Khoa Công Nghệ Thông Tin
Trường Đại Học Cần Thơ

Máy học véctơ hỗ trợ
Support vector machines

Đỗ Thanh Nghị
dtnghi@cit.ctu.edu.vn

Cần Thơ
02-12-2008

Nội dung

- Giới thiệu về SVM
- Giải thuật học của SVM
- Ứng dụng của SVM
- Kết luận và hướng phát triển
- Demo chương trình (20 – 30 phút)

- **Giới thiệu về SVM**
- Giải thuật học của SVM
- Ứng dụng của SVM
- Kết luận và hướng phát triển

Support vector machines?

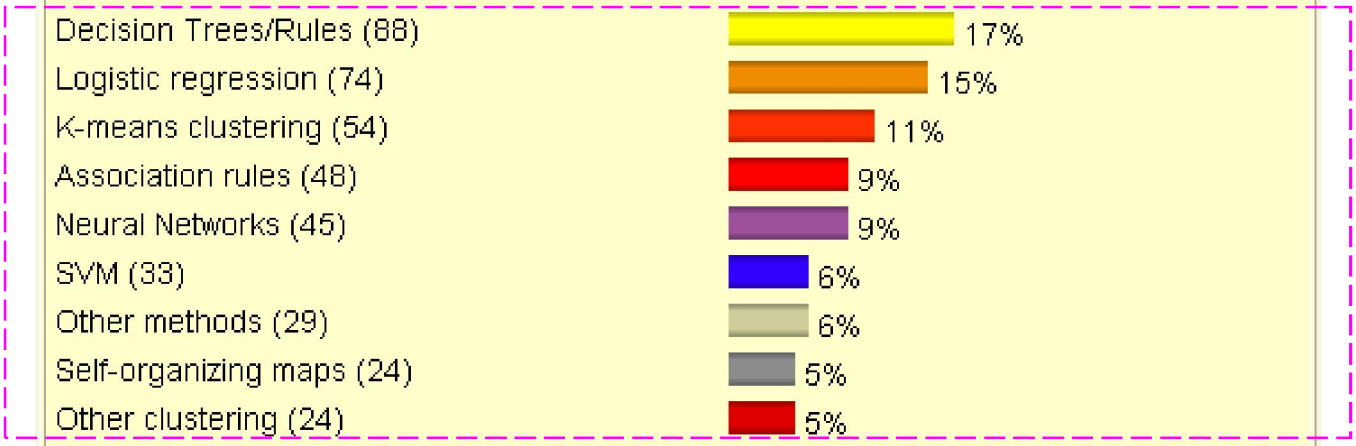
- lớp các giải thuật học
  - tìm siêu phẳng trong không gian *N-dim* để phân loại dữ liệu
  - SVM + hàm kernel = mô hình
  - giải thuật SVM = lời giải của bài toán quy hoạch toàn phương
  - tối ưu toàn cục
  - có nhiều giải thuật để giải
  - SVM có thể mở rộng để giải các vấn đề của hồi quy, gom nhóm, etc.
  - Được ứng dụng thành công : nhận dạng, phân tích dữ liệu, phân loại gien, ký tự, etc.

Kỹ thuật DM thành công trong ứng dụng thực (2004)

KDnuggets : **Polls** : Deployed data mining techniques

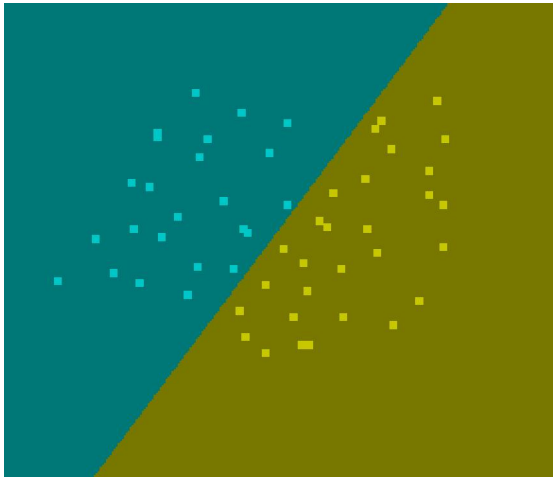
Poll

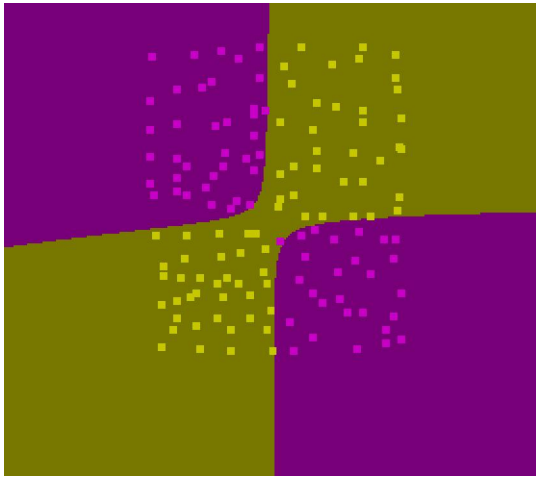
Which data mining techniques you used in a successfully deployed application? [173 voters, 509 votes total]

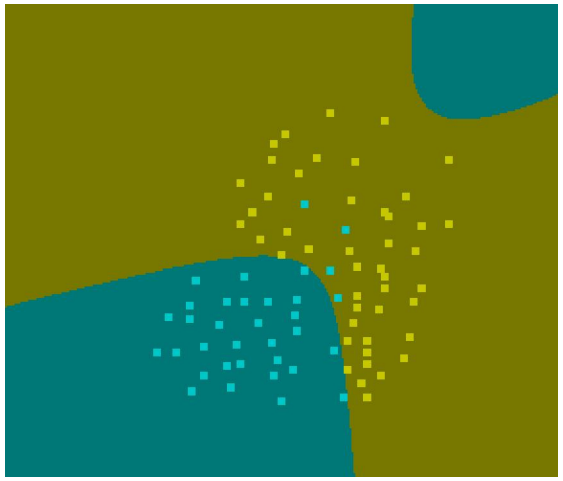


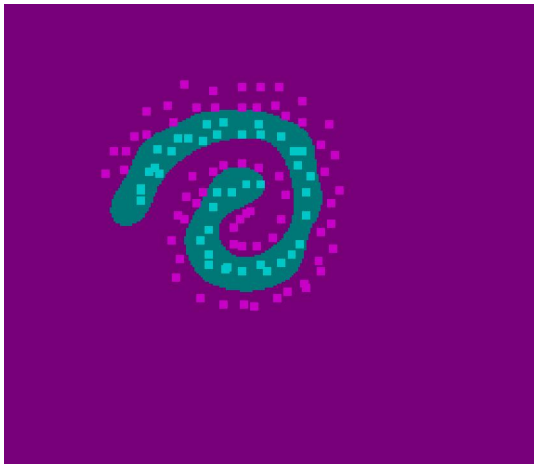
- Giới thiệu về SVM
- **Giải thuật học của SVM**
- Ứng dụng của SVM
- Kết luận và hướng phát triển

Support vector machines?

