E9 205 Machine Learning for Sensory Signals

Support Vector Machines

30-03-2017

SVM Formulation

- Goal - **1) Correctly classify all training data**

$$\left.\begin{array}{l}\mathbf{w}^T\phi(\mathbf{x}_n)+b\geq 1 \quad if \quad t_n=+1\\ \mathbf{w}^T\phi(\mathbf{x}_n)+b\leq 1 \quad if \quad t_n=-1\end{array}\right\}$$

$$t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)\geq 1$$

2) Define the Margin

$$\frac{1}{||\mathbf{w}||}min_n \; [t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)]$$

3) Maximize the Margin

$$argmax_{\mathbf{w},b}\left\{\frac{1}{||\mathbf{w}||}min_n \; [t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)]\right\}$$

- Equivalently written as

$$argmin_{\mathbf{w},b}\,\frac{1}{2}||\mathbf{w}||^2 \text{ such that } t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)\geq 1$$

Solving the Optimization Problem

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* a_n is associated with every constraint in the primary problem:
- The dual problem in this case is maximized

Find $\{a_1, .., a_N\}$ such that

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} t_n t_m a_n a_m k(\mathbf{x}_n, \mathbf{x}_m)$$ maximized

and $\sum_n a_n t_n = 0$, $a_n \geq 0$

Solving the Optimization Problem

- The solution has the form:

$$\mathbf{w} = \sum_{n=1}^{N} a_n \phi(\mathbf{x}_n)$$

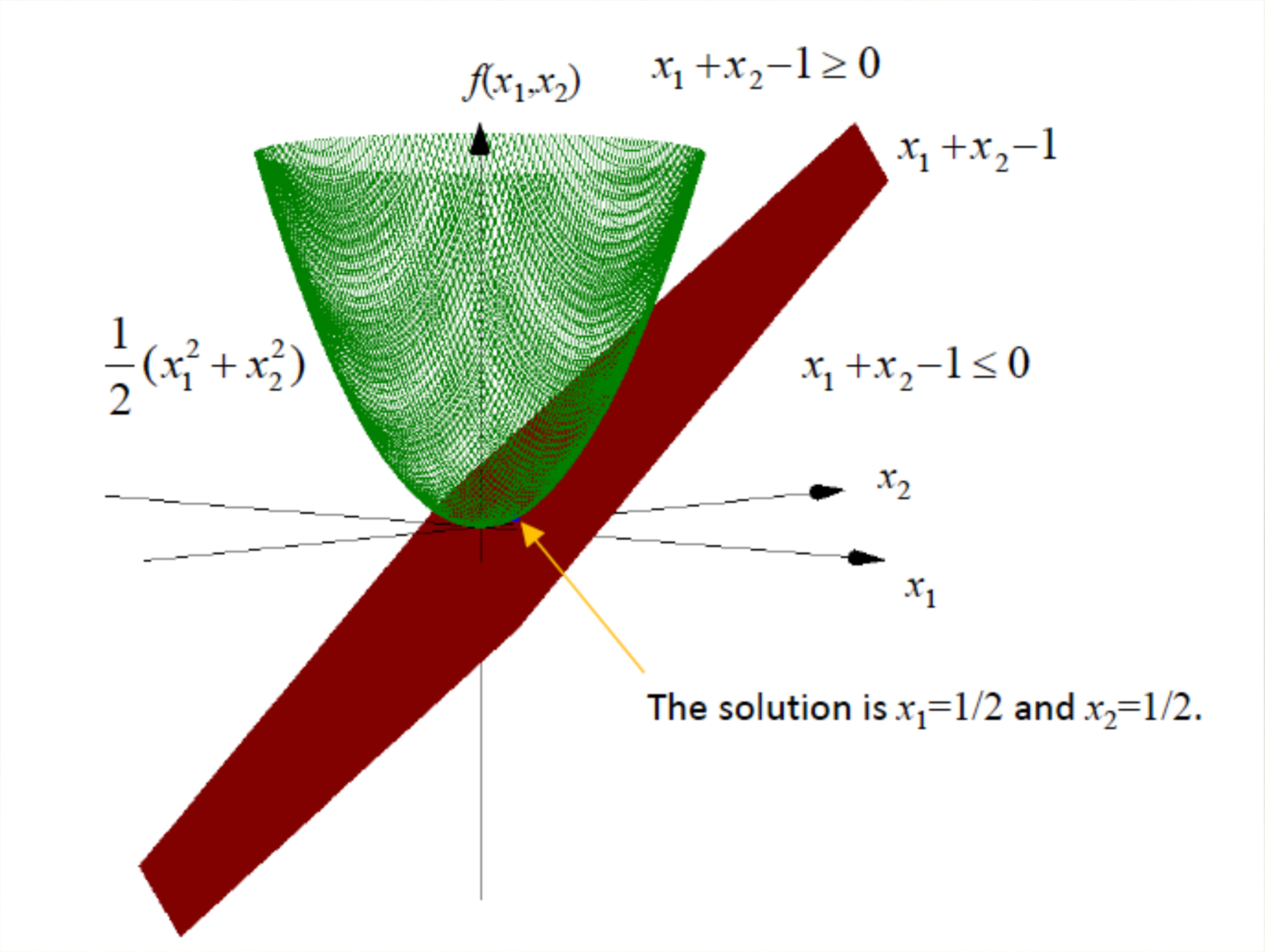
- Each non-zero a_n indicates that corresponding $\mathbf{x_n}$ is a support vector. Let S denote the set of support vectors.

