SUPPORT VECTOR MACHINES & NEURAL NETWORKS

LECTURE 5 – SUPPORT VECTOR MACHINES PART # II

A. Bi-classification

History, LSVM, Approximate LSVM, Soft LSVM, Kernel-based linear SVM, nonlinear SVM

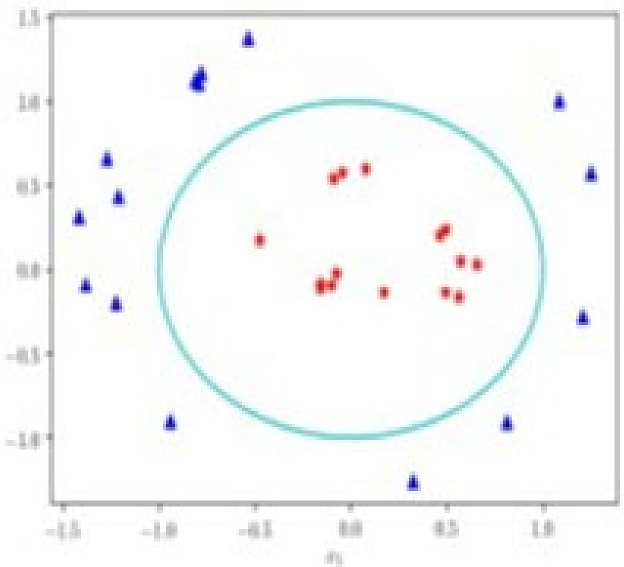
B. Multi-classification

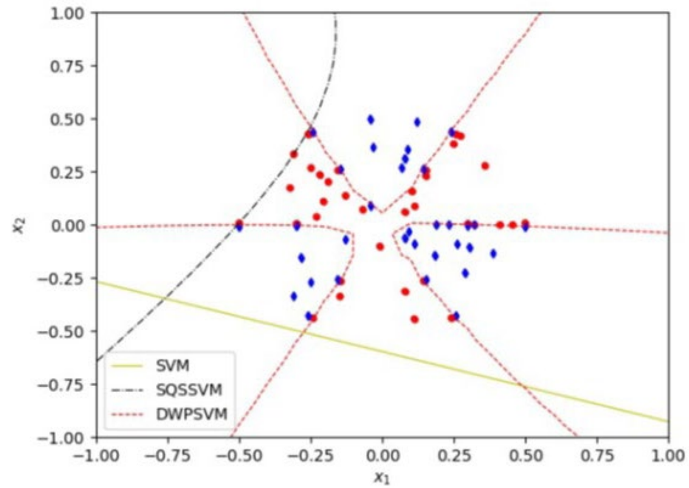
OVO, OVA, Twin SVM

C. Prediction

Support Vector Regression (SVR)

SVM for not linearly separable data sets





- Will LSVM, Approximate LSVM, LSSVM work ?
- How well can they be?
- Any better SVM classifier?

SVM for not linearly separable data sets

- Basic ideas:

  1. Reformulate the problem in a higher dimensional space for linear separability
     (Kernel Method): LSVM with kernel functions

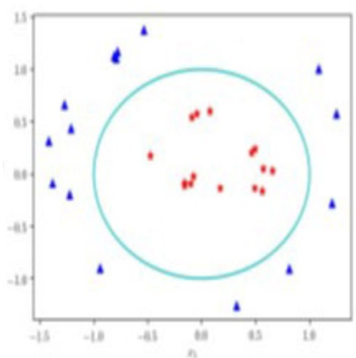
  2. Adopt nonlinear surface to separate data points apart in the original space
     - Quadratic surface SVM
     - Double-well potential function based SVM

Idea of kernel based SVM

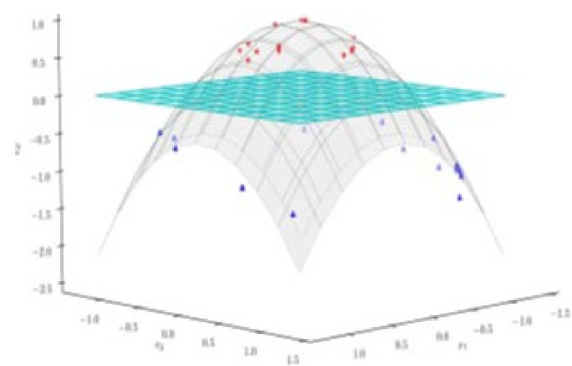
- Feature map: a function $\phi(\cdot)$: $\mathbb{R}^n \rightarrow \mathbb{R}^l$, with $l \geq n$, that maps all data points to a higher dimensional space for linear separation.