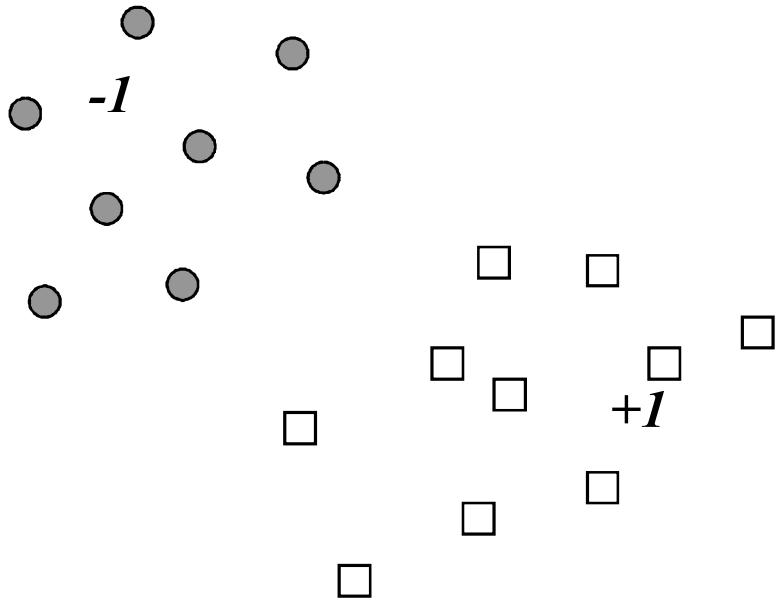




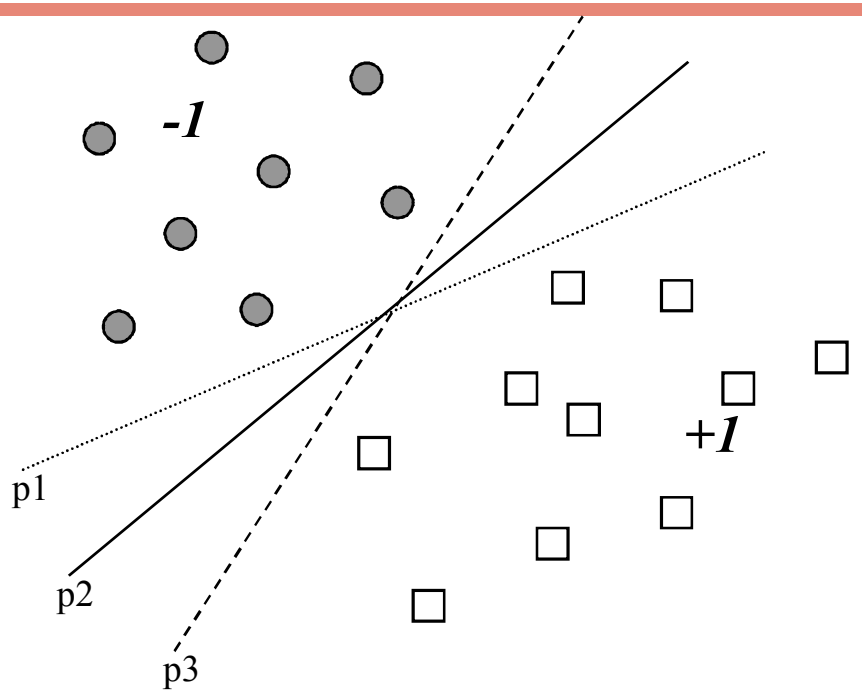




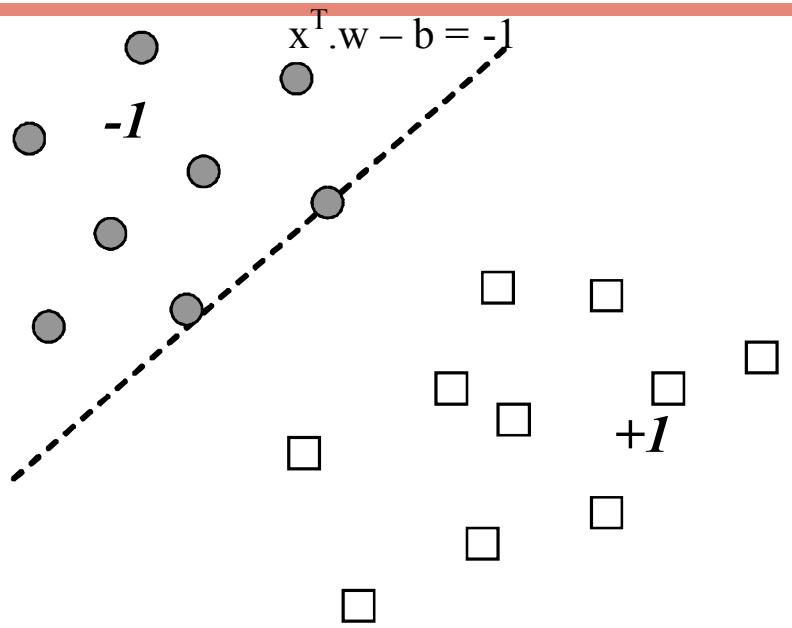
Support vector machines?



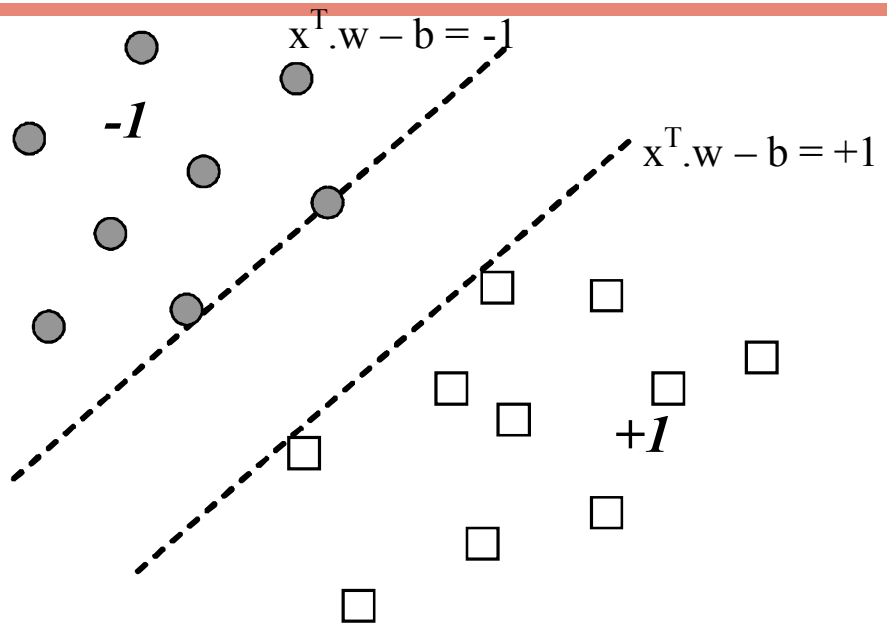
Support vector machines?



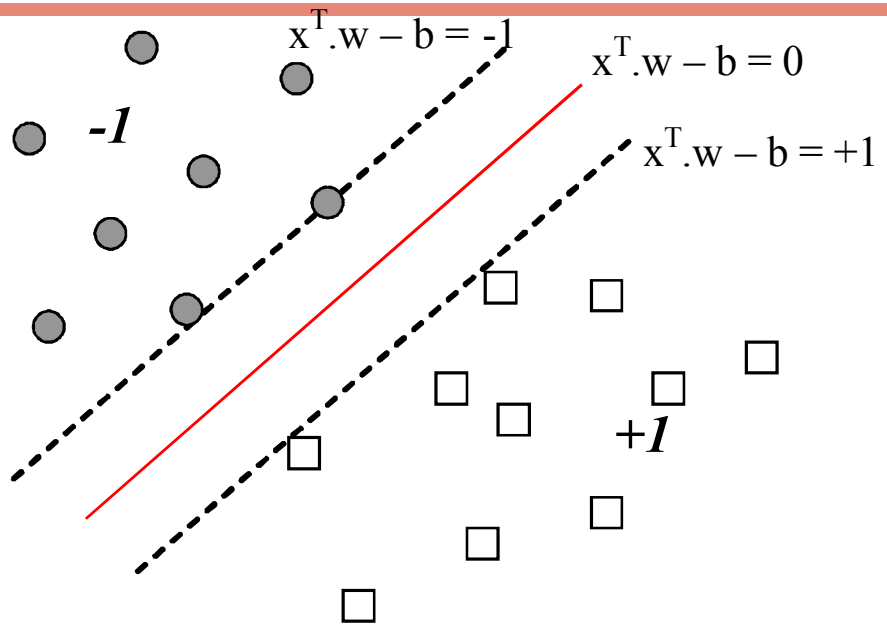
Support vector machines?



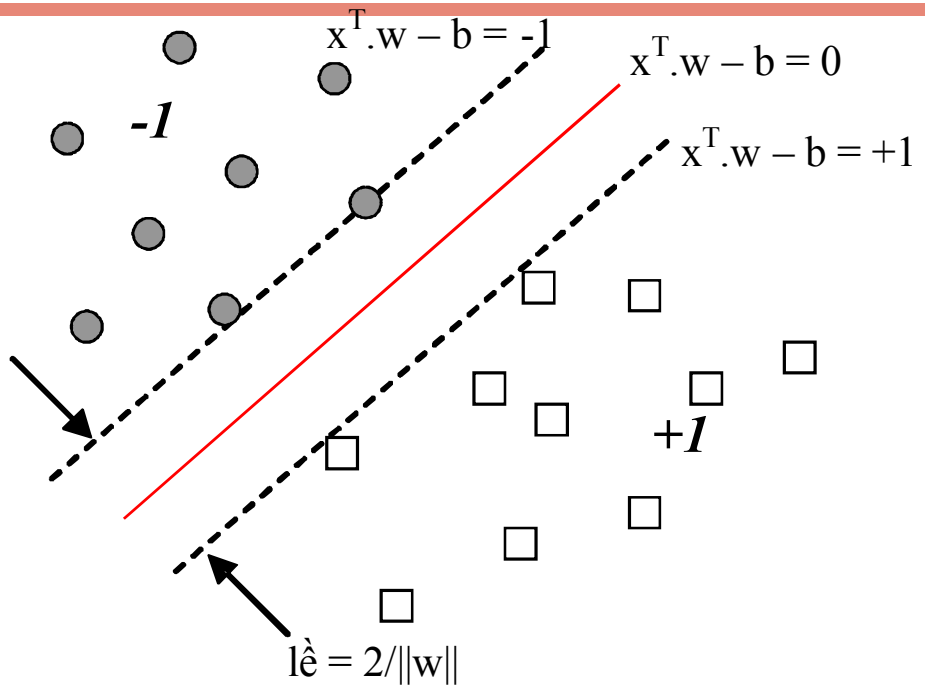
Support vector machines?



