CAP 5510: Introduction to Bioinformatics
CGS 5166: Bioinformatics Tools

Giri NARASIMHAN

www.cis.fiu.edu/~giri/teach/BioinfF18.html

Machine Learning

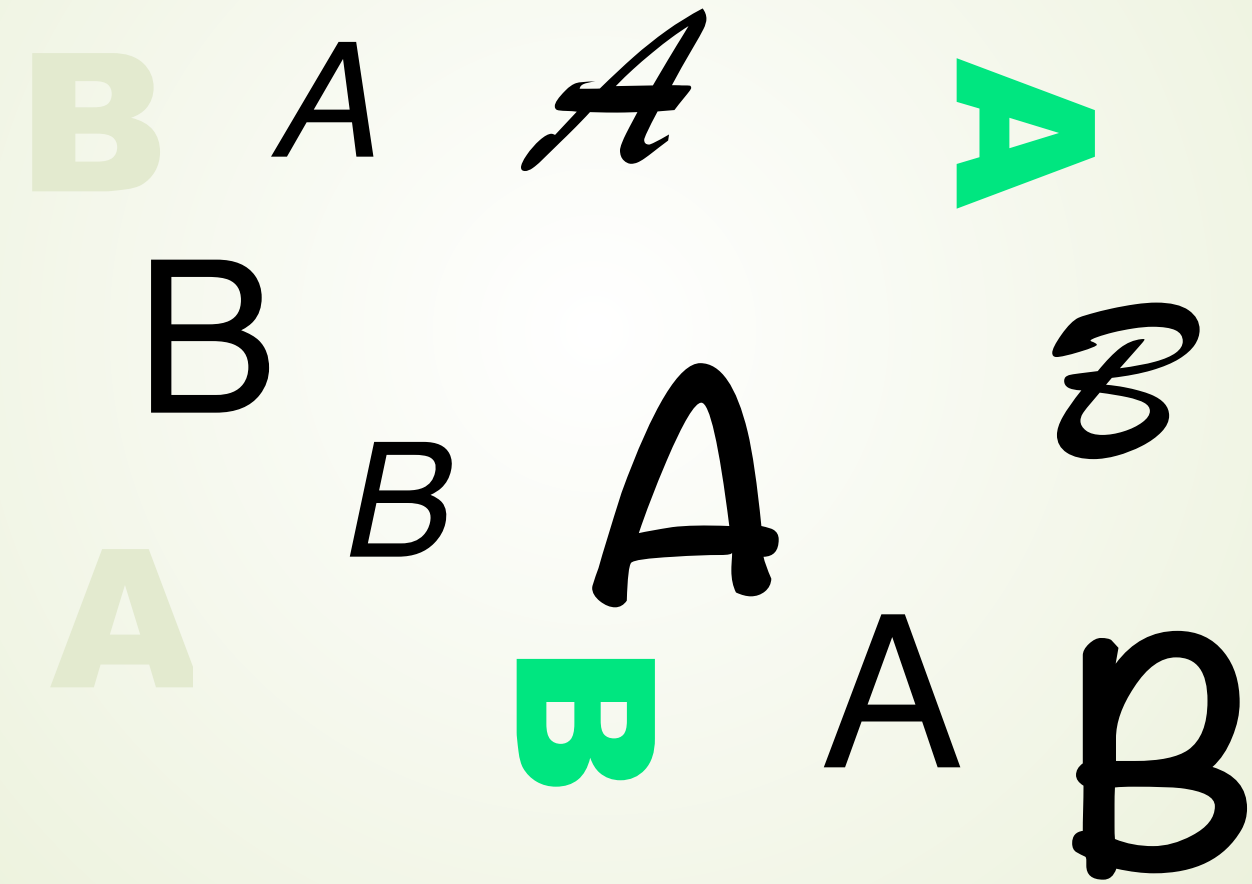
Machine Learning

- Human Endeavor
  - Data → Information → Knowledge
- Machine Learning
  - Automatically extracting information from data
- Types of Machine Learning
  - Unsupervised
    - Clustering
    - Pattern Discovery
  - Supervised
    - Learning
    - Classification

Support Vector Machines

- Supervised Statistical Learning Method for:
  - Classification
  - Regression
- Simplest Version:
  - Training: Present series of labeled examples (e.g., gene expressions of tumor vs. normal cells)
  - Validation: Step to fine-tune hyperparameters
  - Prediction: Predict labels of new examples.

