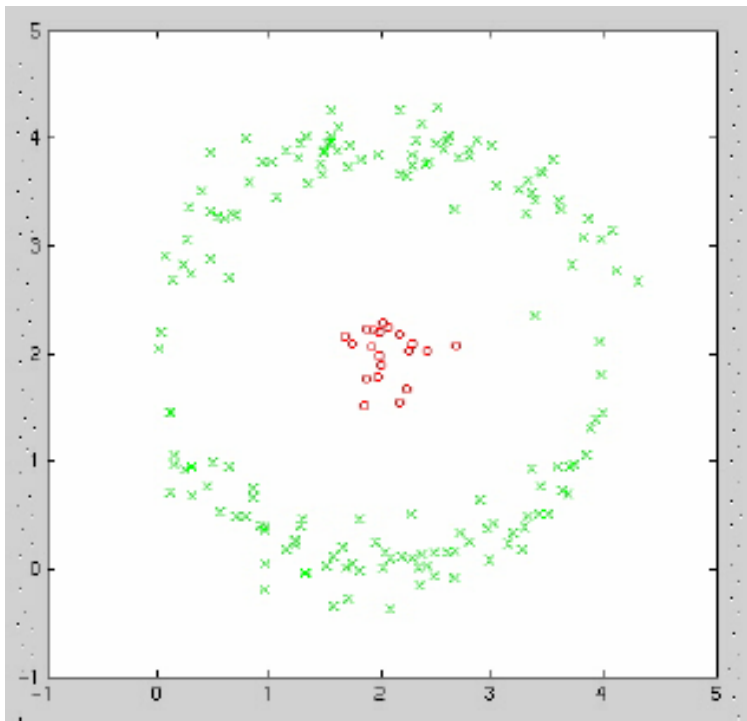
- Minsky and Pappert (1969) chỉ ra hạn chế của perceptrons.
- **Neural networks** (từ giữa 1980s) vượt qua các hạn chế bằng cách gắn vào nhau nhiều đơn vị tuyến tính (multi-layer neural networks). Gặp hạn chế của tốc độ và cực tiểu địa phương.
- **Kernel methods** (2000s) nối các hàm tuyến tính nhưng trong không gian feature với số chiều cao (high dimensional feature space.)



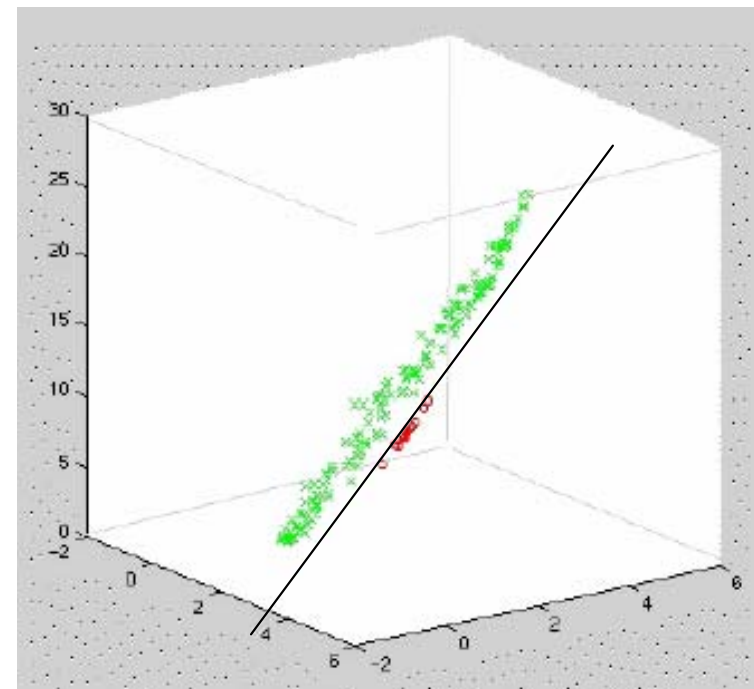
Kernel methods: the basic idea



Biến đổi dữ liệu vào không gian nhiều chiều hơn
có thể biến dữ liệu thành tách được tuyến tính.

