Support Vector Machines

Thanh-Nghi Do

Can Tho University

dtnghi@cit.ctu.edu.vn

Can Tho

Dec. 2019

Top 10 Data Mining Algorithms

(Kdnuggets)

Here are the algorithms:

- 1. C4.5
- 2. k-means
- 3. Support vector machines
- 4. Apriori
- 5. EM
- 6. PageRank
- 7. AdaBoost
- 8. kNN
- 9. Naive Bayes
- 10. CART

Support vector machines (Vapnik and Chervonenkis, 1964)

SVM

References

- K. Bennett and C. Campbell. Support Vector Machines: Hype or Hallelujah?. *SIGKDD Explorations* 2(2): 1-13, 2000
- C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* 2(2): 121-167, 1998.
- A. Cornuéjols. Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à Vaste Marge. *Bulletin de l'AFIA*, No. 51, pp. 14-23, 2002.
- N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000

References

- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001
- B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. MIT Press, Cambridge, 2002
- V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995

References

- B. Schölkopf and A. Smola. Kernel-Machines.org.
  - - communauté des SVM et méthodes de noyaux
  - - publications, logiciels, activités, forum, etc.
- I. Guyon. SVM Application List. http://www.clopinet.com/isabelle/Projects/SVM/applist.html
- C. Chang and C. Lin. LIBSVM : a Library for Support Vector Machines. 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm
- T. Joachims. Making Large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1999. http://svmlight.joachims.org/

References

- A. Asuncion and D. Newman. UCI machine learning repository. 2007. http://www.ics.uci.edu/~mlearn/MLRepository.html.

- L. Jinyan and L. Huiqing. Kent ridge bio-medical data set repository. 2002. http://datam.i2r.a-star.edu.sg/datasets/krbd/.

Support Vector Machines

SVM

Content

- Introduction
- SVM learning algorithm
    - classification
    - other tasks
- Applications of SVM
- Conclusions
- Demo

Content

- **<u>Introduction</u>**
- SVM learning algorithm
  - classification
  - other tasks
- Applications of SVM
- Conclusions
- Demo

Introduction

- Support vector machines (SVM)
  - A class of machine learning algorithms
  - Try to find the optimal hyper-plane for separating furthest from 2 classes (can be extended for multi-class)
  - Use the idea of kernel substitution
  - Most accurate models
  - Deal with many tasks: classification, regression and novelty detection
  - Applications: face recognition, handwritten characters recognition, text classification, bioinformatics, etc.

Introduction

- Support vector machines (SVM)
  - Geometrical and mathematical approach
  - SVM + kernel function (lin, poly, rbf, sigmoïd) => model
  - Training task: quadratic programming
  - Convex problem: global minima
  - Sparse model: support vectors

Introduction

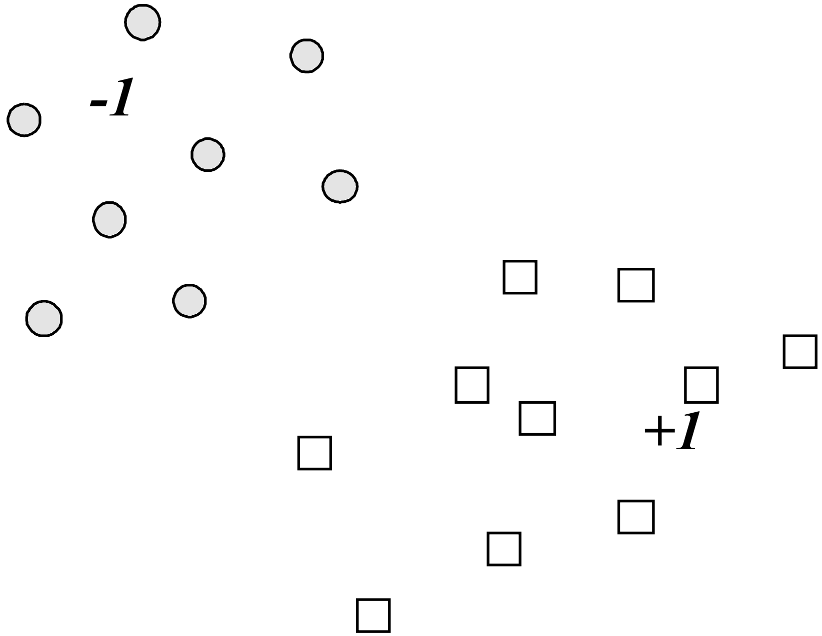
Content

- Introduction
- **SVM learning algorithm**
  - classification
  - other tasks
- Applications of SVM
- Conclusions
- Demo

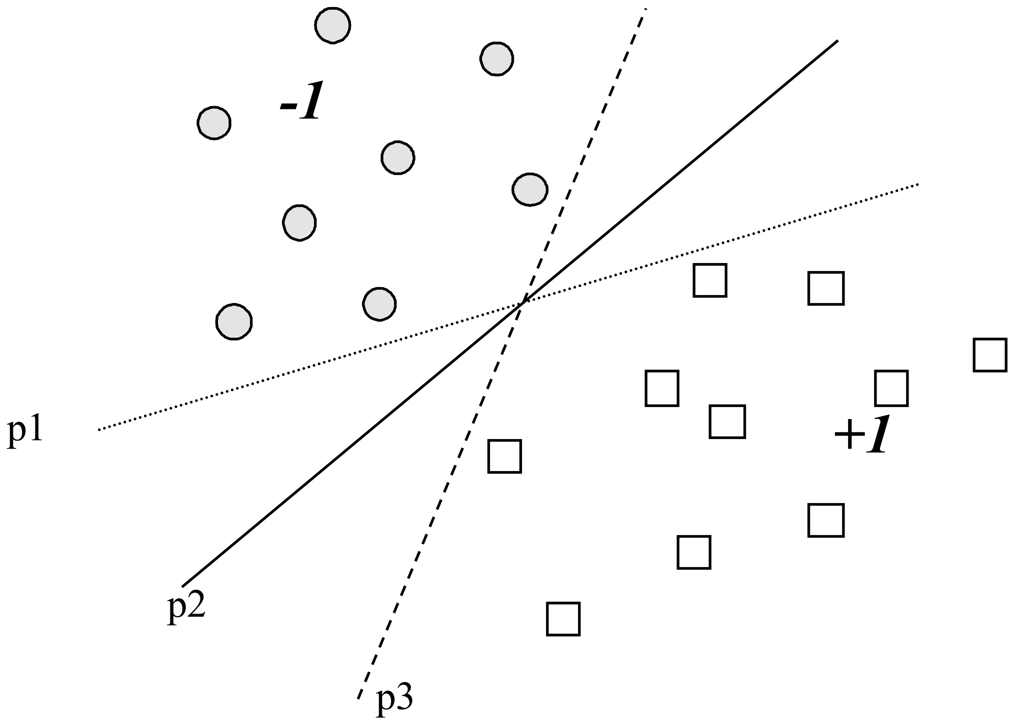
Notations

- n: number of attributes (dimensions)
- m: nomber of examples
- example-i and its class: x_i, y_i (i=1,m)
- A_{mxn}: m examples in n dimensions
- x^T : transpose of x
- w, b : hyper-plane
- $||w||$: 2-norm of the vector w
- dot product of two vectors u, v : u.v, $u^T v$