Learning Problems



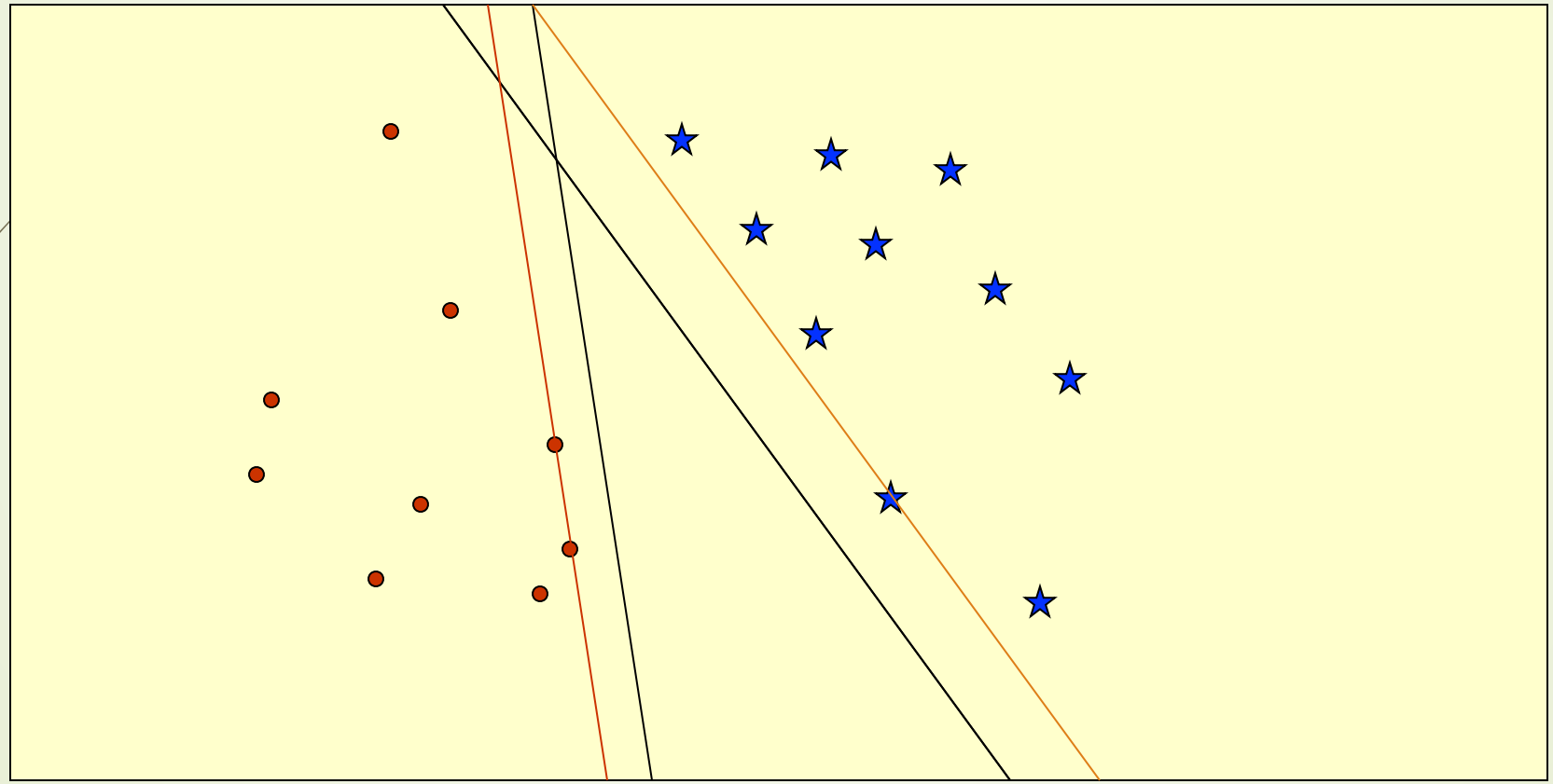
Learning Problems

- Binary Classification
- Multi-class classification
- Regression

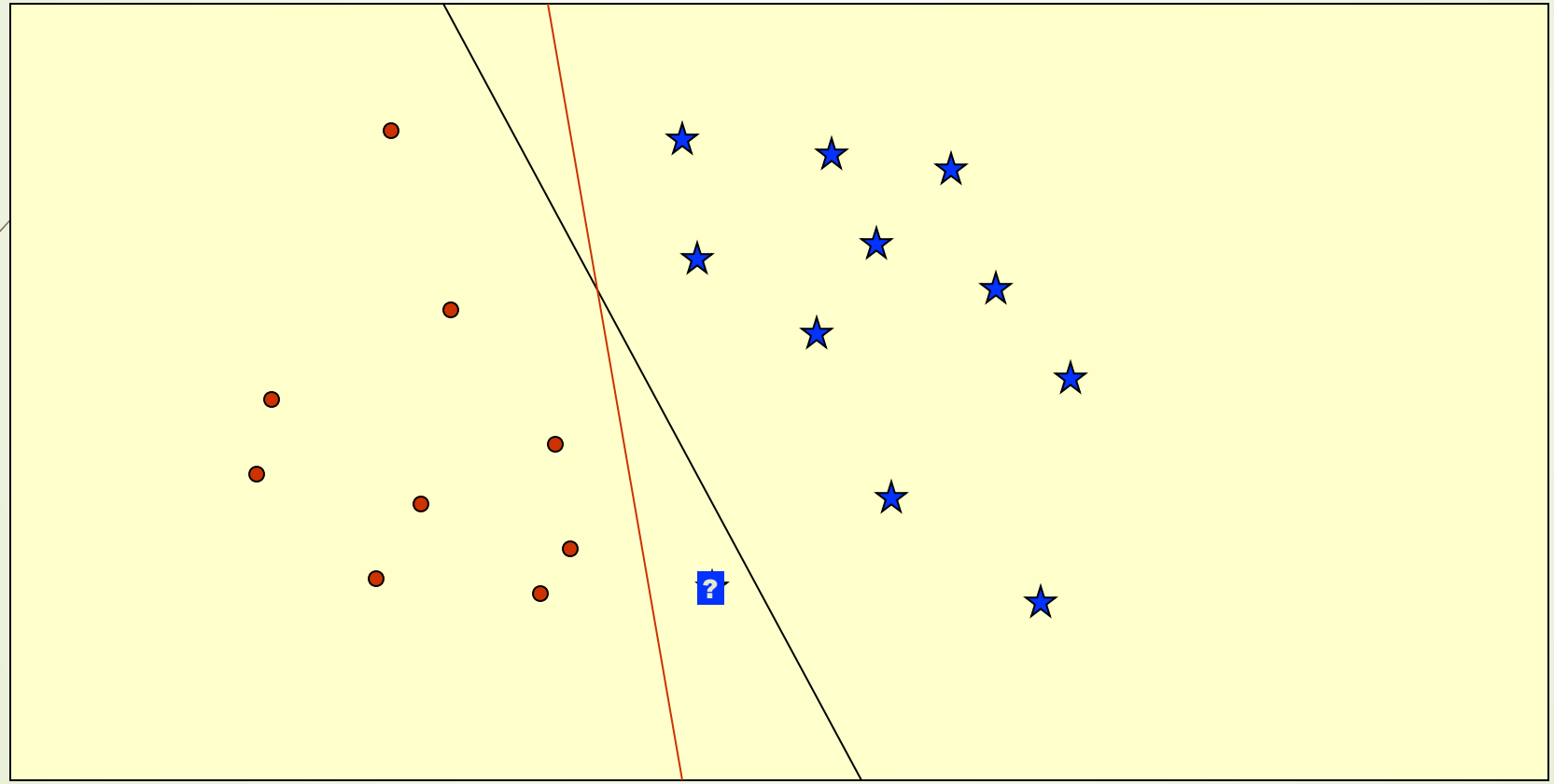
SVM – Binary Classification

- Partition feature space with a surface.
- Surface is implied by a subset of the training points (vectors) near it. These vectors are referred to as **Support Vectors**.
- Efficient with high-dimensional data.
- Solid statistical theory
- Subsume several other methods.