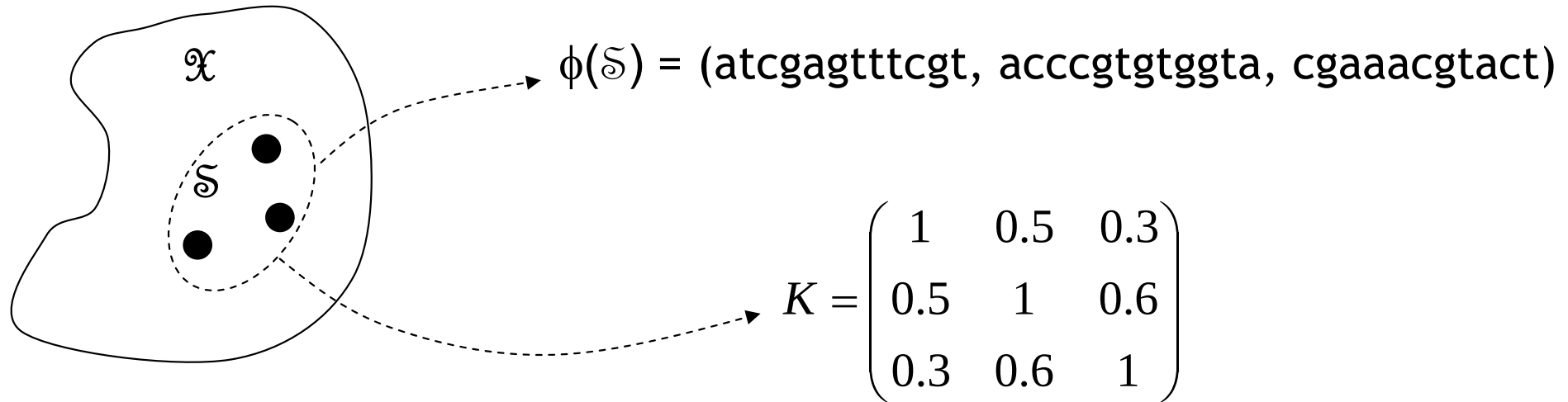


$$\phi: \mathcal{X} = \mathbb{R}^2 \rightarrow \mathcal{H} = \mathbb{R}^3$$



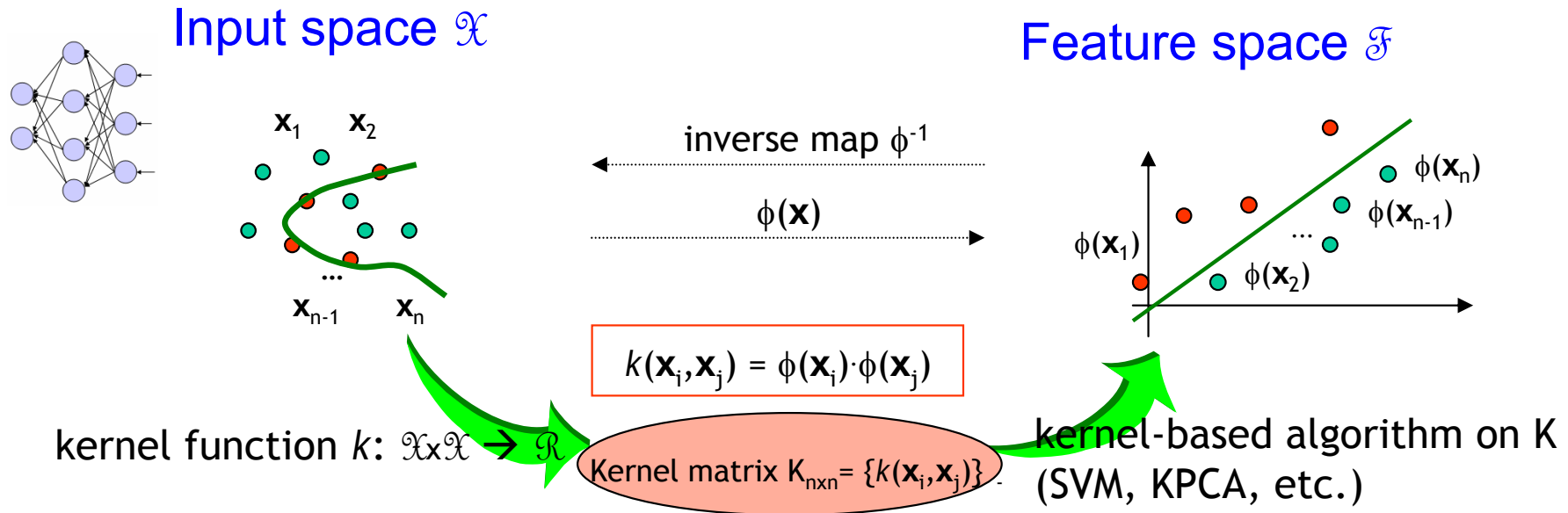
$$(x_1, x_2) \mapsto (x_1, x_2, x_1^2 + x_2^2)$$