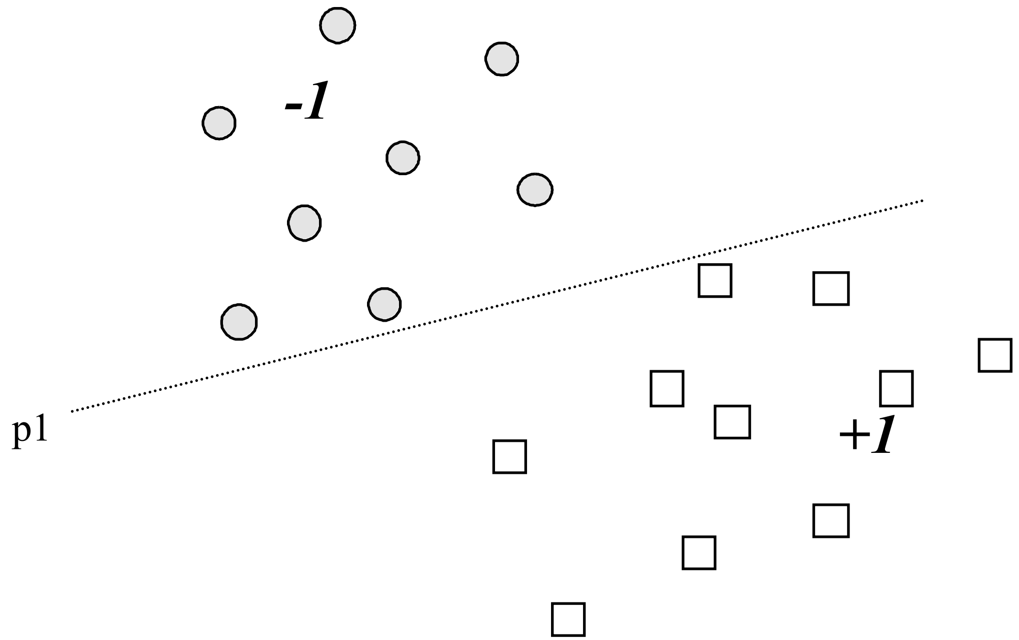
SVM



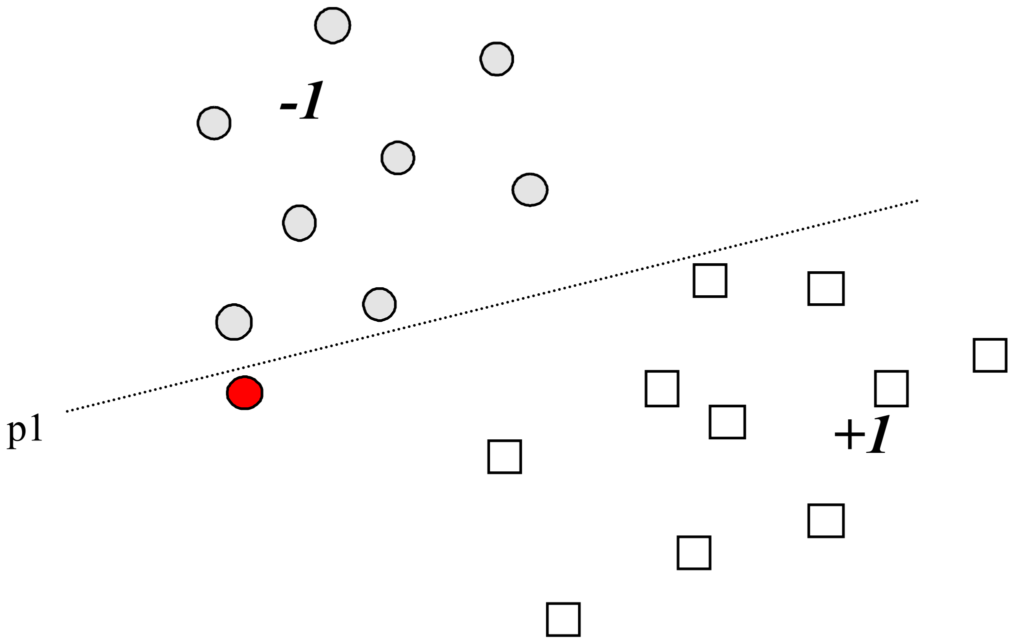
SVM



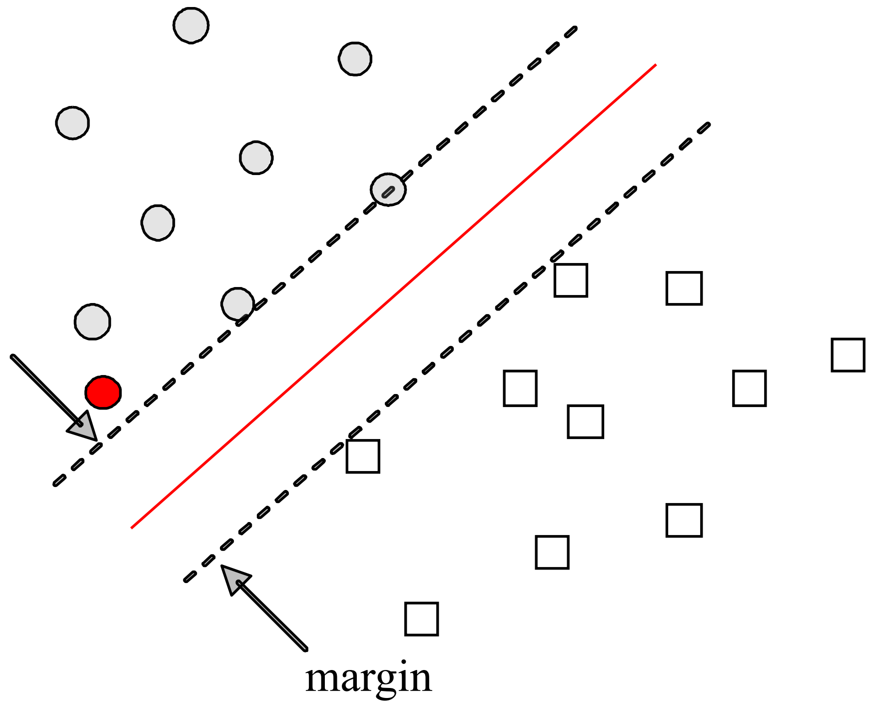
SVM



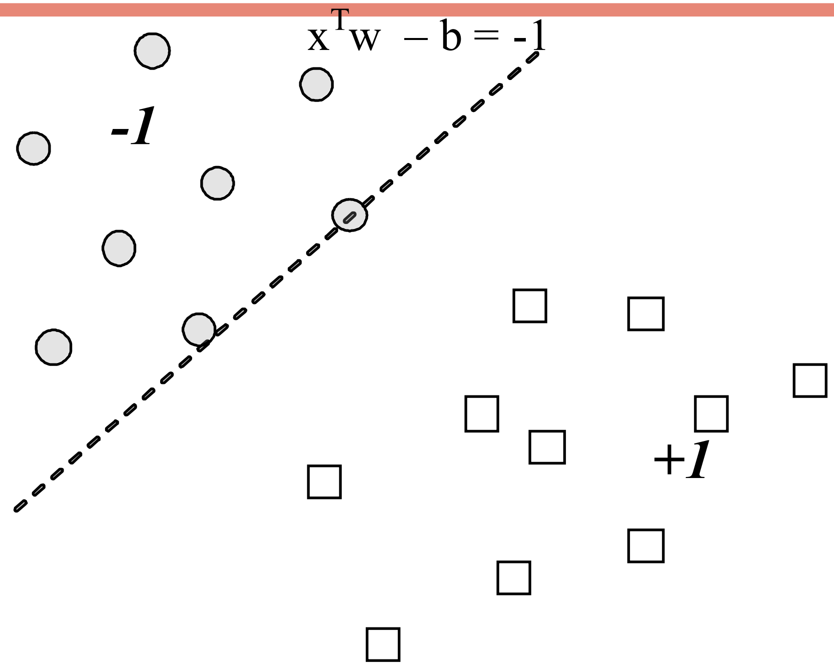
SVM



SVM

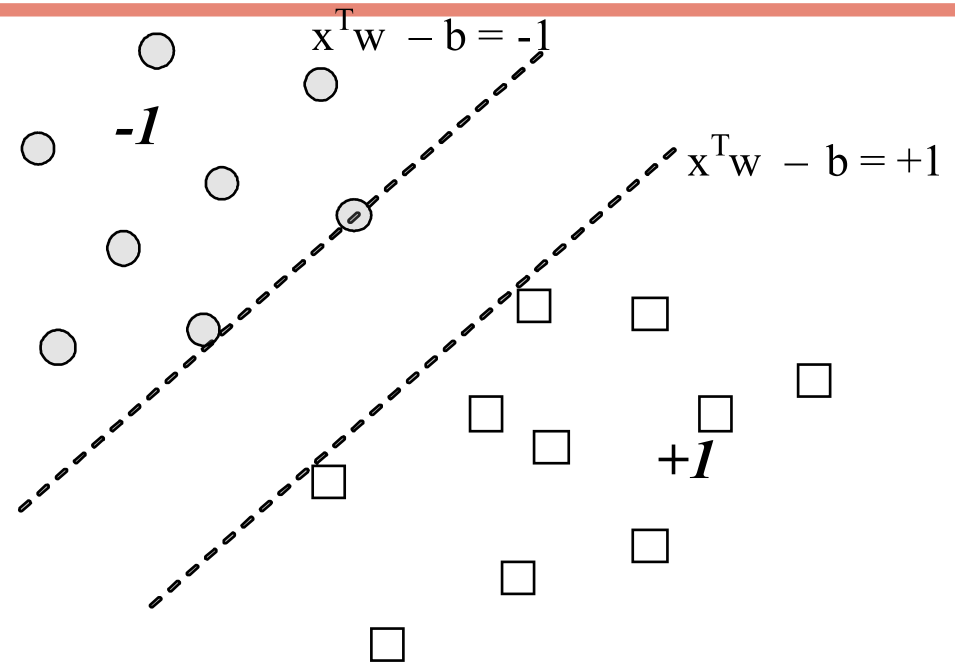


SVM



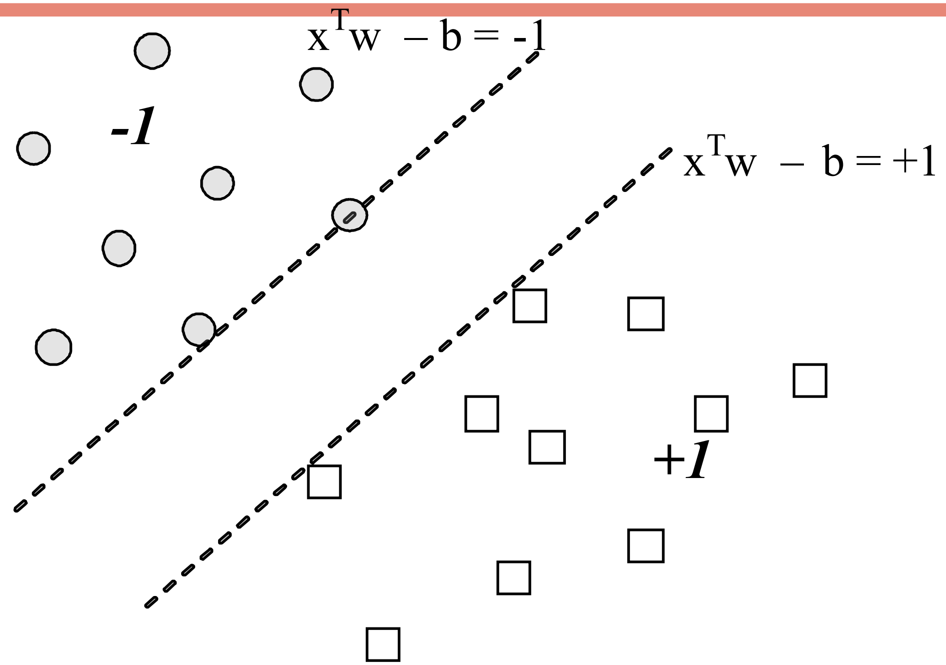
- supporting plane for class -1:
  - $x_i^T w - b \leq -1$ for $y_i = -1$

SVM



- supporting plane for class +1:
  - $x_j^T w - b \geq +1$ *for* $y_j = +1$

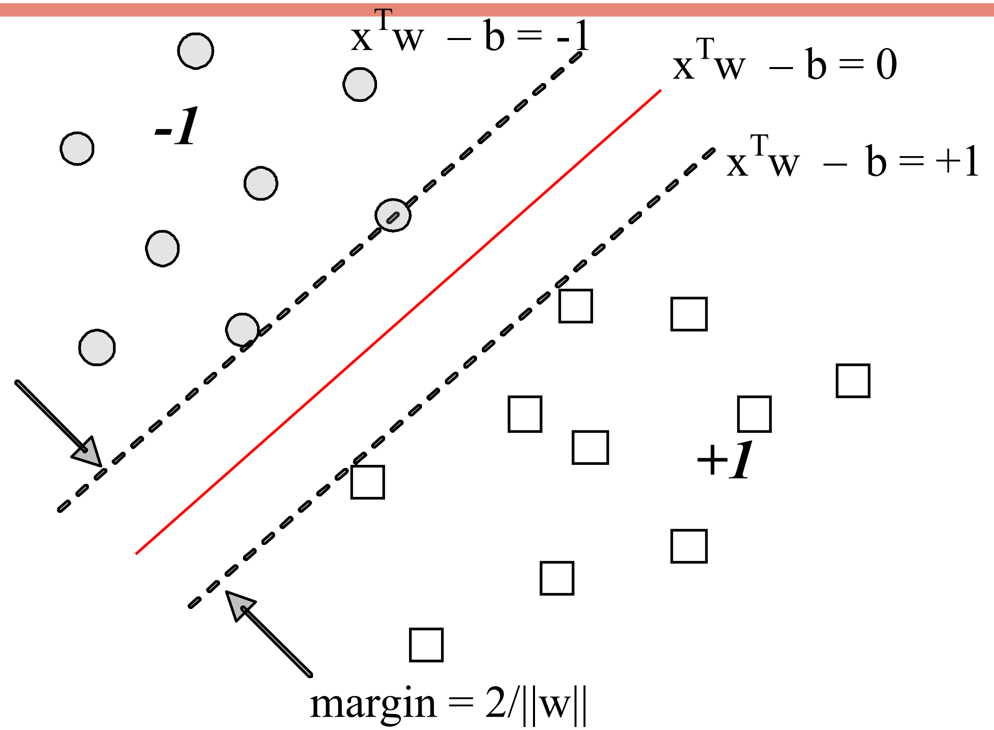
SVM



- supporting plane for class -1:
  - $x_i^T w - b \le -1 \quad for \;\; y_i = -1$
- supporting plane for class +1 :
  - $x_j^T w - b \ge +1 \quad for \;\; y_j = +1$

$$y_i(x_i^T w - b) \ge +1$$

SVM



- Distance between two supporting planes: margin
- SVM aims to maximize the margin (biggest margin => safest model)

SVM

- Maximizing the margin

$$\begin{aligned} & min\ f(w,b) = (1/2)\|w\|^2 \\ & s.t. \quad y_i(w.x_i - b) \geq 1 \quad \forall i = 1,2,...,m \end{aligned} \tag{1}$$

- Solving QP (1): w, b
- Classifying a new example x: predict(x) = sign(w.x - b)
  - predict(x) = 1 if w.x – b > 0
  - predict(x) = -1 if w.x – b < 0

SVM

- Dual formula (Lagrange multipliers α_i) of QP (1) :