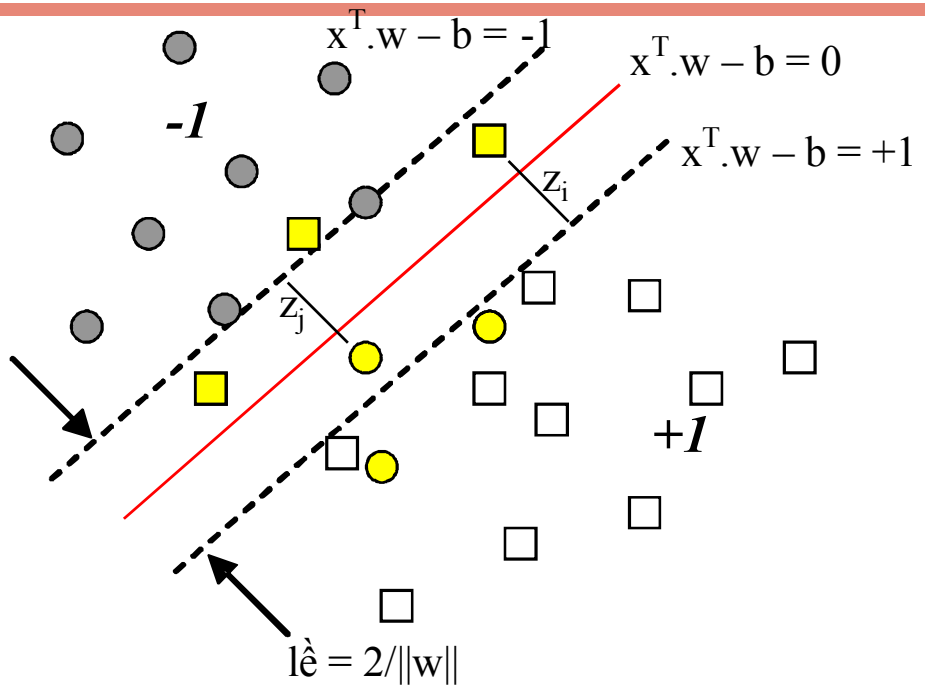
Support vector machines?



Support vector machines?



Support vector machines?



Phân loại với SVM

- cực đại hóa lề + cực tiểu hóa lỗi

$$min\ \Psi(w, b, z) = (1/2)\ ||w||^2 + c\sum_{i=1}^{m} z_i$$

$$y_i(w.x_i - b) + z_i \geq 1 \qquad (1)$$

$$z_i \geq 0 \quad (i=1, 2, \ldots, m)$$

- giải (1): w, b

- phân loại x: $sign(x.w - b)$

Phân loại với SVM

- bài toán đối ngẫu của (1):

$$min\ \Psi(\alpha) = (1/2) \sum_{i=1}^{m} \sum_{j=1}^{m} (y_i\ y_j\ \alpha_i\ \alpha_j\ x_i.x_j) \ - \sum_{i=1}^{m} \alpha_i$$

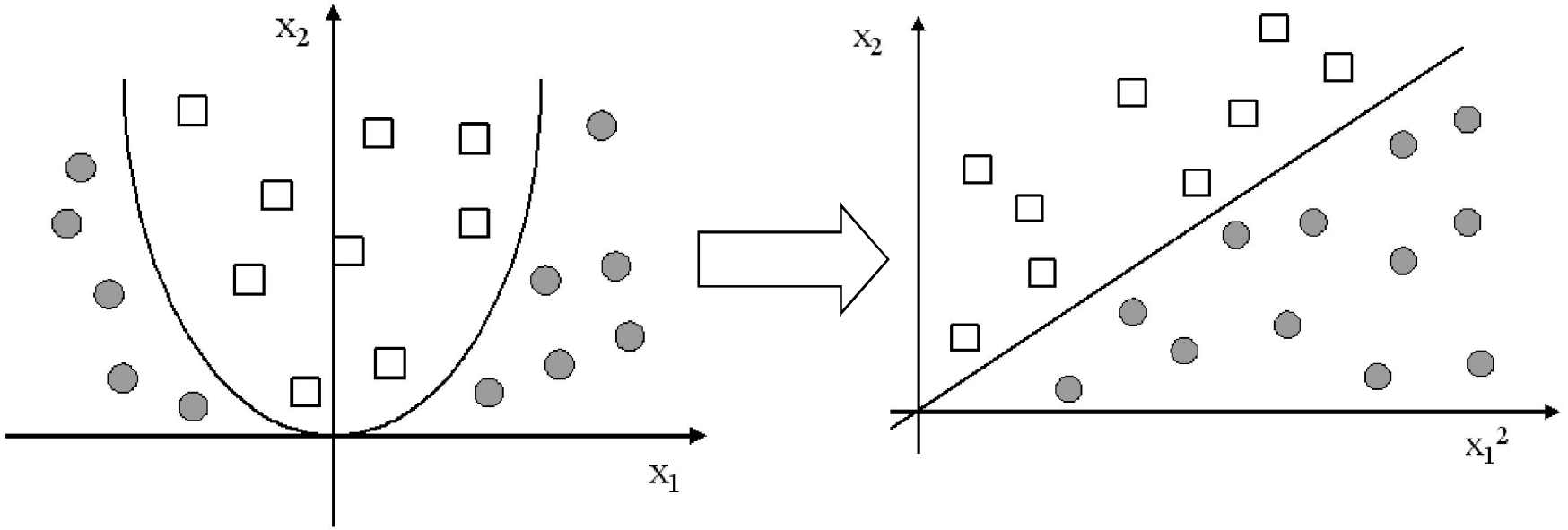
$$\sum_{i=1}^{m} y_i\ \alpha_i = 0 \qquad (2)$$

$$c \geq \alpha_i \geq 0$$

- giải (2): α_i
  - những x_i tương ứng với $\alpha_i > 0$ là véctơ hỗ trợ (SV)
  - tính *b* dựa trên tập SV
- phân loại *x*: $sign(\sum_{i=1}^{\#SV} y_i\ \alpha_i\ x_i.x\ - b)$

SVM phi tuyến

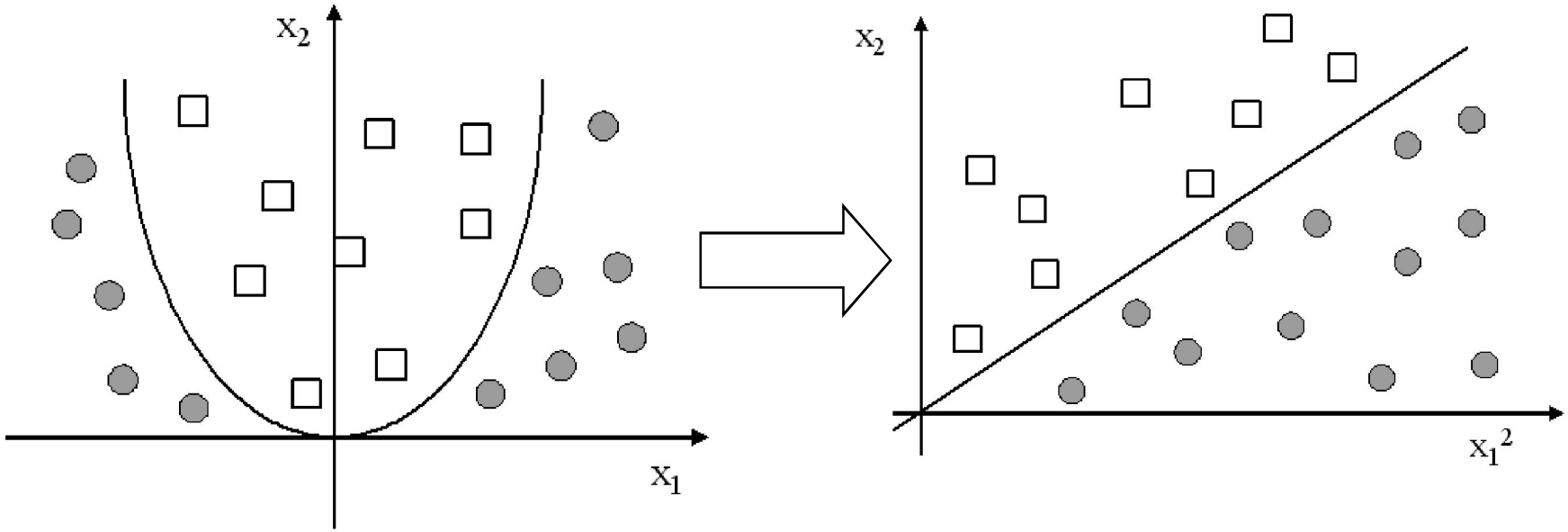
- không gian input: phi tuyến
- không gian trung gian: tuyến tính