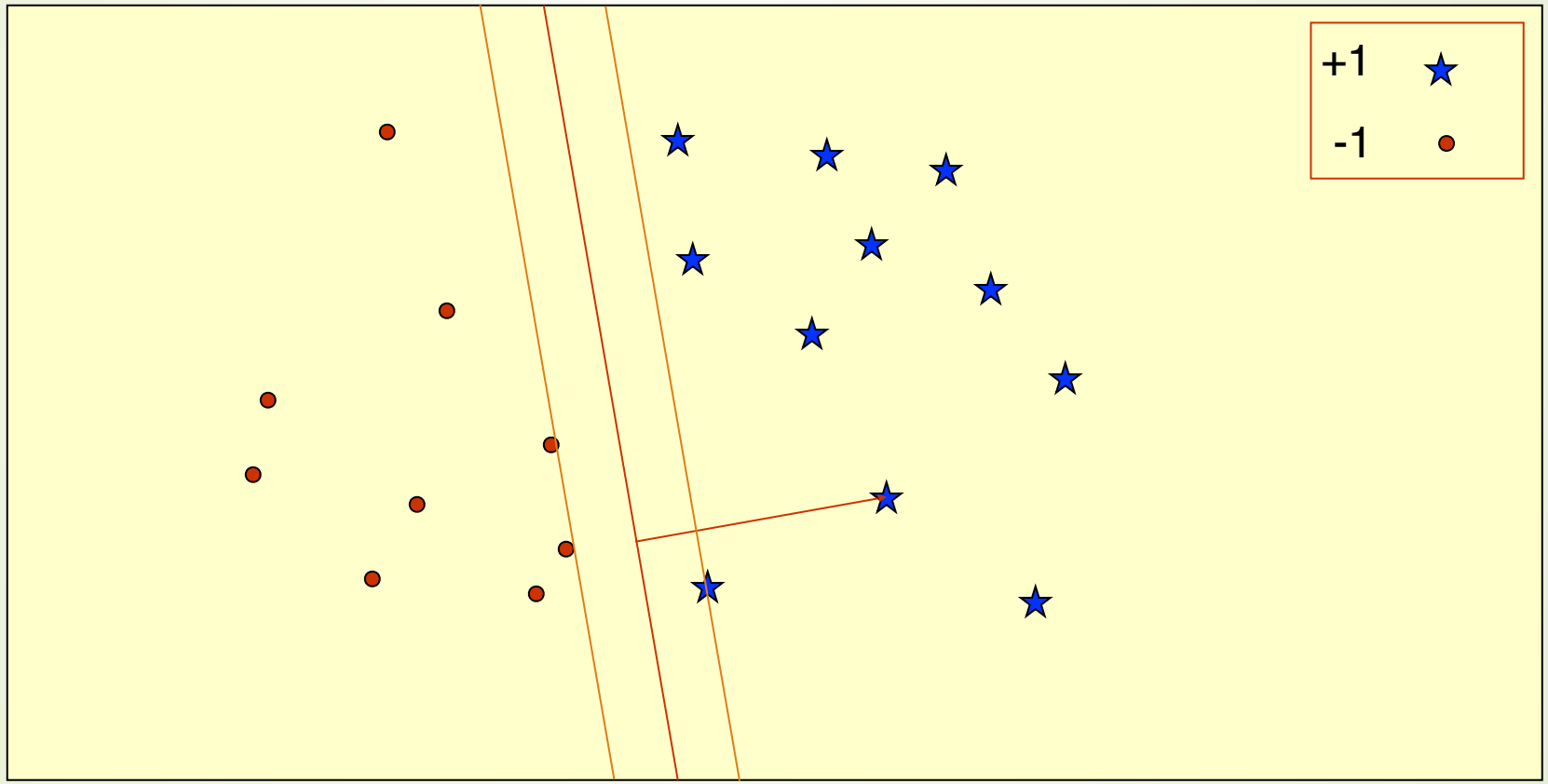
Classification of 2-D (Separable) data



Classification of 2-D (Separable) data

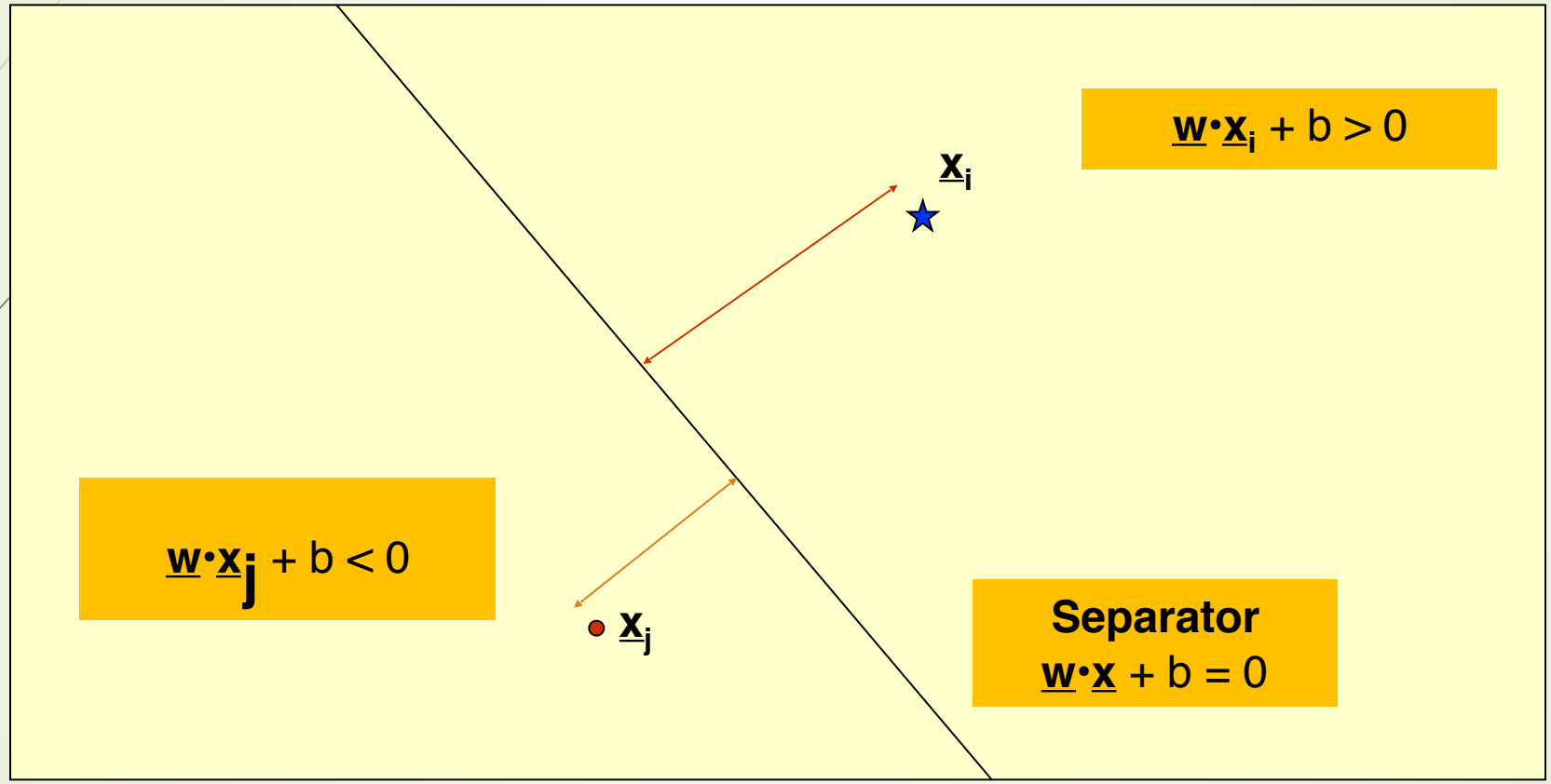


Classification of (Separable) 2-D data



- Margin of a point
- Margin of a point set

Classification using the Separator



Perceptron Algorithm (Primal)

Rosenblatt, 1956

Given separable training set S and learning rate $\eta>0$

$\underline{\mathbf{w}}_0 = \underline{0}$; // Weight

$b_0 = 0$; // Bias

$R = \max |\underline{x}_i|$

$\underline{\mathbf{w}} = \Sigma\, a_i y_i \underline{\mathbf{x}}_i$

repeat

k = 0;

for i = 1 to N

if $y_i\,(\underline{\mathbf{w}}_k \cdot \underline{x}_i + b_k) \le 0$ **then**

$\underline{\mathbf{w}}_{k+1} = \underline{\mathbf{w}}_k + \eta y_i \underline{\mathbf{x}}_i$

$b_{k+1} = b_k + \eta y_i R^2$

$k = k + 1$

Until no mistakes made within loop

Return k, and $(\underline{\mathbf{w}}_k, b_k)$ where k = # of mistakes

Performance for Separable Data

Theorem:

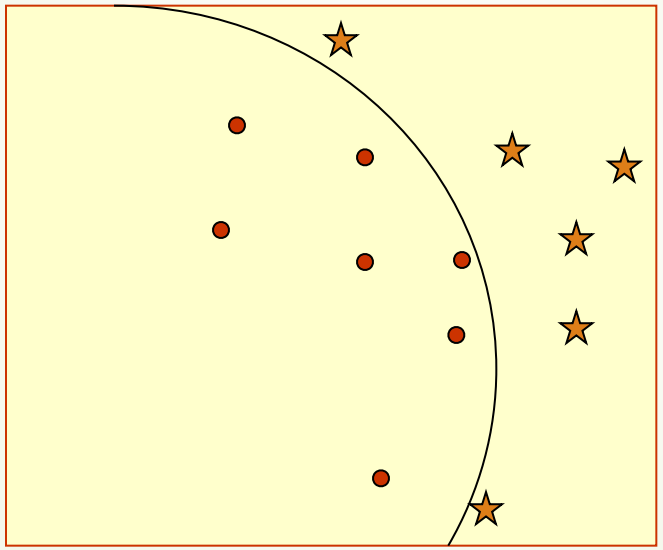
If **margin** m of S is positive, then

$$k \leq (2R/m)^2$$

i.e., the algorithm will always converge,

and will converge quickly.

Non-linear Separators



Main idea: Map into feature space

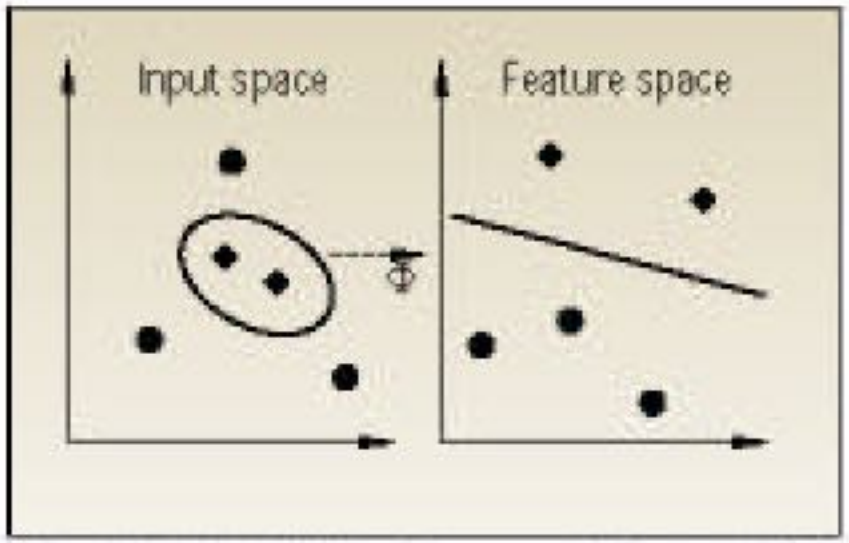
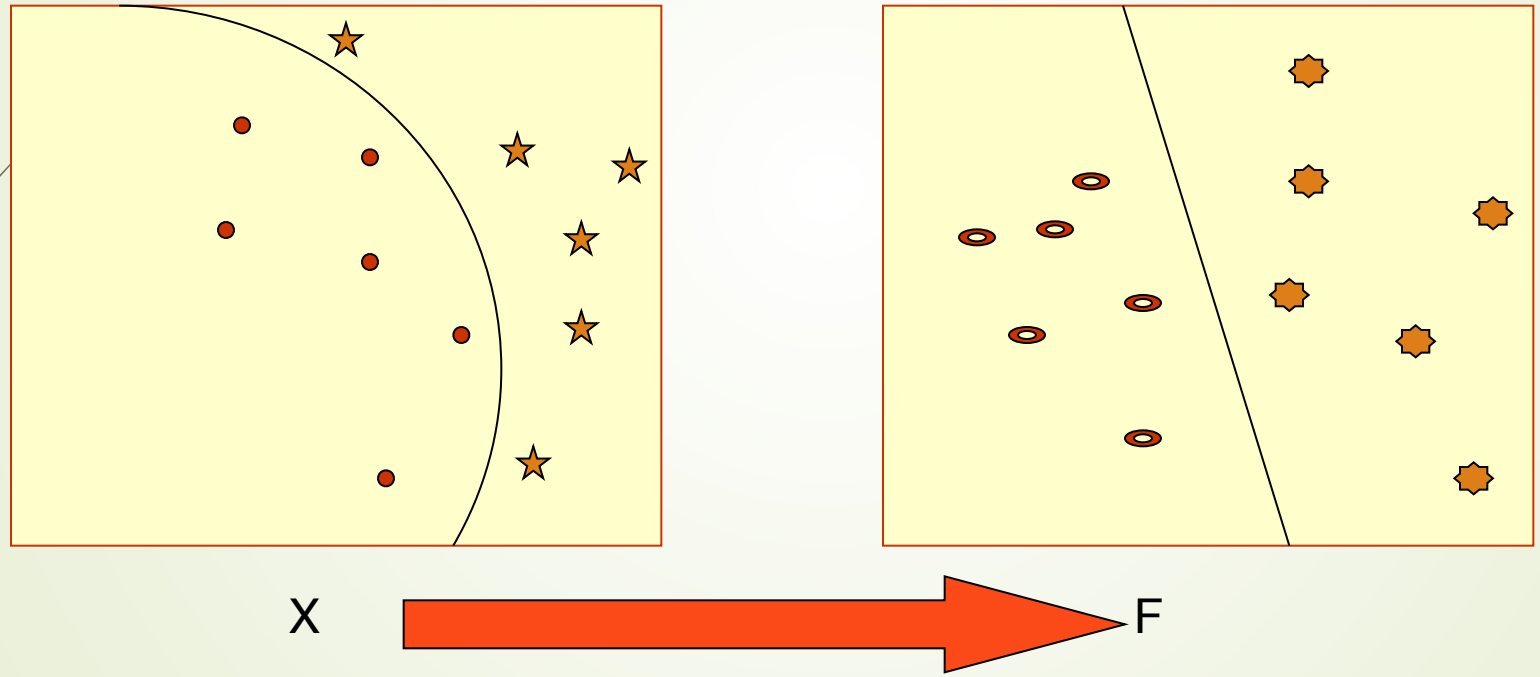