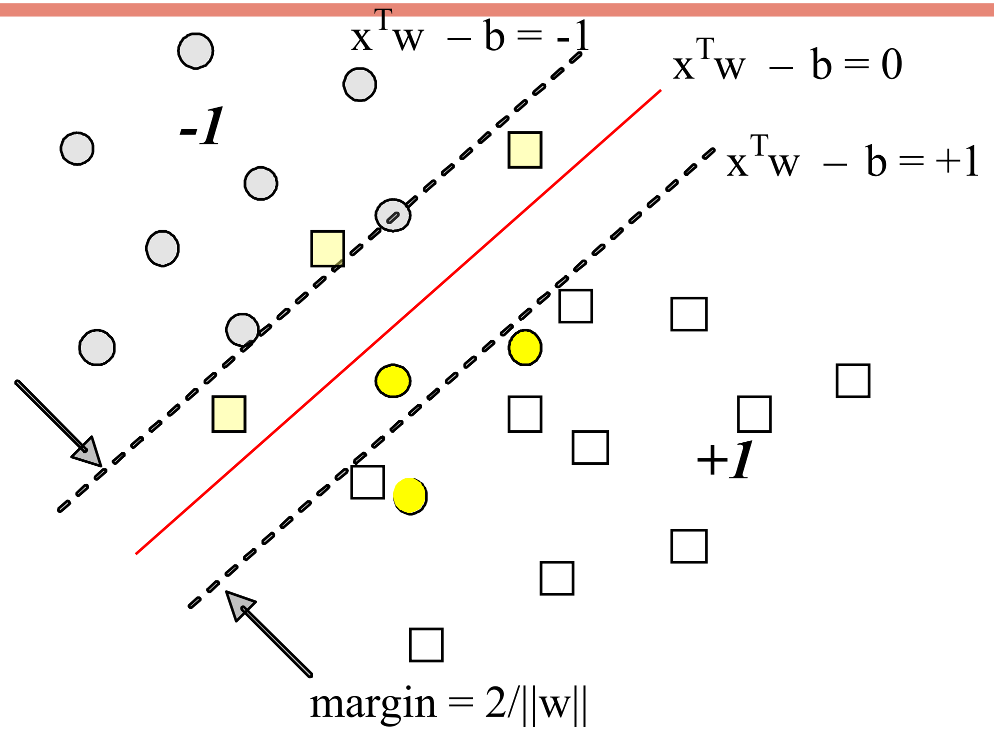
$$\min \Phi(\alpha) = (1/2)\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \langle x_i \cdot x_j \rangle - \sum_{i=1}^{m} \alpha_i$$

$$s.t. \begin{cases} \sum_{i=1}^{m} y_i \alpha_i = 0 \\ 0 \leq \alpha_i \quad \forall i = 1, 2, ..., m \end{cases} \tag{2}$$

with Lagrange multipliers α_i

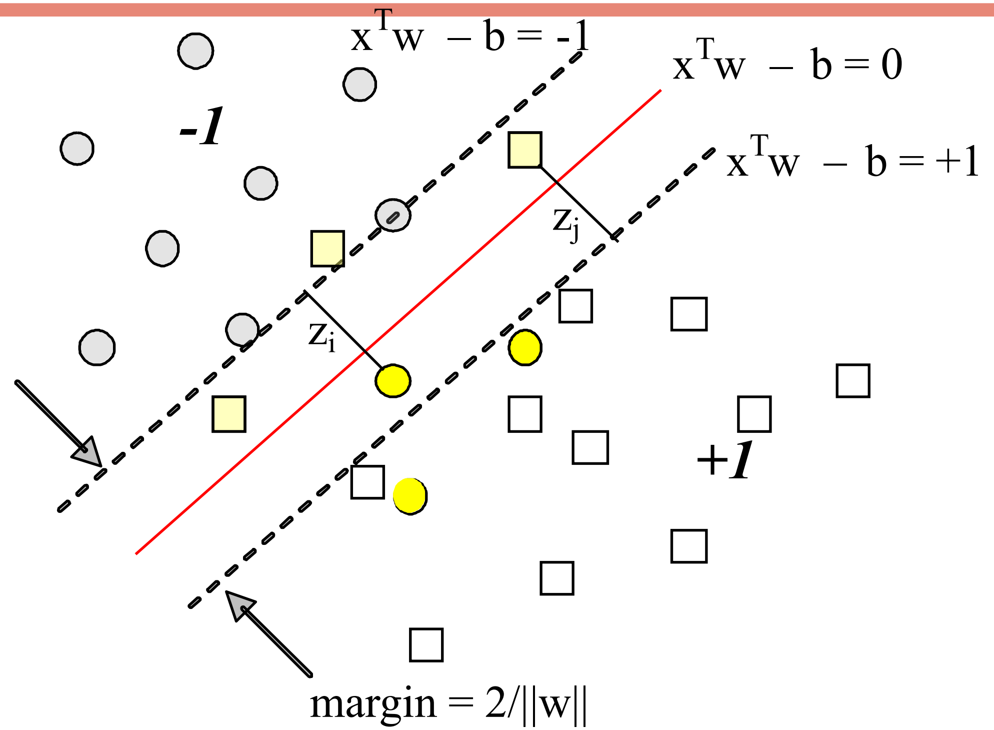
- Solving QP (2) : $\alpha_i > 0$ *(x_i: support vector)*
- $w = \sum_{i=1}^{\#SV} y_i \alpha_i x_i$ *and* $b = ½(w.x_p + w.x_q)$
- with x_p *(+1) and* x_q *(-1)* are support vectors
- Classifying a new example x, $predict(x) = sign(\sum_{i=1}^{\#SV} y_i \alpha_i \langle x \cdot x_i \rangle - b)$

SVM



- SVM aims to maximize the margin and minimize errors

SVM



- SVM aims to maximize the margin and minimize errors

SVM

- Maximizing the margin and minimizing errors

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

$$s.t. \begin{cases} y_i(w.x_i - b) + z_i \geq 1 \\ z_i \geq 0 \quad \forall i = 1,2,...,m \end{cases} \qquad (3)$$

- Solving QP (3) : w, b

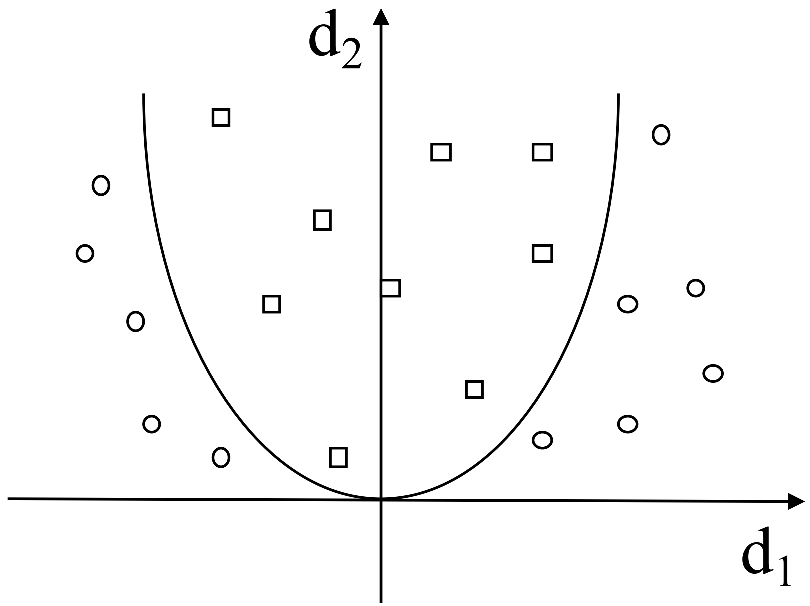
SVM

- Dual formula (Lagrange multipliers α_i) of QP (3) :