- chuyển không gian input => không gian trung gian:
  - 2 chiều [x_1, x_2] không gian input => 5 chiều [x_1, x_2, x_1x_2, x_1^2, x_2^2] không gian trung gian (feature space)

SVM phi tuyến

- không gian input: phi tuyến
- không gian trung gian: tuyến tính



- chuyển không gian input => không gian trung gian:
  - phép chuyển đổi khó xác định, số chiều trong không gian trung gian rất lớn.
  - hàm kernel (Hilbert Schmidt space): làm việc trong không gian input nhưng ngầm định trong không gian trung gian

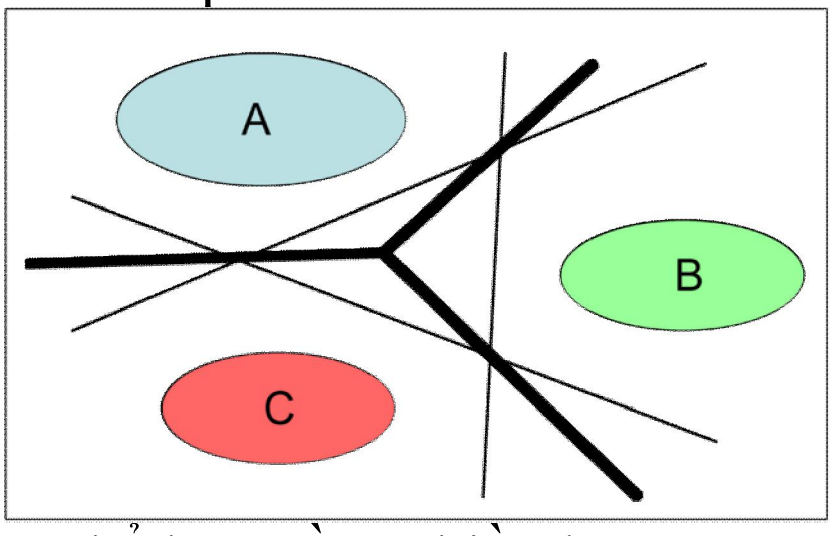
SVM phi tuyến sử dụng kernel

- SVM + kernel = mô hình
  - thay dot product bởi hàm kernel, $K(u, v)$
  - tuyến tính: $K(u, v) = u.v$
  - đa thức bậc d: $K(u, v) = (u.v + c)^d$
  - Radial Basis Function: $K(u, v) = exp(-2||u - v||^2/\sigma^2)$

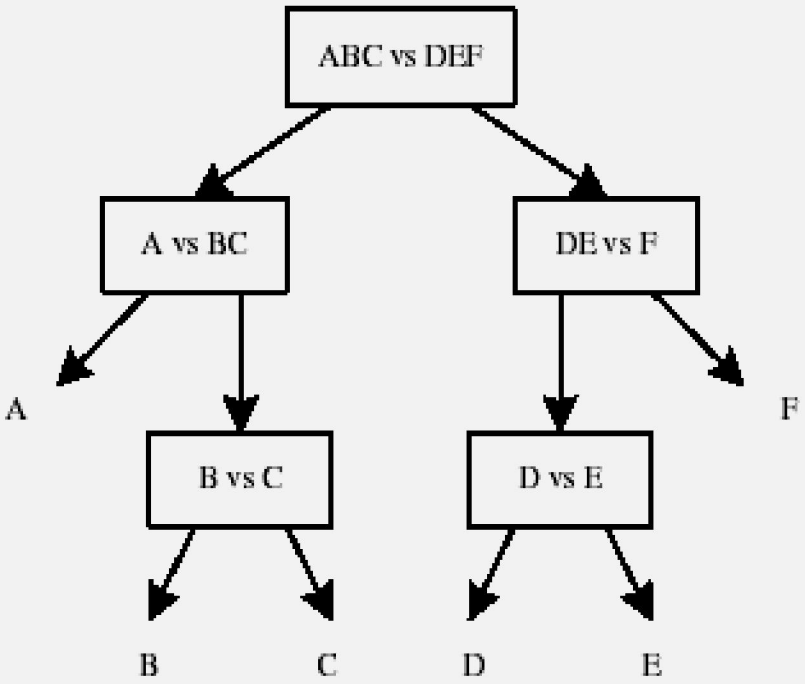
- ví dụ, hàm kernel đa thức bậc 5
  - dữ liệu có số chiều là 250 trong không gian input
  - tương đương với 10^{10} chiều trong không gian trung gian

SVM cho vấn đề nhiều lớp (> 2)

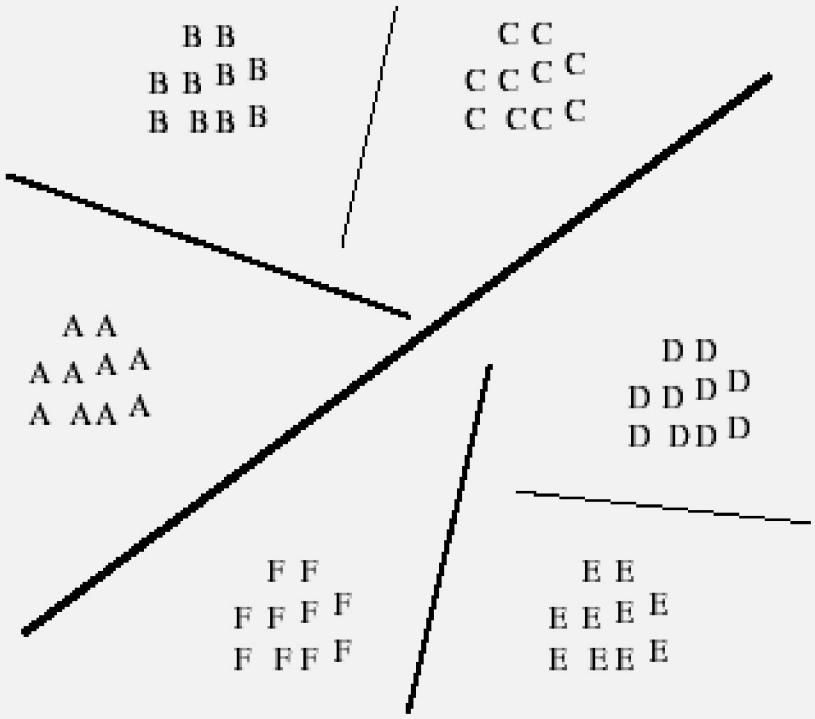
- xây dựng trực tiếp mô hình cho nhiều lớp
  - xây dựng bài toán tối ưu cho k lớp
- 1-tất cả (1 vs all)
  - mỗi mô hình phân tách 1 lớp từ các lớp khác
  - k lớp : k mô hình
- 1-1 (1 vs 1)
  - mỗi mô hình phân tách 2 lớp
  - k lớp : k*(k-1)/2 mô hình
- phương pháp khác
  - phân tách 2 nhóm, mỗi nhóm có thể bao gồm nhiều lớp
  - xác định cách phân tách nhóm sao cho có lợi nhất



SVM cho vấn đề nhiều lớp (> 2)



1)



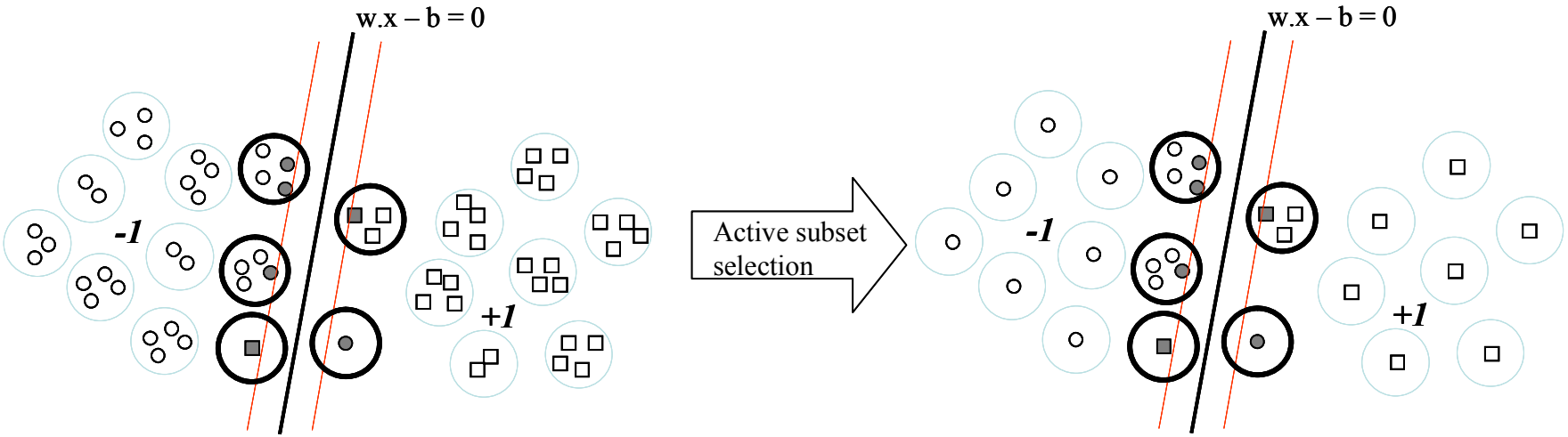
2)