- Example 1: $\|x\|_2^2 < 1$, $\|\boldsymbol{x}\|_2^2 > 1$,

$$\phi_1(\boldsymbol{x}): \mathbb{R}^2 \rightarrow \mathbb{R}^3,\ \phi_1(\boldsymbol{x}) = \phi_1\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ 1-x_1^2-x_2^2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{x} \\ 1-\|\boldsymbol{x}\|^2 \end{pmatrix}$$



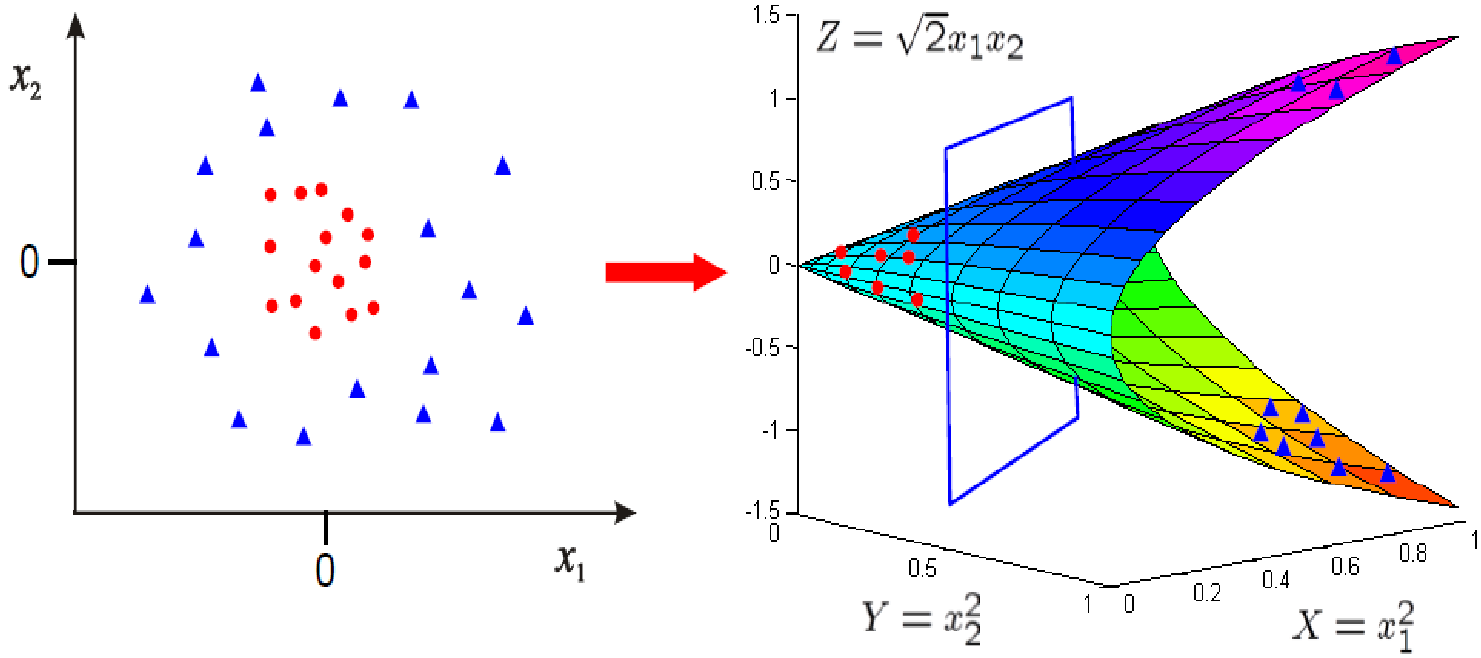
$\phi \rightarrow$



A different feature map

- Picture from Lecture 3, C19 Machine Learning, Hilary 2015, A. Zisserman

$$\phi_2^h\left(\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \left(x_1^2, x_2^2, \sqrt{2}x_1 x_2\right)^T$$



Other feature maps

- Example 2: (quadratic feature)

$\phi_2^h(\boldsymbol{x}): \mathbb{R}^2 \to \mathbb{R}^3$ (homogeneous quadratic feature)

$$\phi_2^h(\boldsymbol{x})^T = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$\phi_2(\boldsymbol{x}): \mathbb{R}^2 \to \mathbb{R}^6$ (inhomogeneous quadratic feature)

$$\phi_2(\boldsymbol{x})^T = \left(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$

Other feature maps

- Example 3: (cubic feature)

$\phi_3^h(\boldsymbol{x})\colon \mathbb{R}^2 \to \mathbb{R}^4$

$$\phi_3^h(\boldsymbol{x})^T = (x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3)$$

$\phi_3(\boldsymbol{x})\colon \mathbb{R}^2 \to \mathbb{R}^{10}$

$$\phi_3(\boldsymbol{x})^T = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3)$$

** What are the effects of ϕ_2 and ϕ_3?

Kernel-based soft SVM - KSSVM

- Using a *feature map* $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^l \ (l \geq n)$ to transform the problem to a higher dimensional space for linear separability.
- Build upon LSSVM
- Primal model

$$\min \quad \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N} \xi_i$$

$$s.t. \quad y_i\left(\boldsymbol{w}^T\phi(\boldsymbol{x}^i) + b\right) \geq 1 - \xi_i, i = 1, \ldots, N \quad \text{(KSSVM)}$$

$$\boldsymbol{w} \in \mathbb{R}^l, \ b \in \mathbb{R}, \ \boldsymbol{\xi} \in \mathbb{R}_+^N$$

where $C > 0$ is a given parameter.