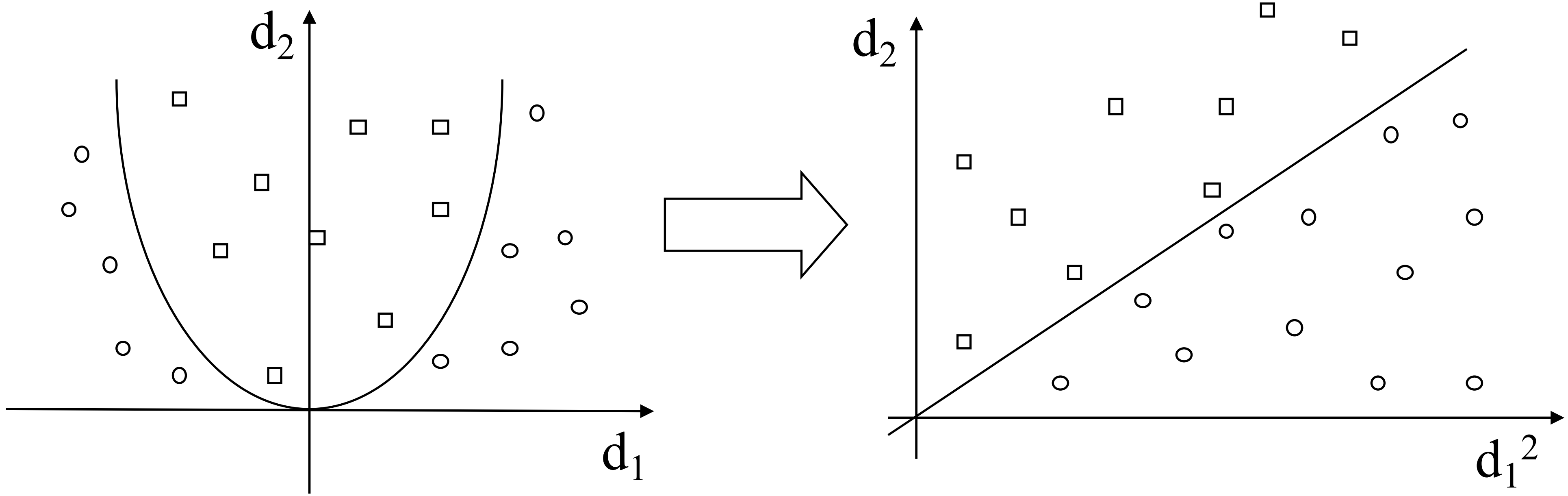
$$\min \Phi(\alpha) = (1/2)\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \langle x_i \cdot x_j \rangle - \sum_{i=1}^{m} \alpha_i$$
$$s.t. \begin{cases} \sum_{i=1}^{m} y_i \alpha_i = 0 \\ 0 \le \alpha_i \le C \quad \forall i = 1, 2, ..., m \end{cases} \qquad (4)$$

- Solving QP (4) : *$\alpha_i > 0$ (x_i: support vector)*
- $w = \sum_{i=1}^{\#SV} y_i \alpha_i x_i$ *and* $b = ½(w.x_p + w.x_q)$
- with x_p *(+1) and* x_q *(-1)* are support vectors
- Classifying a new example x, $predict(x) = sign(\sum_{i=1}^{\#SV} y_i \alpha_i \langle x \cdot x_i \rangle - b)$

Non-linear SVM



Non-linear SVM



- Original input space 2-d (d_1, d_2), mapping operation $\Phi(d_1, d_2)$ =>
- feature space in 5-d, ($d_1, d_2, d_1.d_2, d_1^2, d_2^2$)
- Linear SVM in feature space => non-linear SVM in original input space

Non-linear SVM

- QP (4) => QP (5)

$$
\begin{aligned}
& min\ \Phi(\alpha) = (1/2)\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \phi(x_i)\cdot\phi(x_j) - \sum_{i=1}^{m}\alpha_i \\
& s.t. \begin{cases} \sum_{i=1}^{m} y_i\alpha_i = 0 \\ 0 \le \alpha_i \le C \quad \forall i = 1, 2, ..., m \end{cases}
\end{aligned} \tag{5}
$$

with Lagrange multipliers α_i

- Dimensionality of the feature space explodes exponentially => overfitting
- How to do mapping operation Φ

Non-linear SVM

- Mercer theorem: for certain mappings *Φ* and any two examples *u, v*, the dot product *Φ(u).Φ(v)* is evaluated using the kernel function without ever explicitly knowing the mapping, *Φ(u).Φ(v) = K(u, v)*

- QP (5) => QP (6)

$$min\ \Phi(\alpha) = (1/2)\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K\langle x_i, x_j\rangle - \sum_{i=1}^{m}\alpha_i$$

$$s.t. \begin{cases} \sum_{i=1}^{m} y_i \alpha_i = 0 \\ 0 \le \alpha_i \le C \quad \forall i = 1, 2, ..., m \end{cases} \qquad (6)$$

with Lagrange multipliers α_i, kernel function $K\langle x_i, x_j\rangle$

- Classifying a new example x, $predict(x) = sign(\sum_{i=1}^{\#SV} y_i \alpha_i K\langle x, x_i\rangle - b)$

Non-linear SVM

- Most popular kernel functions (Hilbert Schmidt kernels)

  linear: $x_i . x_j$

  degree d polynomial: $(x_i . x_j + 1)^d$

  RBF (Radial Basis Function): $e^{-\gamma \|x_i - x_j\|^2}$

- Polynomial kernel function with degree 5, mapping examples in *250* dimensions in input space to ~ *10^{10}* dimensions in feature space

Training SVM model = Solving QP

- Existing general-purpose QP algorithms: Quasi-Newton methods, primal-dual interior point methods, MINOS (Murtagh et Saunders, 1992) or LOQO (Vanderbei, 2000), for handling problems of small size (thousands)

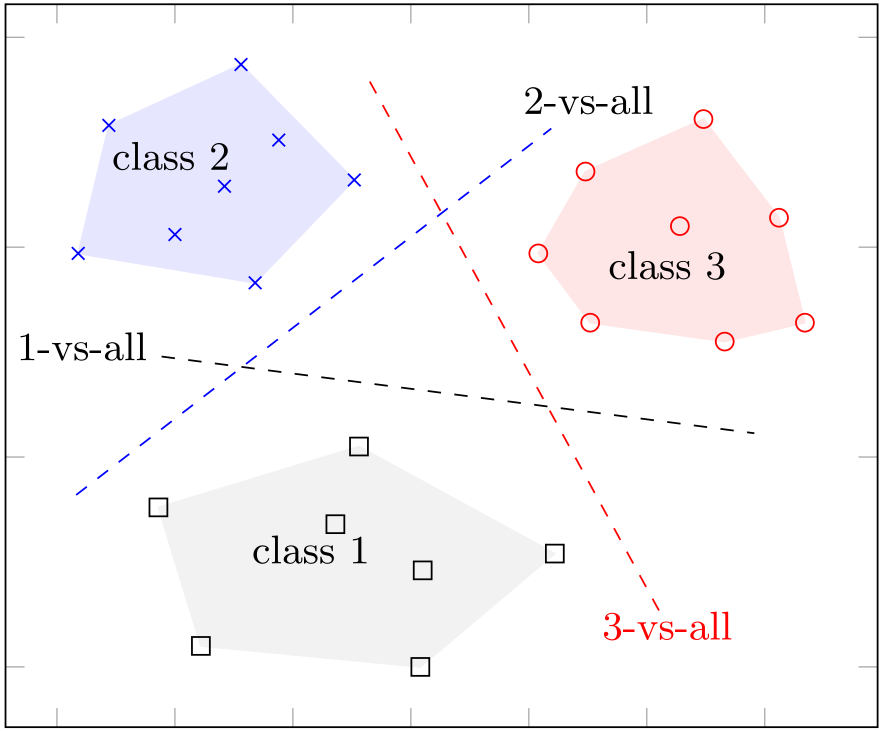
- KernelAdatron (Friess et al., 1998) is to evaluate and to discard kernel components for sequentially updating the Lagrange multipliers with a gradient method (easy to implement, long training time, thousands)