Giải thuật SVM
(Bennett & Campell, 2000)
(Cristianini & Shawe-Taylor, 2000)

- SVM = giải của bài toán quy hoạch toàn phương
- giải quadratic program
  - sử dụng trình tối ưu hóa sẵn có: phương pháp Quasi-Newton, gradient
  - phân rã vấn đề (decomposition hoặc chunking)
  - phân rã vấn đề thành những vấn đề con kích thước 2 như SMO (Platt, 1998)
- diễn giải SVM bằng phương pháp khác
  - lớp các giải thuật của Mangasarian & sinh viên của ông
  - hoặc đưa về giải bài toán quy hoạch tuyến tính hay hệ phương trình tuyến tính

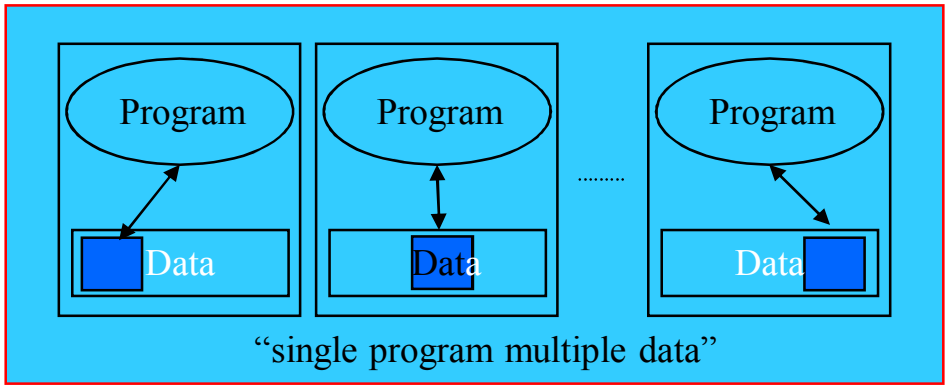
Giải thuật SVM cho khối dữ liệu khổng lồ

- active SVM
  - chọn tập dữ liệu con từ tập dữ liệu ban đầu
  - học trên tập dữ liệu con đó



Giải thuật SVM cho khối dữ liệu khổng lồ

- học song song SVM
  - chia thành những khối dữ liệu nhỏ phân trên các máy tính con
  - học độc lập và song song
  - tập hợp những mô hình con để sinh ra mô hình tổng thể

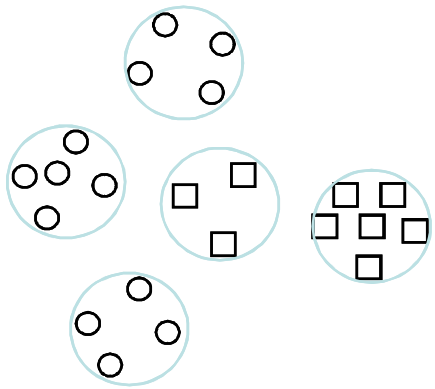


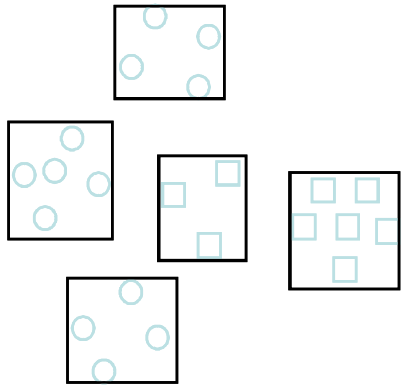
Giải thuật SVM cho khối dữ liệu khổng lồ

- học tiến hóa SVM (incremental learning)
  - chia thành những khối dữ liệu nhỏ
  - load lần lượt từng khối
  - cập nhật mô hình

Giải thuật SVM cho khối dữ liệu khổng lồ

- học trên dữ liệu symbolic
  - chuyển dữ liệu về dạng trình bày ở mức cao hơn
  - có thể là clusters, interval, ...
  - học trên dữ liệu trừu tượng này
  - thay đổi cách tính hàm kernel

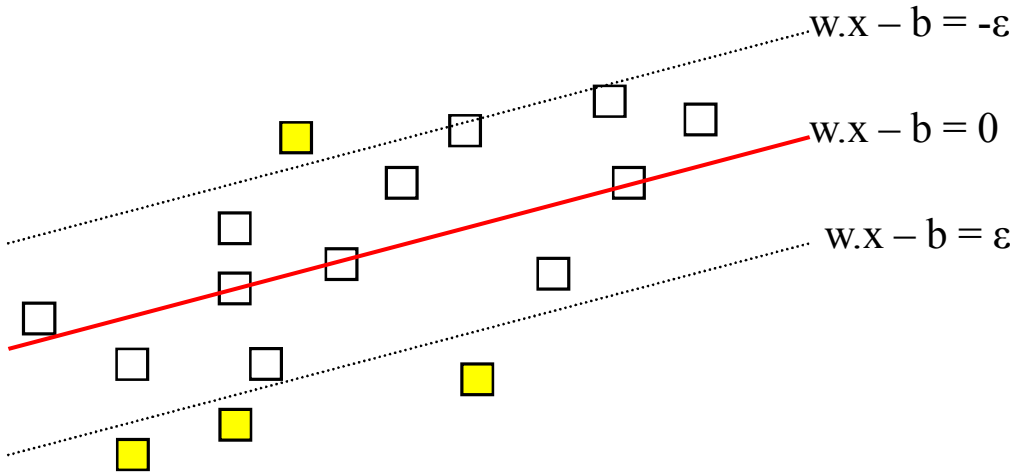




Giải thuật SVM cho khối dữ liệu khổng lồ

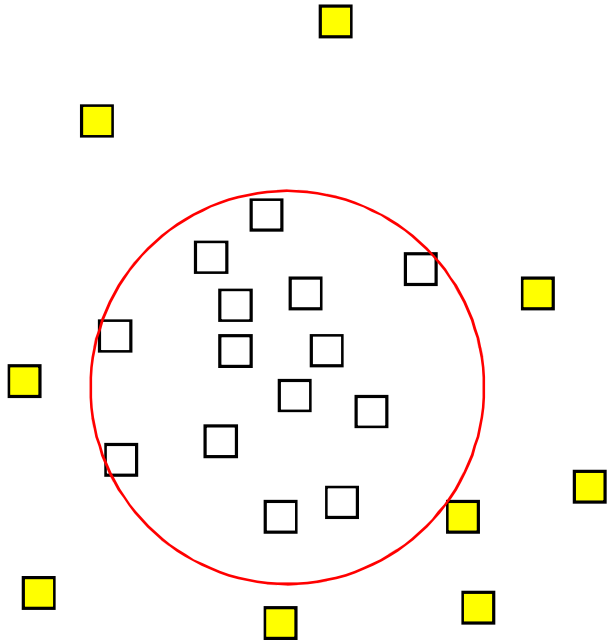
- Boosting SVM
  - lấy mẫu con từ tập dữ liệu ban đầu
  - học trên tập dữ liệu con này
  - phân loại toàn bộ dữ liệu
  - những dữ liệu bị phân loại sai sẽ được cập nhật trọng lượng tăng
  - lấy mẫu dựa trên trọng lượng gán cho dữ liệu
  - tiếp tục học, etc.
  - quá trình lặp *k* lần
  - các mô hình con sẽ bình chọn khi phân loại dữ liệu mới đến

Hồi quy với SVM



- tìm siêu phẳng (Input hoặc Hilbert space)
  - đi qua tất cả các điểm với độ lệch chuẩn là epsilon

Một lớp với SVM



- tìm siêu cầu (Input hoặc Hilbert space)
  - tâm *o* bán kính nhỏ nhất *r*
  - chứa hầu hết các điểm dữ liệu

- Giới thiệu về SVM
- Giải thuật học của SVM
- **Ứng dụng của SVM**
- Kết luận và hướng phát triển

Ứng dụng của SVM

- tham khảo, (Guyon, 1999)
  - http://www.clopinet.com/isabelle/Projects/SVM/applist.html
  - nhận dạng: tiếng nói, ảnh, chữ viết tay (hơn mạng nơron)
  - phân loại văn bản, khai mỏ dữ liệu văn bản
  - phân tích dữ liệu theo thời gian
  - phân tích dữ liệu gien, nhận dạng bệnh, công nghệ bào chế thuốc
  - phân tích dữ liệu marketing
  - etc.