Figure 2. The idea of SV machines: map the training data nonlinearly into a higher-dimensional feature space via Φ, and construct a separating hyperplane with maximum margin there. This yields a nonlinear decision boundary in input space. By the use of a kernel function, it is possible to compute the separating hyperplane without explicitly carrying out the map into the feature space.

Non-linear Separators



Useful URLs

- http://www.support-vector.net

Perceptron Algorithm (Primal)

Rosenblatt, 1956

Given separable training set S and learning rate $\eta>0$

$\underline{\mathbf{w}}_0 = \underline{0}$; // Weight

$b_0 = 0$; // Bias

$R = \max |\underline{x}_i|$

repeat

$k = 0$;

for $i = 1$ to N

if $y_i (\underline{\mathbf{w}}_k \cdot \underline{x}_i + b_k) \leq 0$ **then**

$\underline{\mathbf{w}}_{k+1} = \underline{\mathbf{w}}_k + \eta y_i \underline{\mathbf{x}}_i$

$b_{k+1} = b_k + \eta y_i R^2$

$k = k + 1$

Until no mistakes made within loop

Return k, and $(\underline{\mathbf{w}}_k, b_k)$ where k = # of mistakes

$\underline{\mathbf{w}} = \Sigma \eta a_i y_i \underline{\mathbf{x}}_i$

Perceptron Algorithm (Dual)

Given a separable training set S
$\underline{\mathbf{a}} = \underline{0}$; $b_0 = 0$;

$R = \max \|\underline{x}_i\|$

repeat
  for i = 1 to N
    if $y_i \left(\sum \eta\, a_j y_j\, \underline{x}_i \cdot \underline{x}_j + b\right) \le 0$ **then**
      $a_i = a_i + 1$
      $b = b + y_i R^2$
    endif
Until no mistakes made within inner for-loop
Return ($\underline{\mathbf{a}}$, b)

Perceptron Algorithm (Dual)

Given a separable training set S
$\mathbf{a} = \underline{0}$; $b_0 = 0$;

$R = \max |\underline{x}_i|$

repeat
  for i = 1 to N
    if $y_i \left(\sum a_j y_j \Phi'(\underline{x}_i, \underline{x}_j) + b\right) \leq 0$ **then**
      $a_i = a_i + 1$
      $b = b + y_i R^2$
Until no mistakes made within loop
Return ($\mathbf{a}$, b)

$\Phi'(\underline{x}_i, \underline{x}_j) = \Phi(\underline{x}_i) \bullet \Phi(\underline{x}_j)$

Different Kernel Functions

- **Polynomial kernel**

$$\kappa(X, Y) = (X \bullet Y)^d$$

- **Radial Basis Kernel**

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

- **Sigmoid Kernel**

$$\kappa(X, Y) = \tanh(\omega(X \bullet Y) + \theta)$$

SVM Ingredients

- Support Vectors
- Mapping from Input Space to Feature Space
- Dot Product – Kernel function
- Weights

Generalizations

- How to deal with more than 2 classes?
  - Idea: Associate weight and bias for each class.
- How to deal with non-linear separator?
  - Idea: Support Vector Machines.
- How to deal with linear regression?
- How to deal with non-separable data?

Applications

- **Text Categorization & Information Filtering**
  - 12,902 Reuters Stories, 118 categories (91% !!)
- **Image Recognition**
  - Face Detection, tumor anomalies, defective parts in assembly line, etc.
- **Gene Expression Analysis**
- **Protein Homology Detection**

Class	Method	Learned threshold					Optimized threshold				
		FP	FN	TP	TN	Cost	FP	FN	TP	TN	Cost
Thioridazine acid	Radial SVM	5	5	7	2442	24	4	7	10	2446	18
	Dot-product-1 SVM	11	9	5	2436	29	3	4	11	2447	15
	Dot-product-2 SVM	5	10	7	2445	25	4	4	11	2446	16
	Dot-product-3 SVM	4	10	5	2447	26	4	4	11	2446	16
	Parzen	4	12	5	2440	28	0	12	5	2450	24
	FLD	9	10	3	2441	29	1	9	8	2445	19
	C4.5	7	17	0	2443	41					
	MOC1	3	16	1	2445	35	–	–	–	–	–
Respiration	Radial SVM	9	6	24	2428	21	8	4	26	2429	16
	Dot-product-1 SVM	21	10	20	2416	41	6	5	21	2431	16
	Dot-product-2 SVM	7	14	15	2430	35	5	4	24	2430	19
	Dot-product-3 SVM	3	15	15	2434	33	5	4	24	2430	19
	Parzen	22	10	20	2415	42	7	12	18	2430	31
	FLD	10	10	25	2427	30	14	4	26	2423	22
	C4.5	18	17	13	2416	89	–	–	–	–	–
	MOC1	12	26	4	2425	64	–	–	–	–	–
Isoleucine	Radial SVM	9	4	110	2330	17	2	1	120	2340	8
	Dot-product-1 SVM	13	6	115	2338	25	18	1	120	2325	13
	Dot-product-2 SVM	7	10	118	2336	27	9	1	120	2334	11
	Dot-product-3 SVM	3	28	103	2343	59	7	1	120	2339	9
	Parzen	6	8	113	2340	22	5	5	113	2341	21
	FLD	18	8	116	2328	34	8	3	118	2338	14
	C4.5	31	21	100	2315	73	–	–	–	–	–
	MOC1	26	26	95	2320	78	–	–	–	–	–