$$b = y(\mathbf{x}_n) - \sum_{m \in S} a_m k(\mathbf{x}_m, \mathbf{x}_n)$$

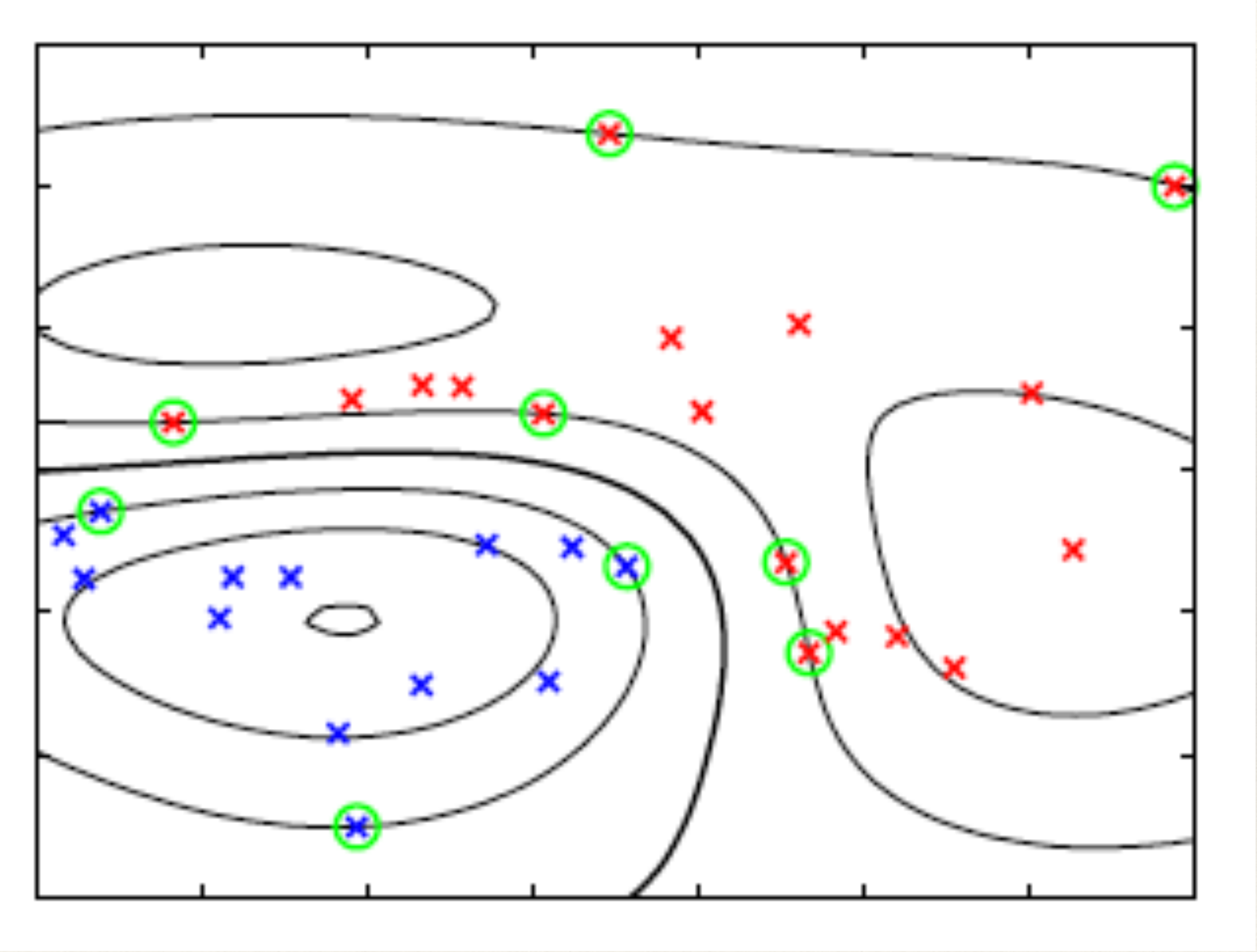
- And the classifying function will have the form:

$$y(\mathbf{x}) = \sum_{n \in S} a_n k(\mathbf{x}_n, \mathbf{x}) + b$$

Solving the Optimization Problem



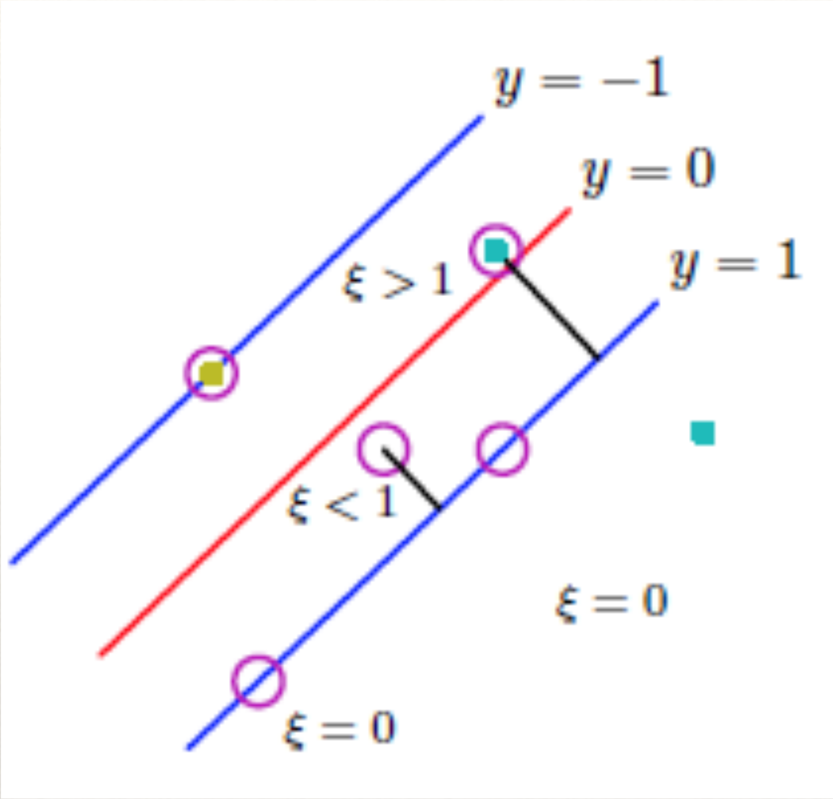
Visualizing Gaussian Kernel SVM



Overlapping class boundaries

- The classes are not linearly separable - Introducing slack variables ζ_n
- Slack variables are non-negative $\zeta_n \geq 0$
- They are defined using