Training SVM model = Solving QP

- Chunking: optimize the objective function using an initial subset of data. The support vectors (corresponding to $\alpha_i > 0$) are kept and other examples (with $\alpha_j = 0$) are discarded. A new working set for the next iteration includes these support vectors and additional examples which maximally violate the constraints. This process is iterated until all Karush-Kuhn-Tucker conditions being satisfied => intractable for large number of support vectors
- Decomposition: use a fixed size subset of data with the support vectors for the remainder kept fixed
- SVMLight (Joachims, 1998), LibSVM (Chang and Lin, 2011), SMO (Platt, 1998): chunking/decomposition
- Reformulate SVMs, including LS-SVM (Suykens, 1999), Lagrangian SVM (Mangasarian, 2001), etc.

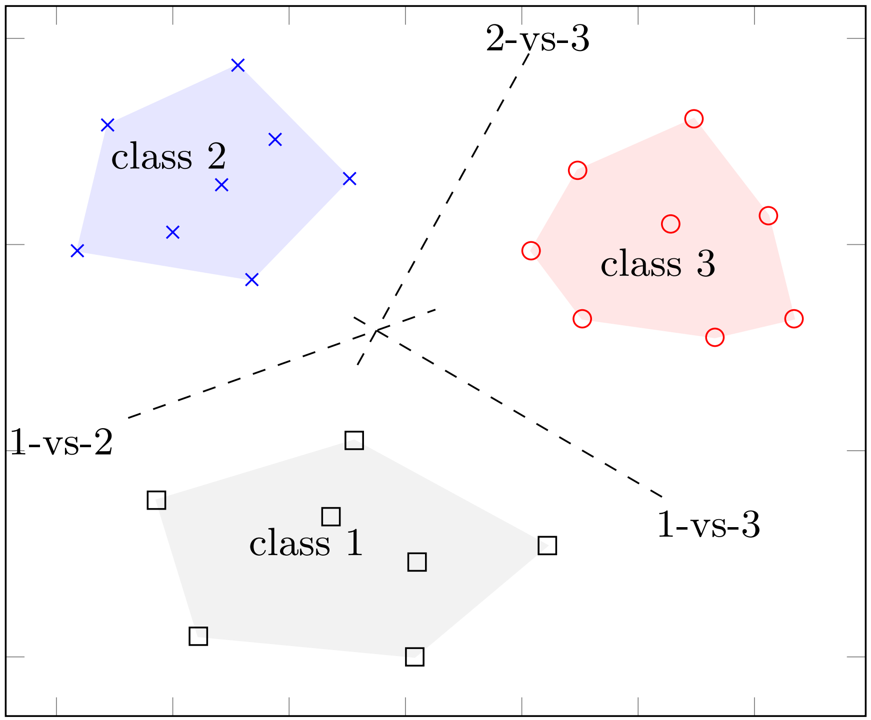
SVM for multi-class (*k* > *2*)

- One-versus-All
- *k* classes => training *k* binary SVM models where the i-th binary one separates the i-th class from the rest
- The class is then predicted with the largest distance vote

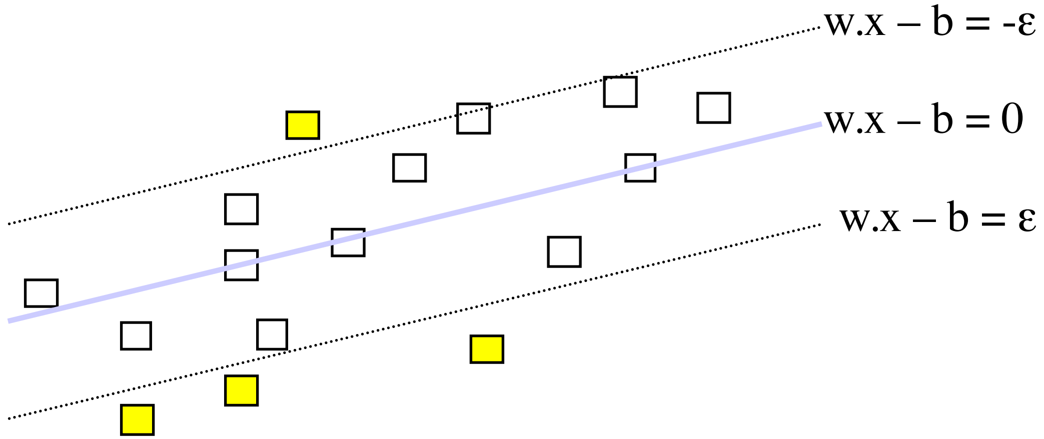


SVM for multi-class (*k > 2*)

- One-versus-One
- *k* classes => training *k(k-1)/2* binary SVM models for all the binary pairwise combinations of the *k* classes
- The class is then predicted with the largest distance vote

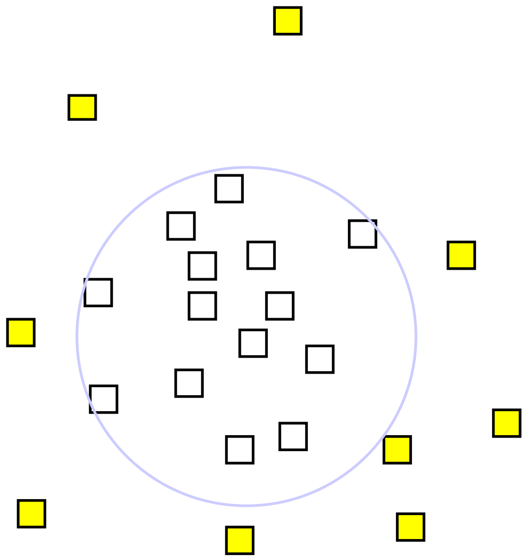


Regression



- Try to find a best hyper-plane that has at most ε deviation from the target value y_i
- Training the regression model: solving QP
- Non-linear regression: kernel substitution

Novelty detection (one-class SVM)



- Try to find a hypersphere with a minimal radius *R* and center *o* which contains most of examples => novel test examples lie outside the boundary of this hypersphere.
- Training the one-class model: solving QP
- Non-linear model: kernel substitution

Content

- Introduction
- SVM learning algorithm
  - classification
  - other tasks
- **<u>Applications of SVM</u>**
- Conclusions
- Demo

Applications of SVM

- Website (Guyon, 1999)
- http://www.clopinet.com/isabelle/Projects/SVM/applist.html
- Pattern recognition: audio, image, handwritten characters,
- Text classification,
- Time series mining,
- Gene expression classification,
- Data analysis,
- etc.

Handwritten digit recognition

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

Dataset US Postal Service (LeCun et al., 1989)

- 9298 images (16x16), handwritten digits scanned from envelopes by the U.S. Postal Service
- Performing the pre-processing stage
- 7291 for trainset, 2007 for testset
- Difficult (Bromley & Sackinger, 1991): ***2,5 % error rate***

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)

B. Schölkopf, DAGM, 14/9/1999

Dataset MNIST (LeCun et al., 1998)

- Subset of NIST (the National Institute of Standards and Technology), 70000 images (28x28), handwritten digits
- 60000 for trainset, 10000 for testset

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

Text classification

- Classifying spams,
- Classifying news
- Dataset reuters-21578 (Lewis, 1987)
- Text representation: Bag-of-words
- Removing stop words: the, a, an, of, and, etc.
- Stemming words: drug, drugs, drugged, etc.
- Text: vector of word frequencies => #dim = #vocab
- Classifying a news into one of classes (earn, acq, money-fx, grain, crude, trade, interest, ship, wheat, corn, etc.)