Table 2: Comparison of error rates for various classification methods. Classes are described in Table 1. The methods are the radial basis function SVM, the SVMs using the scaled dot product kernel raised to the first, second and third power, Parzen window, Fisher's linear discriminant, and the two decision tree learners, C4.5 and MOC1. The next five columns are the false positive, false negative, true positive and true negative rates summed over three cross validation splits, followed by the cost, which is the number of false positives plus twice the number of false negatives. These five columns are repeated twice, first using the threshold learned from the training set, and then using the threshold that minimizes the cost on the test set. The threshold optimization is not possible for the decision tree methods, since they do not produce ranked results.

Class	Method	Learned threshold					Optimized threshold				
		FP	FN	TP	TN	Cost	FP	FN	TP	TN	Cost
Proline	Radial SVM	3	7	28	2425	17	4	5	30	2428	14
	Dot-product-1 SVM	14	11	24	2414	36	4	7	28	2430	16
	Dot-product-2 SVM	4	13	22	2425	35	4	6	29	2428	16
	Dot-product-3 SVM	3	15	17	2429	36	2	7	28	2430	16
	Parzen	21	5	30	2411	31	4	9	26	2429	21
	FLD	7	12	23	2425	31	12	7	28	2420	26
	C4.5	17	16	25	2415	37	–	–	–	–	–
	MOC1	16	17	18	2422	44	–	–	–	–	–
Histine	Radial SVM	0	2	9	2456	4	0	2	9	2456	4
	Dot-product-1 SVM	0	4	7	2456	8	0	2	9	2456	4
	Dot-product-2 SVM	0	2	9	2456	4	0	2	9	2456	4
	Dot-product-3 SVM	0	2	9	2456	4	0	2	9	2456	4
	Parzen	2	1	8	2454	5	1	3	8	2455	7
	FLD	0	3	8	2456	6	2	1	10	2454	4
	C4.5	2	2	9	2454	6	–	–	–	–	–
	MOC1	2	2	9	2454	6	–	–	–	–	–
Methionine	Radial SVM	1	16	9	2450	33	0	16	9	2451	32
	Dot-product-1 SVM	20	16	9	2431	52	0	16	9	2451	32
	Dot-product-2 SVM	4	16	9	2447	36	0	16	9	2451	32
	Dot-product-3 SVM	1	16	9	2450	33	0	16	9	2451	32
	Parzen	14	16	9	2437	46	0	16	9	2451	32
	FLD	14	16	9	2437	46	0	16	9	2451	32
	C4.5	2	16	9	2449	34	–	–	–	–	–
	MOC1	0	18	9	2449	38	–	–	–	–	–

Table 3: Comparison of error rates for various classification methods (continued). See caption for Table 2.

Class	Kernel	Cost for each split					Total
Citrate cycle	Radial	18	21	15	22	21	97
	Dot-product-1	15	22	18	23	22	100
	Dot-product-2	16	22	17	22	22	99
	Dot-product-3	16	22	17	23	22	100
Respiration	Radial	16	18	19	20	16	93
	Dot-product-1	24	24	25	27	23	127
	Dot-product-2	19	18	20	24	20	11
	Dot-product-3	19	15	20	22	21	107
Ribosome	Radial	8	12	13	11	15	59
	Dot-product-1	15	18	14	16	16	77
	Dot-product-2	11	16	14	16	15	72
	Dot-product-3	9	15	11	15	15	65
Proteasome	Radial	14	16	8	11	11	59
	Dot-product-1	16	12	12	17	18	75
	Dot-product-2	16	13	15	17	17	78
	Dot-product-3	16	13	16	16	17	79
Histone	Radial	4	4	4	4	4	20
	Dot-product-1	4	4	4	4	4	20
	Dot-product-2	4	4	4	4	4	20
	Dot-product-3	4	4	4	4	4	20

Table 4: **Comparison of SVM performance using various kernels.** For each of the MYGD classifications, SVMs were trained using four different kernel functions on five different random three-fold splits of the data, starting at two cost and leaving on the remaining trial. The first column contains the class, as described in Table 1. The second column contains the kernel function, as described in Table 2. The next five columns contain the threshold optimized cost (i.e., the number of false positives plus twice the number of false negatives) for each of the five random three-fold splits. The final column is the total cost summed over five splits.