$$t_n y(\mathbf{x}_n) \geq 1 - \zeta_n$$



- The upper bound on mis-classification

$$\sum_n \zeta_n$$

- The cost function to be optimized in this case

$$C \sum_n \zeta_n + \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

SVM Formulation - overlapping classes

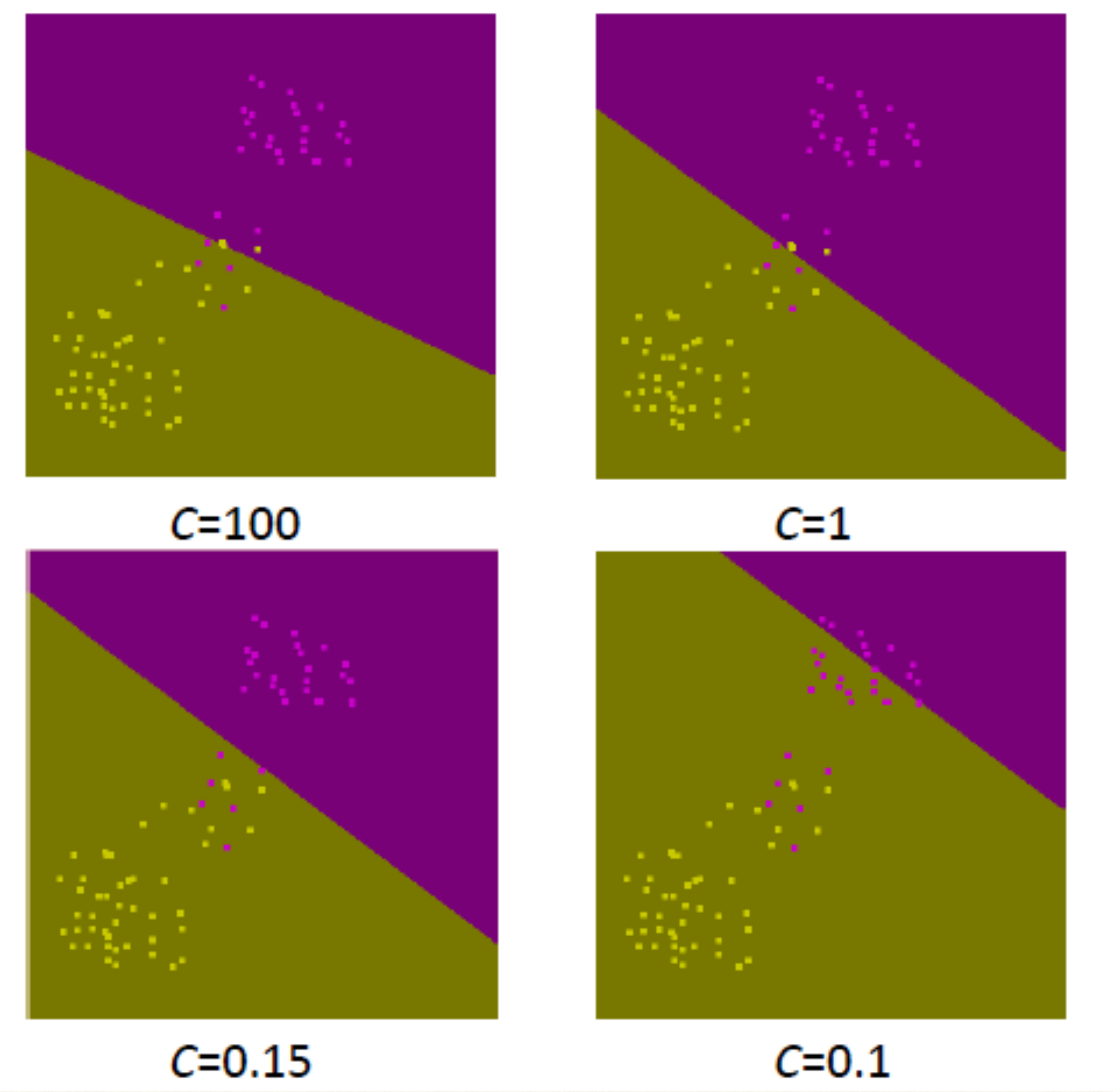
- Formulation very similar to previous case except for additional constraints

$$0 \leq a_n \leq C$$

- Solved using the dual formulation - sequential minimal optimization algorithm
- Final classifier is based on the sign of

$$y(\mathbf{x}) = \sum_{n \in S} a_n k(\mathbf{x}_n, \mathbf{x}) + b$$

Overlapping class boundaries



Properties of SVM

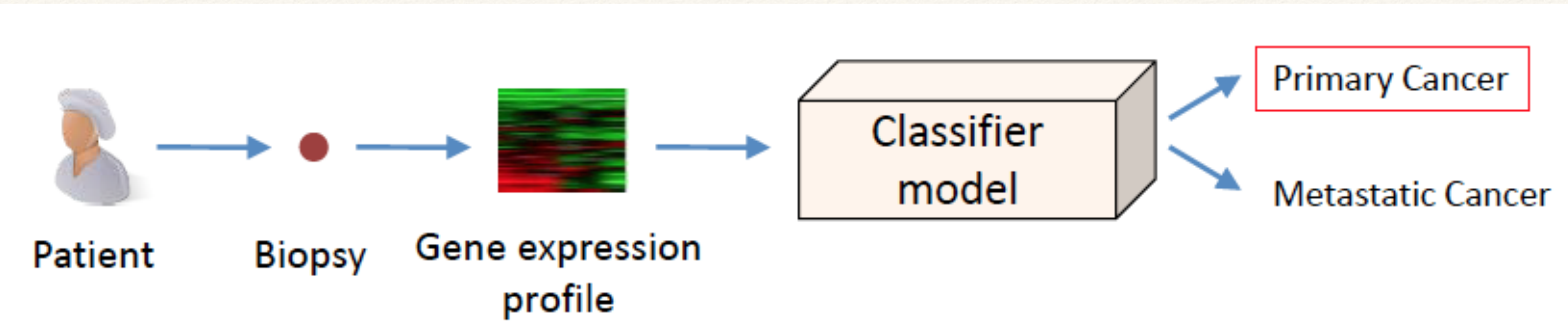
- Flexibility in choosing a similarity function
- Sparseness of solution when dealing with large data sets
  - only support vectors are used to specify the separating hyperplane
- Ability to handle large feature spaces
  - complexity does not depend on the dimensionality of the feature space
- Overfitting can be controlled by soft margin approach
- Nice math property: a simple convex optimization problem which is guaranteed to converge to a single global solution
- Feature Selection