SVM nhận dạng số viết tay

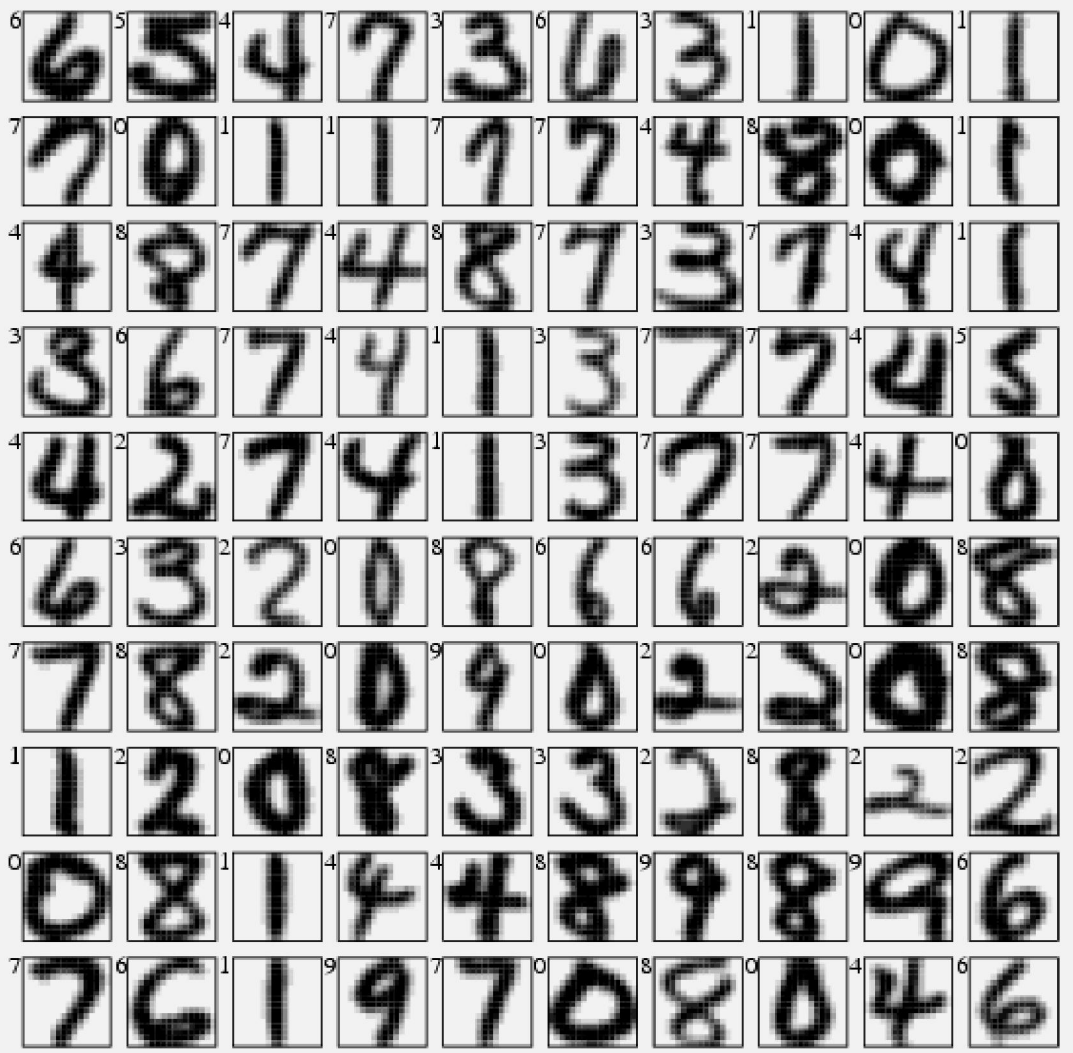


Figure 2. The first 100 USPS training images, with class labels.

SVM nhận dạng số viết tay

The *US Postal Service (USPS)* database (see Figure 2) contains 9298 handwritten digits (7291 for training, 2007 for testing), collected from mail envelopes in Buffalo (LeCun et al., 1989). Each digit is a 16×16 image, represented as a 256-dimensional vector with entries between -1 and 1. Preprocessing consisted of smoothing with a Gaussian kernel of width $\sigma = 0.75$.

It is known that the USPS test set is rather difficult — the human error rate is 2.5% (Bromley and Säckinger, 1991). For a discussion, see

C4.5 decision tree	16%	(Cortes and Vapnik, 1995)
LeNet1	5%	(LeCun et al., 1989)
SVM	4%	(Schölkopf et al., 1995)
SVM + invariances	3%	(Schölkopf, 1997)
Humans; Tangent Distance	2.5%	(Simard et al., 1993)

SVM nhận số viết tay

MNIST Error Rates

handwritten character benchmark (60000 training & 10000 test examples, 28×28)

Classifier	test error
linear classifier	8.4%
3-nearest-neighbour	2.4%
SVM	1.4%
Tangent distance	1.1%
LeNet4	1.1%
Boosted LeNet4	0.7%
Translation invariant SVM	0.56%

Note: the SVM used a polynomial kernel of degree 9, corresponding to a feature

SVM phân loại text (reuters)

Table 1. Break-even performance for five learning algorithms.

	FINDSIM (%)	NAIVE BAYES (%)	BAYESNETS (%)	TREES (%)	LINEARSVM (%)
earn	92.9	95.9	95.8	97.8	98.0
acq	64.7	87.8	88.3	89.7	93.6
money-fx	46.7	56.6	58.8	66.2	74.5
grain	67.5	78.8	81.4	85.0	94.6
crude	70.1	79.5	79.6	85.0	88.9
trade	65.1	63.9	69.0	72.5	75.9
interest	63.4	64.9	71.3	67.1	77.7
ship	49.2	85.4	84.4	74.2	85.6
wheat	68.9	69.7	82.7	92.5	91.8
corn	48.2	65.3	76.4	91.8	90.3
Avg. top 10	64.6	81.5	85.0	88.4	92.0
Avg. all	61.7	75.2	80.0	N/A	87.0

SVM phân loại gien

Table 2. Results for the perceptron using all features

Dataset	Features	FP	FN	SVM FP	SVM FN
Ovarian I	97 802	4.6	4.8	5	3
Ovarian II	97 802	4.4	3.4	0	0
AML/ALL train	7 129	0.6	2.8	0	0
AML treatment	7 129	4.8	3.5	3	6
Colon	2 000	3.8	3.7	3	3

- Giới thiệu về SVM
- Giải thuật học của SVM
- Ứng dụng của SVM
- **Kết luận và hướng phát triển**

Kết luận

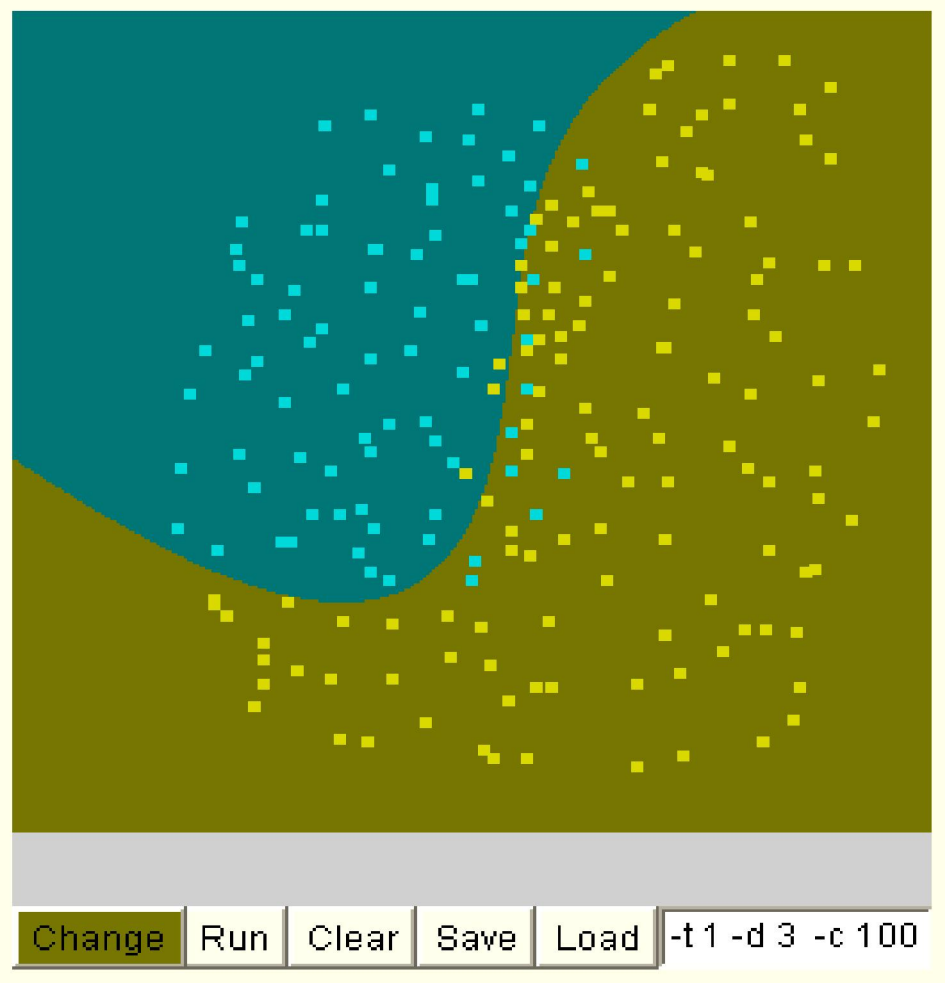
- SVM + kernel methods
  - phương pháp học mới
  - cung cấp nhiều công cụ
  - nền tảng lý thuyết học thống kê
  - tối ưu toàn cục, mô hình chất lượng cao, chịu đựng được nhiễu
  - thành công trong nhiều ứng dụng
- hạn chế
  - khó dịch kết quả
  - độ phức tạp vẫn cao
  - xử lý dữ liệu kiểu số
  - tham số đầu vào