Text classification: reuters-21578 (Dumais, 1998)

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

Gene expression classification

- DNA microarrays are microscope slides that are printed with thousands of tiny spots in defined positions, with each spot containing a known DNA sequence or gene. Often, these slides are referred to as gene chips or DNA chips. The DNA molecules attached to each slide act as probes to detect gene expression, which is also known as the transcriptome or the set of messenger RNA (mRNA) transcripts expressed by a group of genes.
- mRNA molecules are typically collected from both an experimental sample and a reference sample. For example, the reference sample could be collected from a healthy individual, and the experimental sample could be collected from an individual with a disease like cancer.
- Classifying gene expression: patients/normal

Bio-medical datasets (Jinyan & Huiqing, 2002)

ID	Datasets	#Datapoints	#Dimensions	Classes	Protocols
1	Colon Tumor	62	2000	tumor, normal	loo
2	ALL-AML-Leukemia	72	7129	ALL, AML	trn-tst
3	*MLL-Leukemia	72	12582	MLL, rest	trn-tst
4	Breast Cancer	97	24481	relapse, non-relapse	trn-tst
5	Duke Breast Cancer	42	7129	cancer, normal	loo
6	Prostate Cancer	136	12600	cancer, normal	trn-tst
7	Lung Cancer	181	12533	cancer, normal	trn-tst
8	Central Nervous System	60	7129	positive, negative	loo
9	Translation Initiation Site	13375	927	positive, negative	10-fold
10	Ovarian Cancer	253	15154	cancer, normal	loo
11	Diffuse Large B-Cell Lymphoma	47	4026	germinal, activated	loo
12	*Subtypes of Acute Lymphoblastic (Hyperdip)	327	12558	Hyperdip, rest	trn-tst
13	*Subtypes of Acute Lymphoblastic (TEL-AML1)	327	12558	TEL-AML1, rest	trn-tst
14	*Subtypes of Acute Lymphoblastic (T-ALL)	327	12558	TEL-ALL, rest	trn-tst
15	*Subtypes of Acute Lymphoblastic (Others)	327	12558	Others, diagnostic groups	trn-tst

Classification of bio-medical datasets (Do et al., 2009)

Dataset	Precision			Recall			F1-measure			Accuracy		
ID	LibSVM	RF-C4.5	RF-ODT	LibSVM	RF-C4.5	RF-ODT	LibSVM	RF-C4.5	RF-ODT	LibSVM	RF-C4.5	RF-ODT
1	68.18	76.19	**82.61**	75.00	72.73	**86.36**	71.43	74.42	**84.44**	80.65	82.26	**88.71**
2	**100**	95.24	95.24	95.00	**100**	**100**	97.44	**97.56**	**97.56**	**97.06**	**97.06**	**97.06**
3	75.00	**100**	**100**	**100**	**100**	**100**	85.71	**100**	**100**	93.33	**100**	**100**
4	69.23	83.33	**84.62**	75.00	83.33	**91.67**	72.00	83.33	**88.00**	63.16	78.94	**84.21**
5	85.00	**94.12**	90.00	**94.44**	80.00	90.00	89.47	86.49	**90.00**	**90.48**	88.10	**90.48**
6	73.53	75.76	**100**	**100**	**100**	96.00	84.75	86.21	**97.96**	73.53	76.47	**97.06**
7	88.26	**93.75**	**93.75**	**100**	**100**	**100**	93.75	**96.77**	**96.77**	98.66	**99.33**	**99.33**
8	47.62	45.46	**61.91**	55.56	23.81	**61.91**	51.28	31.25	**61.91**	68.33	63.33	**73.33**
9	83.13	**92.58**	90.78	**84.42**	73.83	79.75	83.77	82.15	**84.91**	92.15	92.30	**93.20**
10	**100**	98.78	**100**	**100**	**100**	**100**	**100**	99.39	**100**	**100**	99.21	**100**
11	91.30	**95.65**	92.00	87.50	91.67	**95.83**	89.36	93.62	**93.88**	89.36	**93.62**	**93.62**
12	95.46	95.24	**100**	**95.46**	90.91	**95.46**	95.46	93.02	**97.67**	98.21	97.32	**99.11**
13	**100**	**100**	**100**	**100**	96.30	96.30	**100**	98.11	98.11	**100**	99.11	99.11
14	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
15	92.59	**100**	**100**	39.68	29.63	**55.56**	55.56	45.71	**71.43**	64.29	83.93	**89.29**

Content

- Introduction
- SVM learning algorithm
  - classification
  - other tasks
- Applications of SVM
- **Conclusions**
- Demo

Conclusions

- Support vector machines
  - Geometrical, mathematical models
  - SVM + kernel function => model
  - Training task: quadratic programming
  - Many optimization approaches
  - High accurate models for text, images, handwritten characters, audio, etc.
  - Handling many tasks: classification, regression and novelty detection

Conclusions

- Disadvantages
  - Blackbox models
  - Training task: quadratic program => expensive
  - Handle large datasets?
  - Unsupervised learning?
  - Data types: binary or nominal type?
  - Tuning hyper-parameters?

Future works

- To do
  - Interpret the resulting SVM model: visualization, interactives approaches, etc.
  - Speed-up training task: parallel algorithms
  - Dealing with very large datasets
  - Incremental, parallel, distributed algorithms
  - Active learning algorithms
  - Handling symbolic data
  - Fusing other approaches
  - Dealing with multi-class, imbalanced datasets, non-numeric data, clustering
  - Creating new kernel functions

Content

- Introduction
- SVM learning algorithm
  - classification
  - other tasks
- Applications of SVM
- Conclusions
- **Demo**