SVM Applications

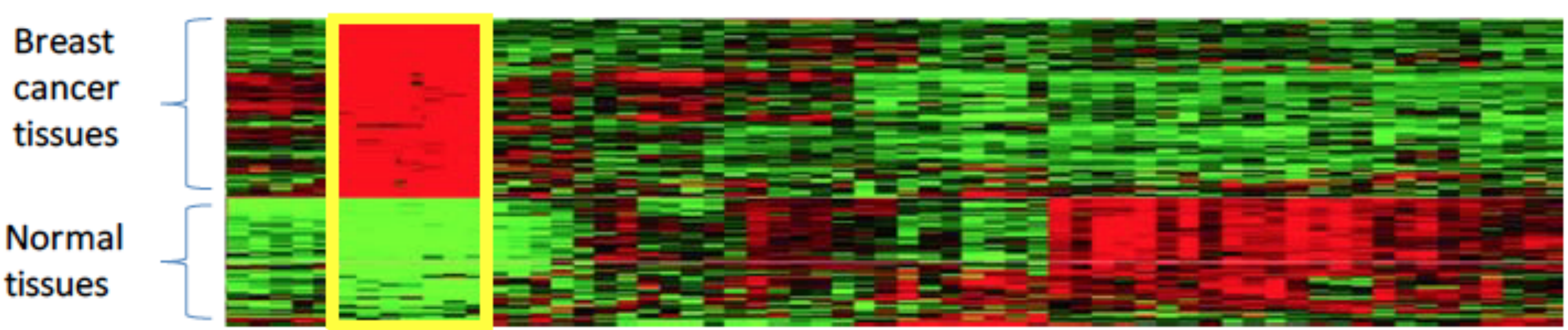
- SVM has been used successfully in many real-world problems

  - text (and hypertext) categorization
  - image classification
  - bioinformatics (Protein classification, Cancer classification)
  - hand-written character recognition

Application 1: Cancer Classification



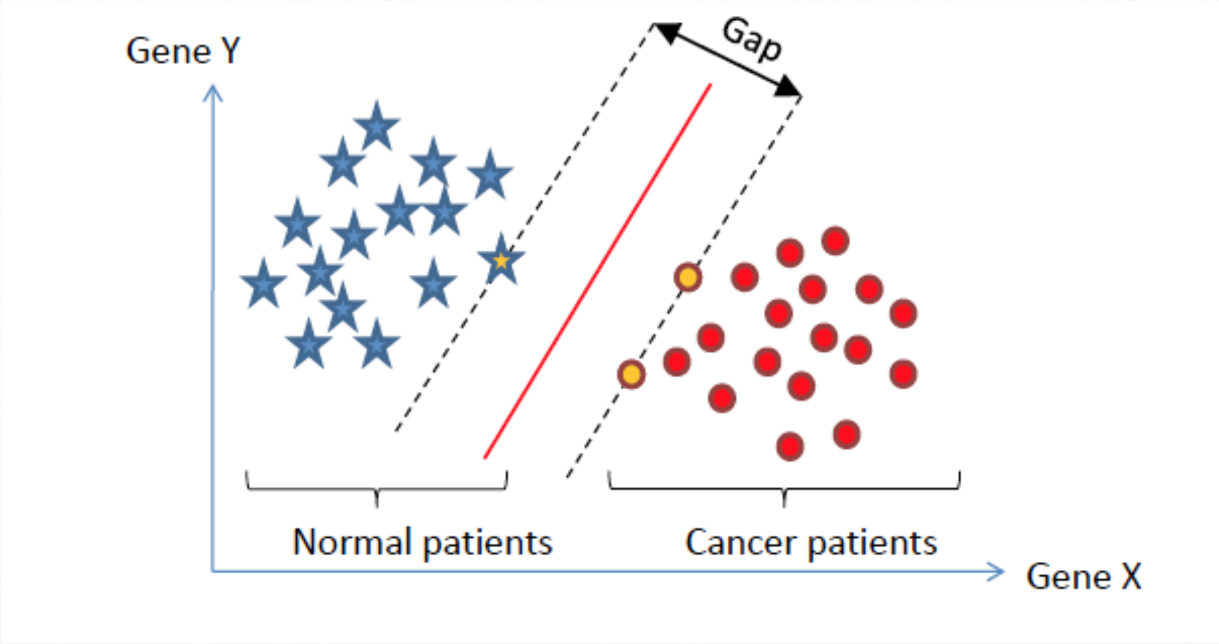
- E.g., find the most compact panel of breast cancer biomarkers from microarray gene expression data for 20,000 genes:

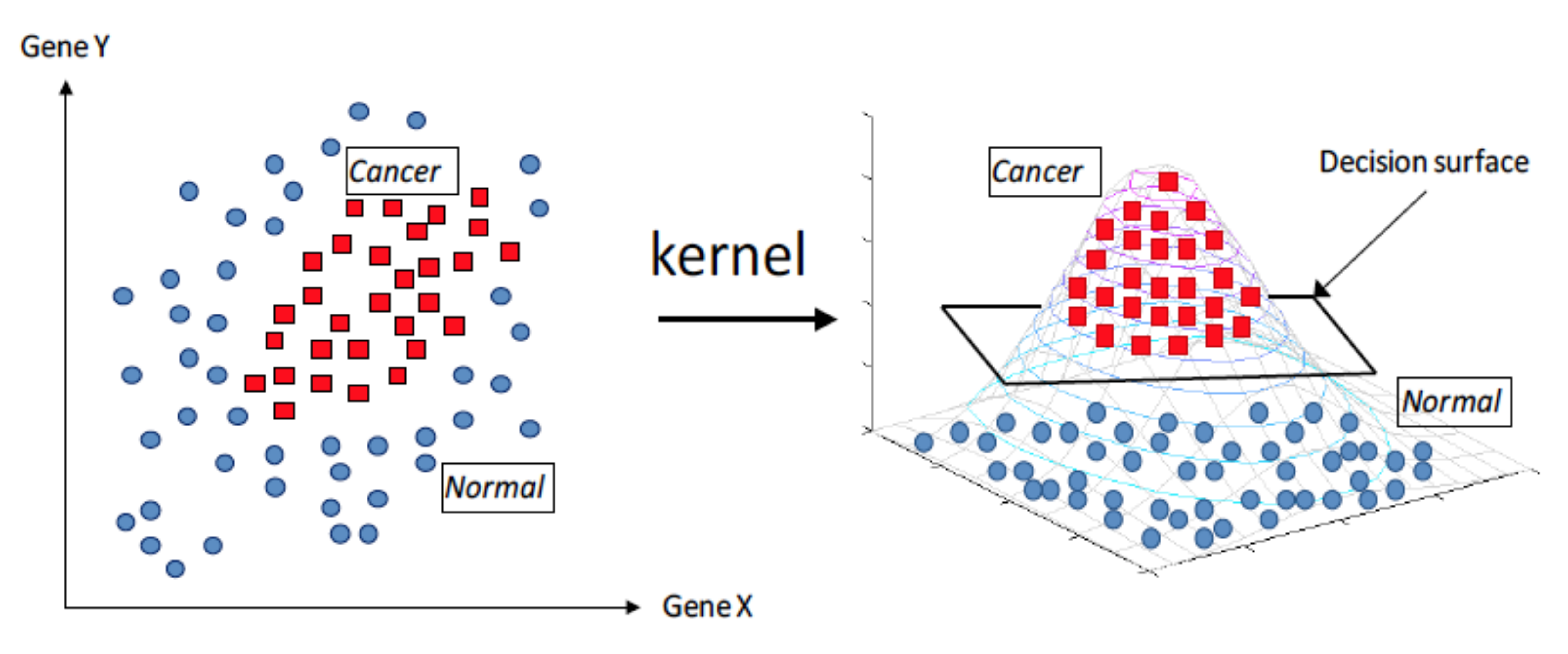


“Gentle Introduction to SVMs in Biomedicine”, Statnikvo et al. NYU school of medicine