Kernel idea: biểu diễn dữ liệu



- $\mathcal{X}$ là tập các oligonucleotides, $\mathcal{S}$ gồm ba oligonucleotides.
- Thông thường, mỗi oligonucleotide được biểu diễn bởi một dãy các chữ cái.
- Trong kernel methods, $\mathcal{S}$ được biểu diễn như một ma trận của độ đo sự tương tự giữa từng cặp phần tử.

Kernel methods: the scheme

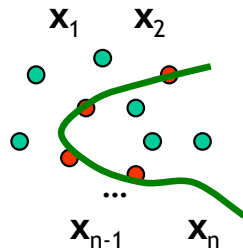


- Ánh xạ dữ liệu trong $\mathcal{X}$ vào một (high-dimensional) không gian vectơ, gọi là feature space $\mathcal{F}$, bởi một hàm ϕ trên từng cặp phần tử $\mathbf{x}$.
- Tìm các dạng tuyến tính (linear pattern) trong $\mathcal{F}$ với các thuật toán quan biết (tính toán trên kernel matrix).
- Khi dùng ánh xạ ngược, các dạng tuyến tính trong $\mathcal{F}$ có thể tìm được các dạng tương ứng phức tạp trong $\mathcal{X}$.
- Việc này được ngầm định thực hiện chỉ bởi nội tích (inner products) trong $\mathcal{F}$ (kernel trick) xác định bởi một hàm kernel

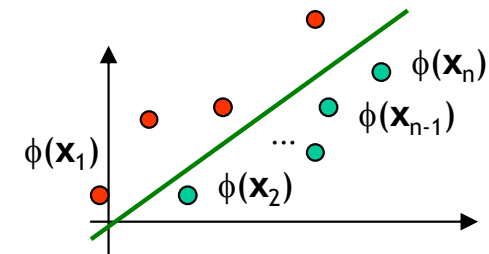
Kernel methods: math background



Input space $\mathcal{X}$



Feature space $\mathcal{F}$



inverse map ϕ^{-1}

$\phi(\mathbf{x})$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

kernel function $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$

Kernel matrix $K_{n \times n} = \{k(\mathbf{x}_i, \mathbf{x}_j)\}$

kernel-based algorithm on K
(SVM, KPCA, etc.)