Hướng phát triển

- hướng phát triển
  - multi-class
  - clustering
  - xử lý dữ liệu lớn
  - dữ liệu không phải kiểu số
  - dữ liệu không cân bằng
  - xây dựng hàm nhân
  - dịch kết quả
  - tìm kiếm thông tin (ranking)

DEMO chương trình LibSVM

www.csie.ntu.edu.tw/~cjlin/libsvm



LibSVM

- định dạng dữ liệu

label att-i:val-i …. att-n:val-n

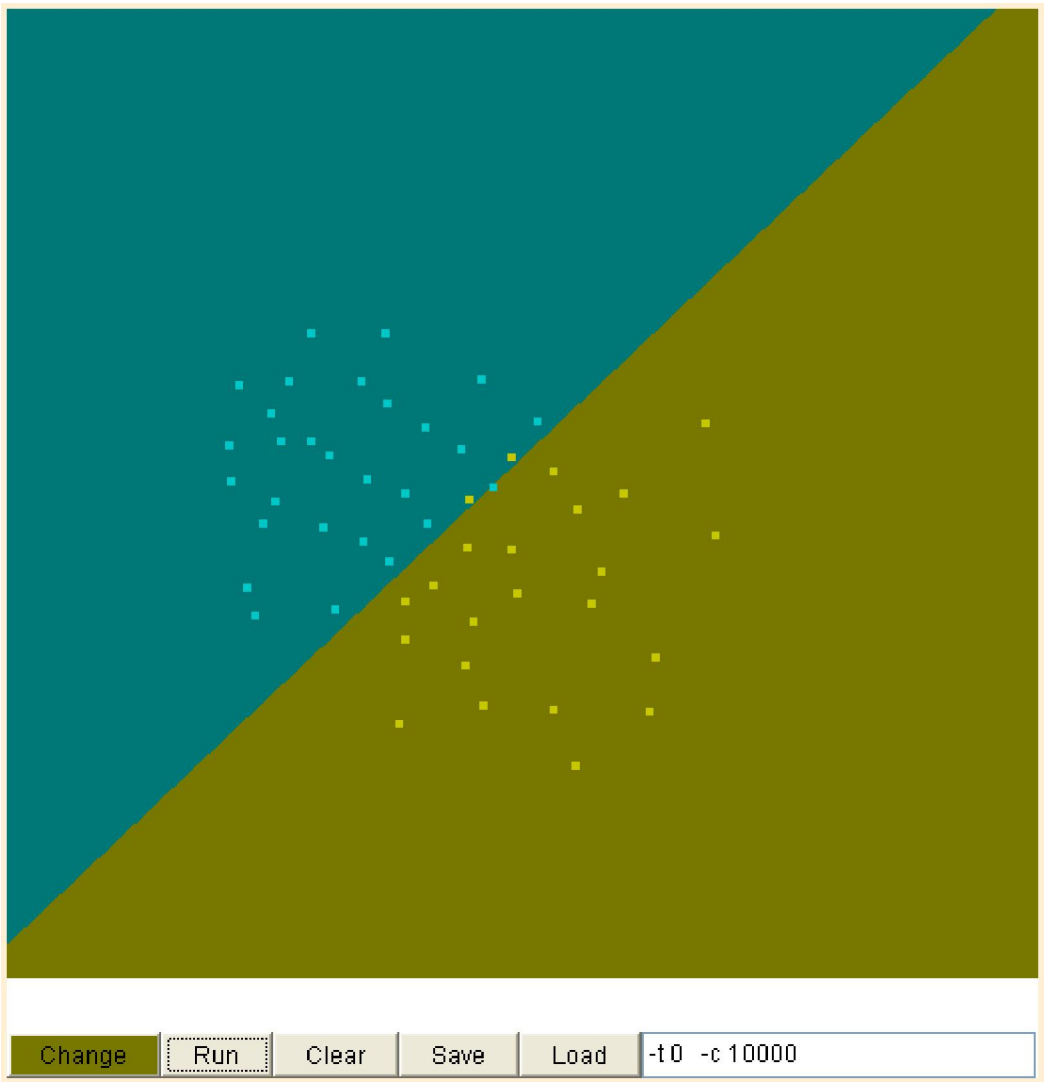
- ví dụ

1 1:-0.555556 2:0.25 3:-0.864407 4:-0.916667
1 1:-0.666667 2:-0.166667 3:-0.864407 4:-0.916667
1 1:-0.777778 3:-0.898305 4:-0.916667
2 1:0.111111 2:-0.583333 3:0.322034 4:0.166667
2 1:-1.32455e-07 2:-0.333333 3:0.254237 4:-0.0833333
3 1:0.222222 2:-0.166667 3:0.525424 4:0.416667
3 1:0.888889 2:0.5 3:0.932203 4:0.75
3 1:0.888889 2:-0.5 3:1 4:0.833333