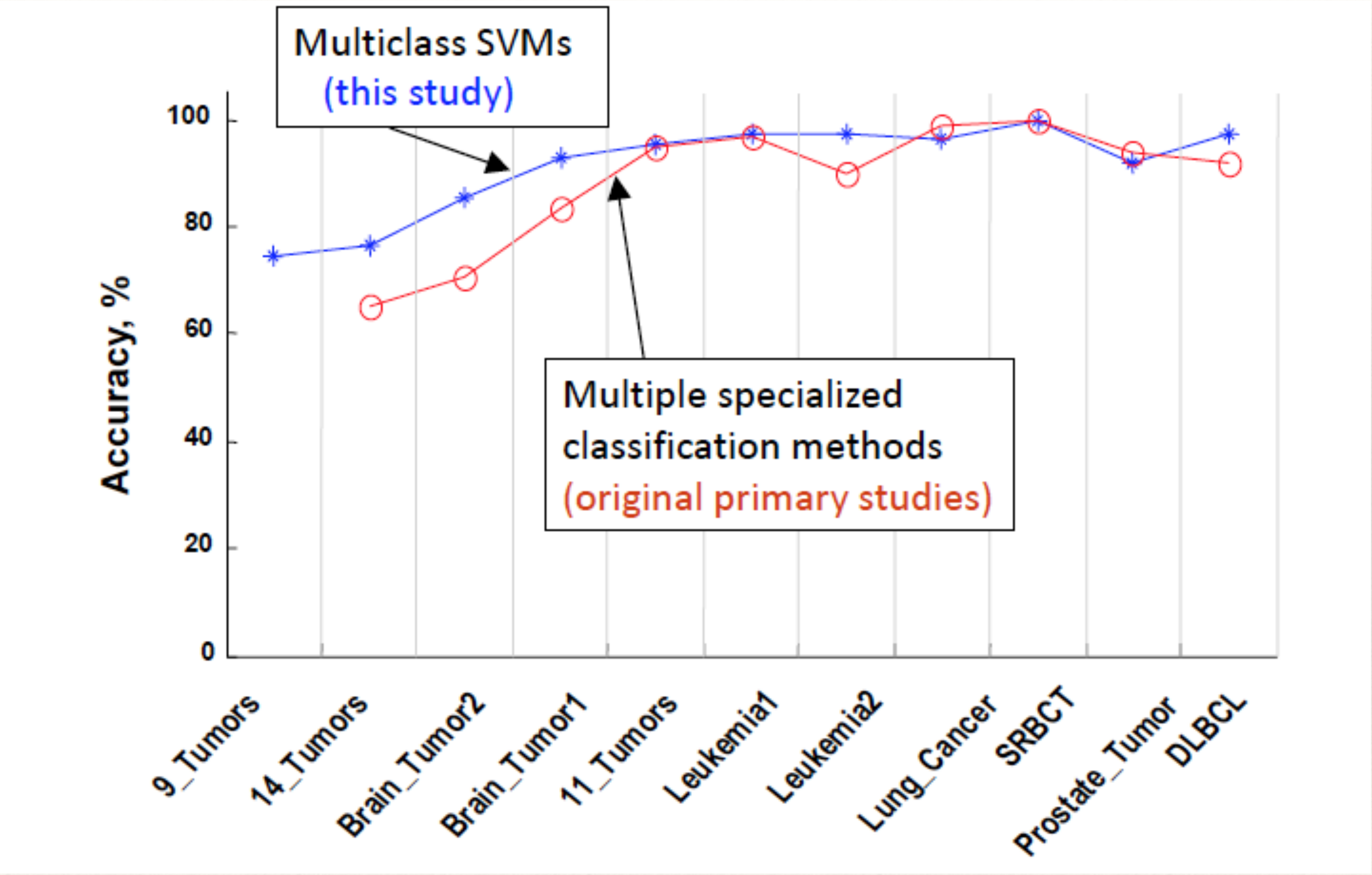
Application 1: Cancer Classification

Linear Versus Non-linear SVMs





Application 1: Cancer Classification



"Gentle Introduction to SVMs in Biomedicine", Statnikvo et al. NYU school of medicine

Weakness of SVM

- **It is sensitive to noise**

  - A relatively small number of mislabeled examples can dramatically decrease the performance

- **It only considers two classes**

  - how to do multi-class classification with SVM?