Linear algebra, probability/statistics, functional analysis, optimization

- Mercer theorem: Mọi hàm xác định dương đều có thể viết như nội tích trong một không gian feature nào đấy.
- Kernel trick: Dùng ma trận kernel thay vì dùng nội tích trong không gian feature space.
- Representer theorem:

Every minimizer of $\min_{f \in \mathcal{H}} \{C(f, \{x_i, y_i\}) + \Omega(\|f\|_H)\}$ admits

a representation of the form $f(\cdot) = \sum_{i=1}^m \alpha_i K(\cdot, x_i)$

Support vector machines



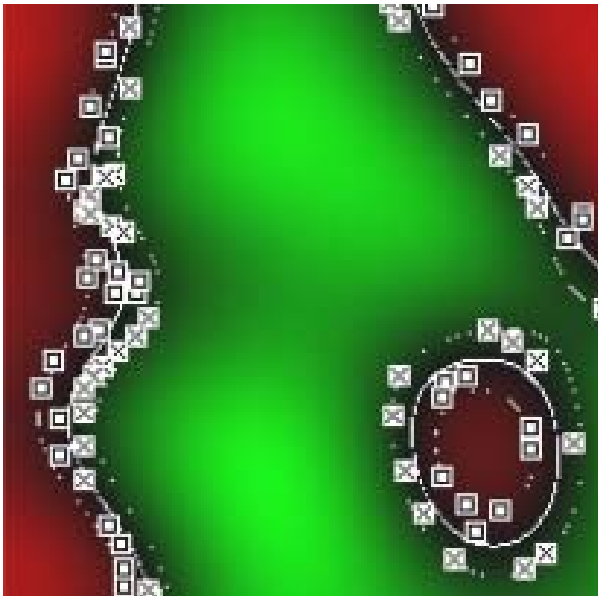
Soft margin problem

$$\min_{\mathbf{w}, b, \xi_1, \dots, \xi_n} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i$$
$$\forall i, \begin{cases} \xi_i \geq 0 \\ \xi_i - 1 + y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 0 \end{cases}$$

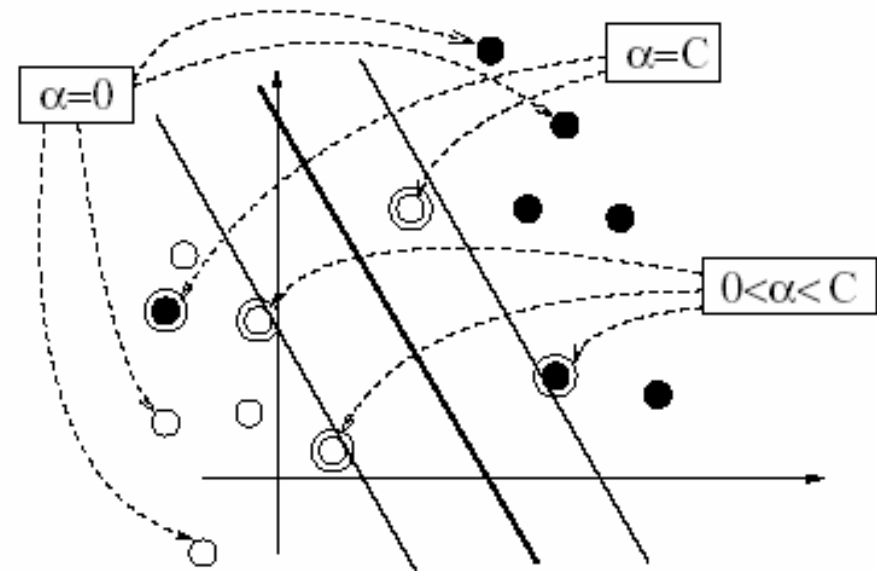
Equivalent to dual problem

$$W(\alpha) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^n \alpha_i$$
$$\begin{cases} \sum_{i=1}^n y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \quad \text{for } i = 1, \dots, n \end{cases}$$

Input space



Feature space



- Có mối quan hệ sâu sắc, rộng lớn giữa toán học và tin học.
- Hầu hết các vấn đề của khoa học máy tính đều cần đến toán học hiện đại ở mức cao hoặc rất cao.