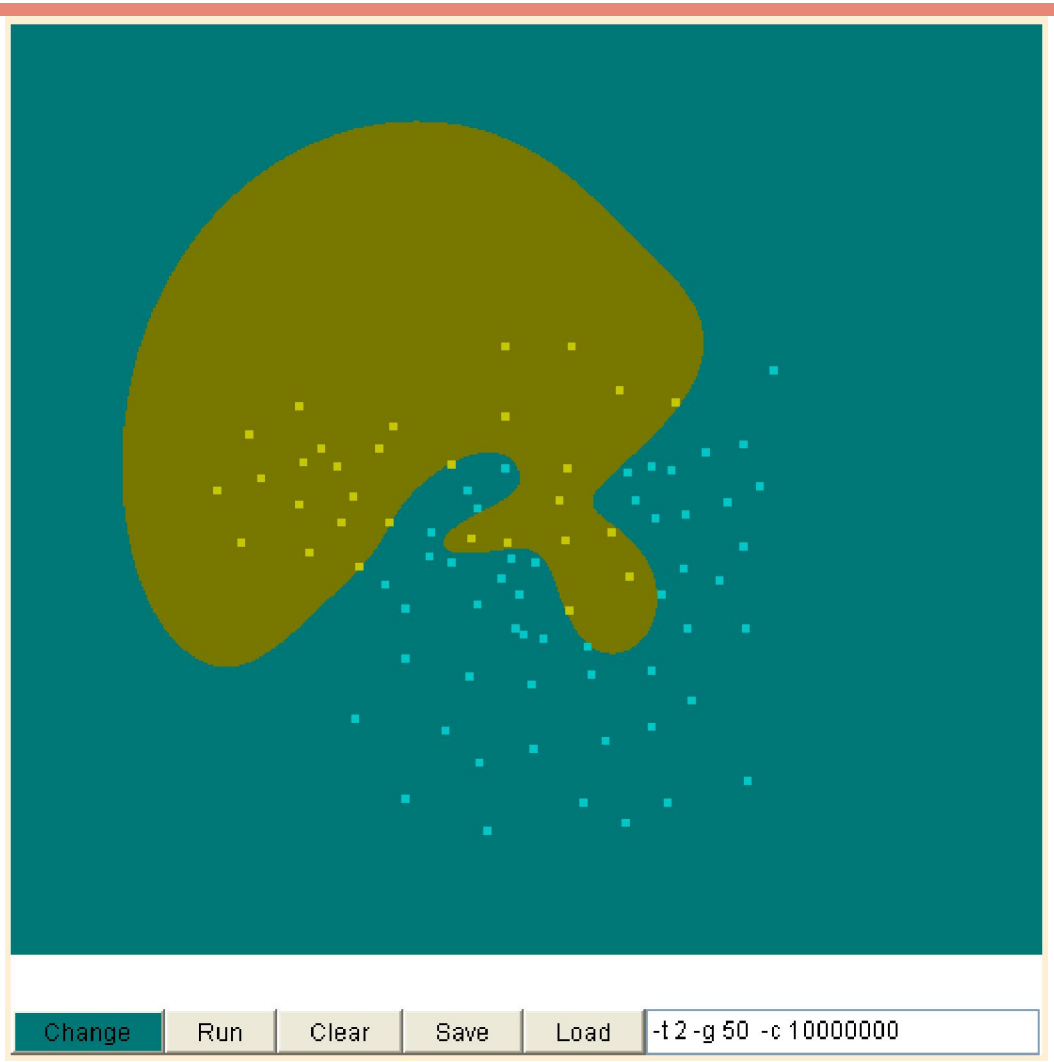
LibSVM

- tùy chọn
  - -s svm_type (default 0) : 0 (SVC), 1 (nu-SVC), 2 (one-class), 3 (epsilon-SVR), 4 (nu-SVR)
  - -t kernel_type (default 2) : 0 (lin), 1 (poly), 2 (RBF), 3 (sigmoid)
  - -d degree (default 3) : polynomial kernel
  - -g gamma (default 1/#attr) : RBF kernel
  - -c cost (default 1) : C-SVC, epsilon-SVR, nu-SVR
  - -p epsilon (default 0.1) : epsilon-SVR
  - -v num_fold : kiểm tra chéo

Test (2d)



Test (2d)



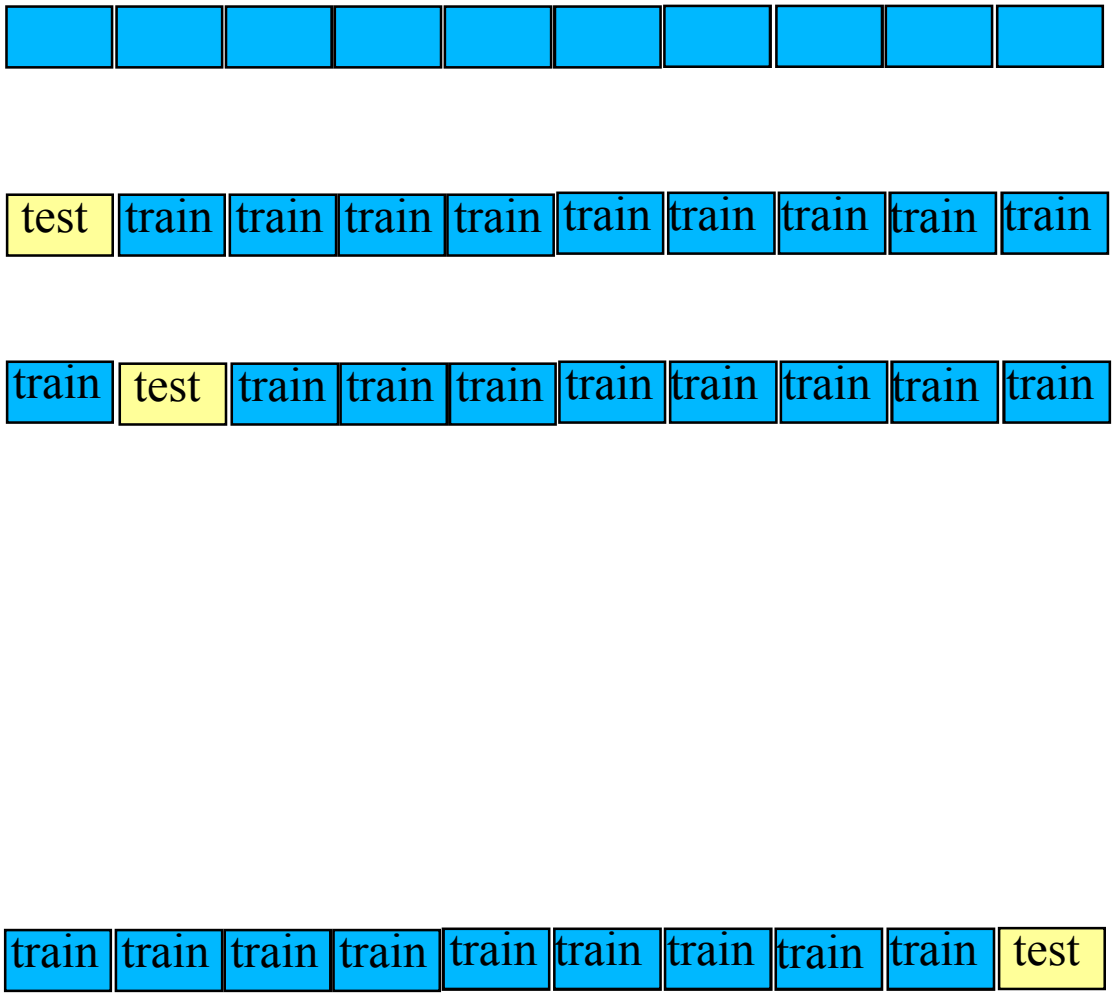
Test (2d)

Tập dữ liệu

- UCI (Asuncion & Newman, 2007)
  - *Spambase, 4601 ind., 57 att., 2 classes (spam, non)
  - *Image Segmentation, 2310 ind., 19 att., 7 classes
  - Landsat Satellite, 6435 ind. (4435 tập học, 2000 tập kiểm tra), 36 att., 6 classes
  - Reuters-21578, 10789 ind. (7770 tập học, 3019 tập kiểm tra), 29406 att., 2 classes (earn, rest)
- Bio-medicales (Jinyan & Huiqing, 2002)
  - ALL-AML Leukemia, 72 ind. (38 tập học, 34 tập kiểm tra), 7129 att., 2 classes (ALL, AML)
  - Lung Cancer, 181 ind. (32 tập học, 149 tập kiểm tra), 12533 attr., 2 classes (cancer, normal)

k-fold

- 10-fold
- it 1 :
- it 2 :
- it 10 :



Demo

- Spambase (spam=1, non-spam=2)
  - nghi thức kiểm tra : 10-fold
  - linear kernel
  - lệnh : svm-train -t 0 -c 10 -v 10 spambase.scale
- kết quả

```
matrix confusion
----------
      |     1      2
    1 |  1618    195
    2 |   134   2654
Cross Validation Accuracy = 92.8494%
```

Demo

- Image Segmentation
  - nghi thức kiểm tra : 10-fold
  - RBF kernel
  - lệnh : svm-train -t 2 -g 0.0002 -c 10000 -v 10 segment.data
- kết quả

```
matrix confusion
---------
      |     1      2      3      4      5      6      7
    1 |   328      0      0      1      1      0      0
    2 |     0    330      0      0      0      0      0
    3 |     0      0    311      5     14      0      0
    4 |     2      0      8    310      9      1      0
    5 |     1      0     10      8    311      0      0
    6 |     0      0      0      0      0    330      0
    7 |     0      0      0      0      0      0    330
Cross Validation Accuracy = 97.4026%
```

Demo

- Landsat Satellite
  - tập học sat.trn, tập kiểm tra sat.tst
  - RBF kernel
  - lệnh : svm-train -t 2 -g 0.001 -c 100000 sat.train sat.rbf
- kết quả

```
matrix confusion
---------
      |     1       2       3       4       5       6       7
    1 |   452       3       3       0       3       0       0
    2 |     0     221       0       0       1       0       2
    3 |     3       4     367      16       2       0       5
    4 |     0       5      30     148       1       0      27
    5 |     0       4       0       1     224       0       8
    6 |     0       0       0       0       0       0       0
    7 |     0       3      10      17      11       0     429
Accuracy = 92.05% (1841/2000) (classification)
```

Demo

- Reuters-21578 (earn=1, rest=2)
  - tập học r.trn, tập kiểm tra r.tst
  - linear kernel
  - lệnh : svm-train -t 0 -c 1000 r.trn r.lin
- kết quả

```
matrix confusion
----------
        |     1       2
      1 |  1079       8
      2 |    34    1898
Accuracy=98.608811
```

Demo

- ALL-AML Leukemia (ALL=1, AML=2)
  - tập học allaml.trn, tập kiểm tra allaml.tst
  - linear kernel
  - lệnh : svm-train -t 0 -c 1000000 allaml.trn allaml.lin
- kết quả

```
matrix confusion
---------
      |     1       2
    1 |    19       1
    2 |     0      14
Accuracy=97.058824
```

Demo

- Lung Cancer (cancer=1, normal=2)
  - tập học lung.trn, tập kiểm tra lung.tst
  - linear kernel
  - lệnh : svm-train -t 0 -c 1000000 lung.trn lung.lin
- kết quả

```
matrix confusion
----------
      |     1       2
    1 |    15       0
    2 |     2     132
Accuracy=98.657718
```

Cám ơn !