LibSVM

- Data format

class-no attribute-i:value-i …. attribute-n:value-n

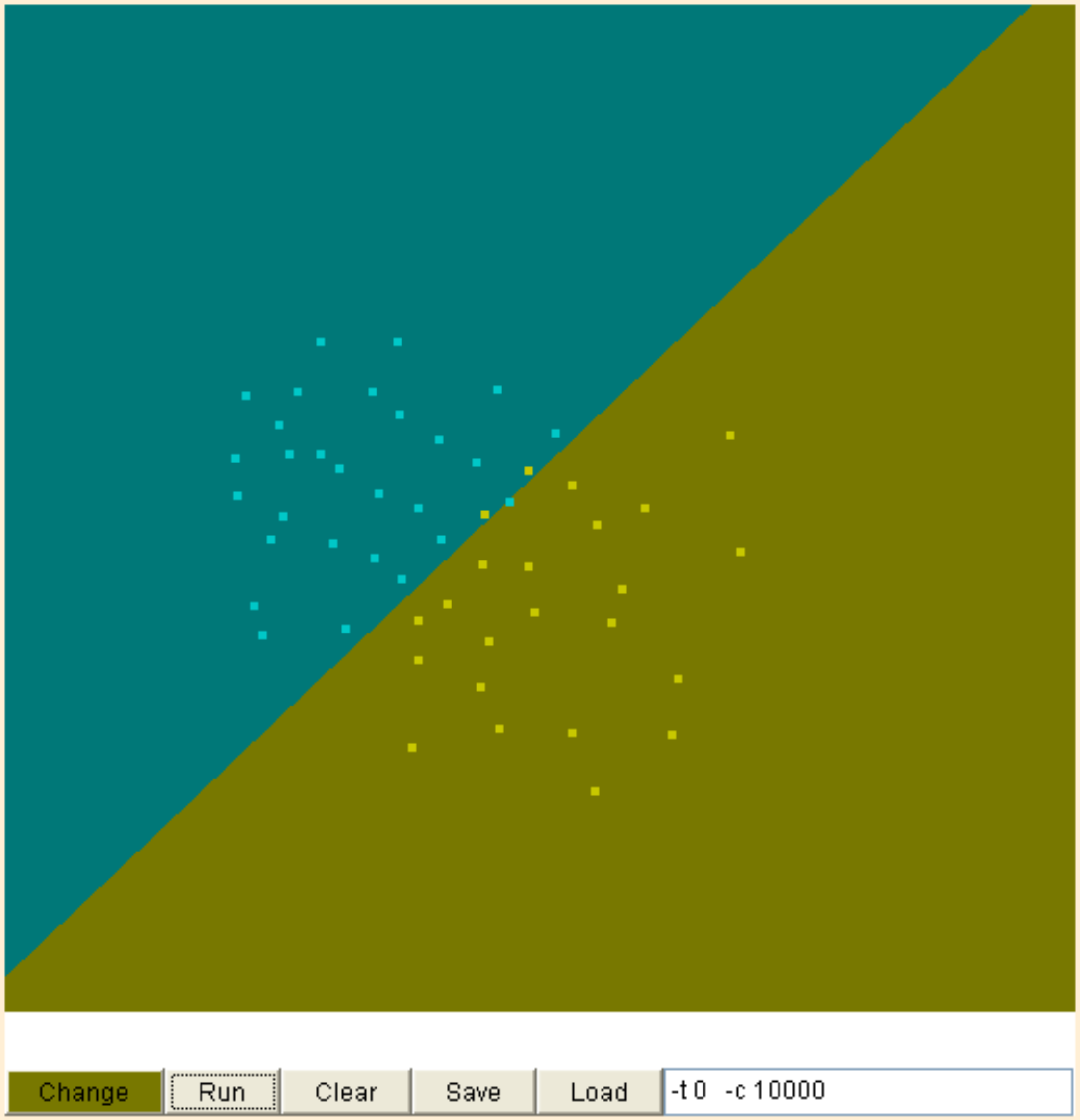
- Example

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

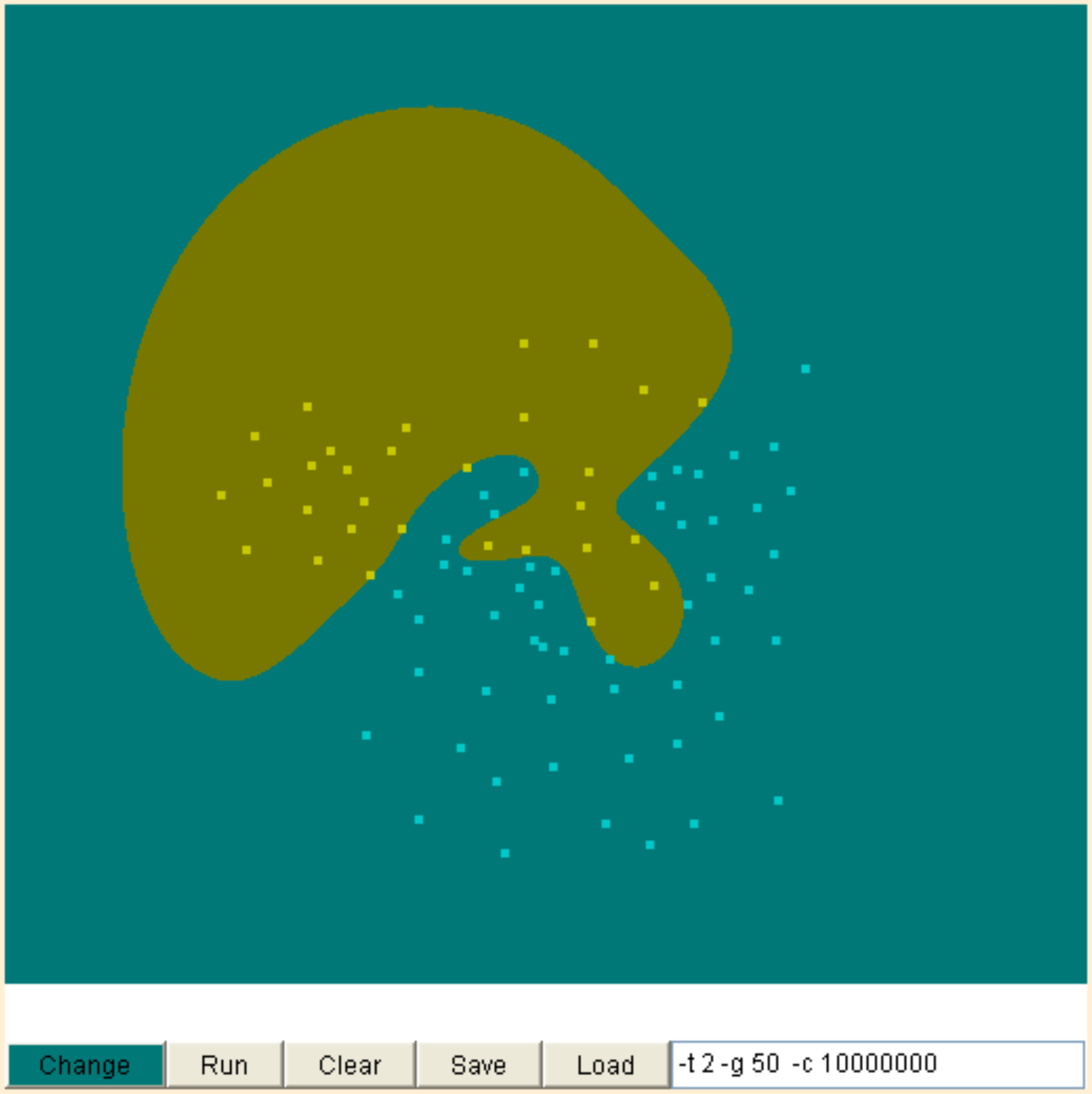
LibSVM

- Options
  - -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 function
  - -g gamma (default 1/#attr) : RBF kernel function
  - -c cost (default 1) : for C-SVC, epsilon-SVR, nu-SVR
  - -p epsilon (default 0.1) : for epsilon-SVR
  - -v num_fold : cross-validation

Dataset (in 2D)



Dataset (in 2D)



Dataset (in 2D)

Datasets

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

k-fold

- 10-fold

- it 1 : test train train train train train train train train train

- it 2 : train test train train train train train train train train

- it 10 : train train train train train train train train train test

Demo

- Spambase (spam=1, non-spam=2)
  - protocol: 10-fold
  - Linear kernel
  - parameters: svm-train -t 0 -c 10 -v 10 spambase.scale
- Result

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

Demo

- Image Segmentation
  - protocol: 10-fold
  - RBF kernel
  - parameters: svm-train -t 2 -g 0.0002 -c 10000 -v 10 segment.data
- Result

```
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
  - sat.trn for trainset, sat.tst for testset
  - RBF kernel
  - parameters: svm-train -t 2 -g 0.001 -c 100000 sat.train sat.rbf
- Result

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)
  - r.trn for trainset, r.tst for testset
  - linear kernel
  - parameters: svm-train -t 0 -c 1000 r.trn r.lin
- Result

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

Demo

- ALL-AML Leukemia (ALL=1, AML=2)
  - allaml.trn for trainset, allaml.tst for testset
  - linear kernel
  - parameters: svm-train -t 0 -c 1000000 allaml.trn allaml.lin
- Result

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

Demo

- Lung Cancer (cancer=1, normal=2)
  - lung.trn for trainset, lung.tst for testset
  - linear kernel
  - parameters: svm-train -t 0 -c 1000000 lung.trn lung.lin
- Result

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

Merci !