  - Answer:

1) with output m, learn m SVM's

- **SVM 1 learns "Output==1" vs "Output != 1"**
- **SVM 2 learns "Output==2" vs "Output != 2"**
- **:**
- **SVM m learns "Output==m" vs "Output != m"**

2)To predict the output for a new input, just predict with each SVM and find out which one puts the prediction the furthest into the positive region.

Application 2: Text Categorization

- Task: The classification of natural text (or hypertext) documents into a fixed number of predefined categories based on their content.

  - email filtering, web searching, sorting documents by topic, etc..

- A document can be assigned to more than one category, so this can be viewed as a series of binary classification problems, one for each category.

Application 2: Text Categorization

IR's vector space model (aka bag-of-words representation)

- A doc is represented by a vector indexed by a pre-fixed set or dictionary of terms
- Values of an entry can be binary or weights

$$\phi_i(x) = \frac{\mathrm{tf}_i \log(\mathrm{idf}_i)}{\kappa},$$

- Doc **x** => $\phi(x)$

Application 2: Text Categorization

- The distance between two documents is $<\boldsymbol{\phi}(\boldsymbol{x})\ \boldsymbol{\phi}(\boldsymbol{z})>$

- $K(\boldsymbol{x},\boldsymbol{z}) = <\boldsymbol{\phi}(\boldsymbol{x})\ \boldsymbol{\phi}(\boldsymbol{z})>$ is a valid kernel, SVM can be used with $K(\boldsymbol{x},\boldsymbol{z})$ for discrimination.

- Why SVM?

 - High dimensional input space

 - Few irrelevant features (dense concept)

 - Sparse document vectors (sparse instances)

 - Text categorization problems are linearly separable

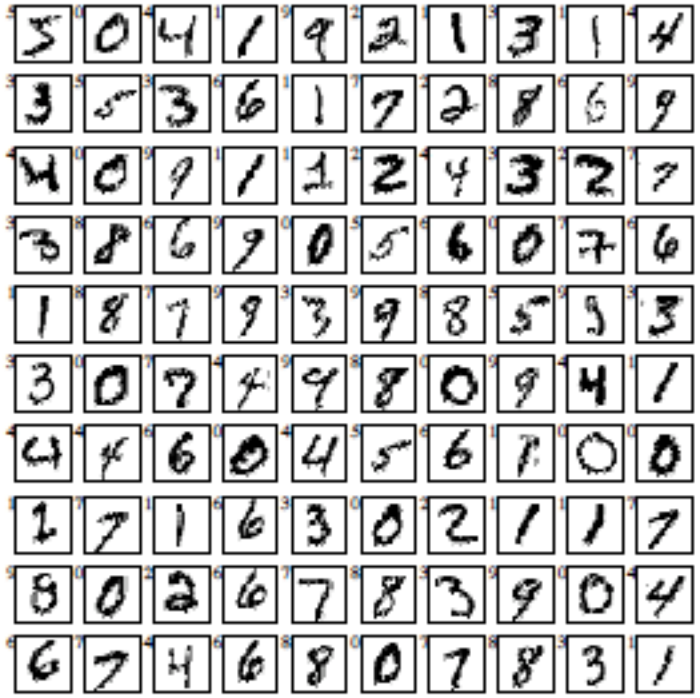
Application 2: Text Categorization

	Bayes	Rocchio	C4.5	k-NN	SVM (poly) degree $d =$					SVM (rbf) width $\gamma =$			
					1	2	3	4	5	0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

Application 3: Handwriting Recognition

For example MNIST hand-writing recognition.
60,000 training examples, 10000 test examples, 28x28.

Linear SVM has around 8.5% test error.
Polynomial SVM has around 1% test error.



SVMs : full MNIST results

Classifier	Test Error
linear	8.4%
3-nearest-neighbor	2.4%
RBF-SVM	1.4 %

Some Considerations

- Choice of kernel
  - Gaussian or polynomial kernel is default
  - if ineffective, more elaborate kernels are needed
  - domain experts can give assistance in formulating appropriate similarity measures

- Choice of kernel parameters
  - e.g. σ in Gaussian kernel
  - σ is the distance between closest points with different classifications
  - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.

- Optimization criterion – Hard margin v.s. Soft margin
  - a lengthy series of experiments in which various parameters are tested

Software

30 SVMs : software

Lots of SVM software:

- LibSVM (C++)
- SVMLight (C)

As well as complete machine learning toolboxes that include SVMs:

- Torch (C++)
- Spider (Matlab)
- Weka (Java)

All available through `www.kernel-machines.org`.