Family	Gene	Locus	Error	Description
TCA	YPR001W	CIT3	FN	mitochondrial citrate synthase
	YOR142W	LSC1	FN	α subunit of succinyl-CoA ligase
	YNR001C	CIT1	FN	mitochondrial citrate synthase
	YLR174W	IDP2	FN	isocitrate dehydrogenase
	YIL125W	KGD1	FN	α-ketoglutarate dehydrogenase
	YDR148C	KGD2	FN	component of α-ketoglutarate dehydrogenase complex in mitochondria
	YLL066W	IDP3	FN	mitochondrial form of isocitrate dehydrogenase
	YBL015W	ACH1	FP	acetyl CoA hydrolase
Resp	YPR191W	QCR2	FN	ubiquinol cytochrome-c reductase core protein 2
	YPL271W	ATP15	FN	ATP synthase epsilon subunit
	YPL262W	FUM1	FP	fumarase
	YML120C	NDI1	FP	mitochondrial NADH ubiquinone 6 oxidoreductase
	YKL085W	MDH1	FP	mitochondrial malate dehydrogenase
	YDL067C	COX9	FN	subunit VIIb of cytochrome c oxidase
Ribo	YPL037C	EGD1	FP	β subunit of the nascent-polypeptide-associated complex (NAC)
	YLR406C	RPL31B	FN	ribosomal protein L31B (L34-B) (YL78)
	YLR075W	RPL10	FP	ribosomal protein L10
	YAL035W	FUN1	FP	translation elongation factor EF-1γ
Prot	YIL052C	RPN5	FN	subunit of 26S proteasome (PA700 subunit)
	YGL232W	YAP1	FN	member of UBP10/PAS1/SUI10 family of AI Prots
	YGR048W	UFD1	FP	ubiquitin fusion degradation protein
	YDR363W	DOA4	FN	ubiquitin isopeptidase
	YIL020C	RPN4	FN	involved in ubiquitin degradation pathway
Hist	YOL012C	HTA2	FN	histone-associated protein
	YKL049C	CSE4	FN	required for proper kinetochore function

Table 6: **Consistently misclassified genes.** The table lists all 35 genes that are consistently misclassified by SVMs trained using the MYGD classifications listed in Table 1. Two types of errors are included: a false positive (FP) occurs when the SVM includes the gene in the given class but the MYGD classification does not; a false negative (FN) occurs when the SVM does not include the gene in the given class but the MYGD classification does.

Kernel	L^p	Features	FP	FN	FP	FN
dot-product	0	2	5	4	10	12
dot-product	2	2	5	2	12	12
dot-product	5	2	4	2	12	13
dot-product	10	2	4	2	12	13
dot-product	0	50	4	2	12	13
dot-product	2	50	3	2	12	14
dot-product	5	50	3	2	12	14
dot-product	10	50	3	2	12	14
dot-product	0	100	4	3	11	13
dot-product	2	100	5	3	11	12
dot-product	5	100	5	3	11	12
dot-product	10	100	5	3	11	12
dot-product	0	500	5	3	11	13
dot-product	2	500	4	3	11	13
dot-product	5	500	4	3	11	13
dot-product	10	500	4	3	11	13
dot-product	0	**2000**	7	3	11	**10**
dot-product	2	**2000**	5	3	11	12
dot-product	5	**2000**	5	3	11	12
dot-product	10	**2000**	5	3	11	12
dot-product	0	97802	17	**0**	14	**0**
dot-product	2	97802	9	2	12	8
dot-product	5	97802	7	3	11	**10**
dot-product	10	97802	5	3	11	12

Table 1: **Error rates for ovarian cancer tissue experiments.** For each setting of the SVM consisting of a kernel and diagonal factor (DF), each fifth was classified. Column 2 is the number of features (genes) used. Remaining are the number of normal tissues misclassified (FP), tumor tissues misclassified (FN), tumor tissues classified correctly (TP), and normal tissue classified correctly (TN).

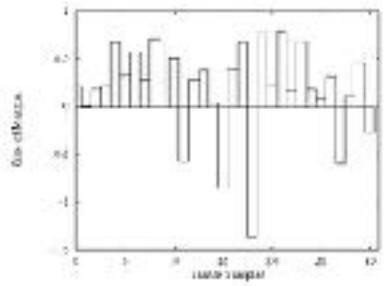


Figure 1: **SVM classification margins for ovarian tissues.** When classifying, the SVM calculates a margin which is the distance of an example from the decision boundary it has learned. In this graph, the margin for each tissue sample calculated using (10) is shown. A positive value indicates a correct classification, and a negative value indicates an incorrect classification. The most negative point corresponds to tissue N539. The second most negative point corresponds to tissue H4804.

Dataset	Features	FP	FN	SVM FP	SVM FN
Ovarian(original)	97802	4.6	3.8	5	3
Ovarian(modified)	97802	4.4	3.4	0	0
AML/ALL train	7129	0.6	2.8	0	0
AML treatment	7129	4.8	3.5	3	2
Colon	2000	3.8	3.7	3	3

Table 5: **Results for the perceptron on all data sets.** The results are averaged over 5 shuffings of the data as this algorithm is sensitive to the order in which it receives the data points. The first column is the dataset used and the second is number of features in the dataset. For the ovarian and colon datasets, the number of normal tissues misclassified (FP) and the number of tumor tissues misclassified (FN) is reported. For the AML/ALL training dataset, the number of AML samples misclassified (FP) and the number of ALL patients misclassified (FN) is reported. For the AML treatment dataset, the number of unsuccessfully treated patients misclassified (FP) and the number of successfully treated patients misclassified (FN) is reported. The last two columns report the best score obtained by the SVM on that dataset.