** More variables involved than using LSSVM.

From LSSVM to KSSVM

- Primal models

$$\min \ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi_i$$

$$s.t. \ \ y_i\left(\boldsymbol{w}^T\boldsymbol{x}^i + b\right) \geq 1 - \xi_i, i = 1, \dots, N \quad \text{(LSSVM)}$$

$$\boldsymbol{w} \in \mathbb{R}^n, \ \ b \in \mathbb{R}, \ \ \boldsymbol{\xi} \in \mathbb{R}_+^N$$

where $C > 0$ is a given parameter.

vs.

$$\min \ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{N}\xi_i$$

$$s.t. \ \ y_i\left(\boldsymbol{w}^T\phi(\boldsymbol{x}^i) + b\right) \geq 1 - \xi_i, i = 1, \dots, N \quad \text{(KSSVM)}$$

$$\boldsymbol{w} \in \mathbb{R}^l, \ \ b \in \mathbb{R}, \ \ \boldsymbol{\xi} \in \mathbb{R}_+^N$$

where $C > 0$ is a given parameter.

Kernel-based soft SVM - KSSVM

- SVM classifier

$$class_{SVM}(\boldsymbol{x}) = sign(f(\boldsymbol{x}))$$

- Primal version KSSVM

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b$$

From LSSVM to KSSVM

- Dual models:

$$\max \quad -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i y_i((\boldsymbol{x}^i)^T\boldsymbol{x}^j)y_j\alpha_j + \sum_i^N \alpha_i$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \qquad \text{(DLSSVM)}$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \dots, N$$

then

(DKSSVM) = ?

Dual kernel-based soft SVM (DKSSVM)

- Lagrangian dual model

$$\max -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i y_i \phi(\boldsymbol{x}^i)^T \phi(\boldsymbol{x}^j) y_j \alpha_j + \sum_{i=1}^{N}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \qquad \text{(DKSSVM)}$$

$$0 \le \alpha_i \le C, i = 1, 2, \ldots, N$$

- The “kernel matrix” is defined as $K = \left(K_{ij}\right) \in \boldsymbol{M}_{N\times N}(\mathbb{R})$ with elements K_{ij} such that

$$K_{ij} = K(\boldsymbol{x}^i, \boldsymbol{x}^j) \triangleq \phi\left(\boldsymbol{x}^i\right)^T \phi(\boldsymbol{x}^j)$$

Dual kernel-based soft SVM (DKSSVM)

Kernel matrix

$$K_{ij} = K(\boldsymbol{x}^i, \boldsymbol{x}^j) \triangleq \phi(\boldsymbol{x}^i)^T \phi(\boldsymbol{x}^j)$$

Example 1: For ϕ_1 feature map

$$K_{ij} = ((\boldsymbol{x}^i)^T \ , 1 - \|\boldsymbol{x}^i\|^2) \begin{pmatrix} \boldsymbol{x}^j \\ 1 - \|\boldsymbol{x}^j\|^2 \end{pmatrix} = < \boldsymbol{x}^i, \boldsymbol{x}^j > + (1 - \|\boldsymbol{x}^i\|^2)(1 - \|\boldsymbol{x}^j\|^2)$$

Example 2: For ϕ_2 feature map: $K_{ij} = \phi_2(\boldsymbol{x}^i)^T \phi_2(\boldsymbol{x}^j)$

$$= (1, \sqrt{2}x_1^i, \sqrt{2}x_2^i, (x_1^i)^2, \sqrt{2}x_1^i x_2^i, (x_2^i)^2) \left(1, \sqrt{2}x_1^j, \sqrt{2}x_2^j, \ (x_1^j)^2, \sqrt{2}x_1^j x_2^j, (x_2^j)^2\right)^T$$

$$= 1 + 2\left(x_1^i x_1^j + x_2^i x_2^j\right) + ((x_1^i)^2 (x_1^j)^2 + (x_2^i)^2 (x_2^j)^2) + 2(x_1^i x_2^i x_1^j x_2^j)$$

$$= 1 + 2(\boldsymbol{x}^i)^T \boldsymbol{x}^j + ((\boldsymbol{x}^i)^T \boldsymbol{x}^j)^2$$

$$= ((\boldsymbol{x}^i)^T \boldsymbol{x}^j + 1)^2$$

--- polynomial kernel with $r = 1, d = 2$.

How difficult to solve DKSSVM?

- Lagrangian dual model

$$\max -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\, y_i\, K_{ij}\, y_j\alpha_j + \sum_{i}^{N}\alpha_i$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \qquad \text{(DKSSVM)}$$

$$0 \le \alpha_i \le C, i = 1, 2, \dots, N$$

where $K_{ij} = K(\boldsymbol{x}^i, \boldsymbol{x}^j) \triangleq \phi(\boldsymbol{x}^i)^T \phi(\boldsymbol{x}^j)$

- Given any feature map ϕ, corresponding K is psd and DKSSVM becomes a convex quadratic program with N bounded variables and only one linear equality constraint.
- In practice, we may use a kernel matrix $K = (K_{ij})$ without knowing the feature map $\phi(x)$.

Kernel-based soft SVM - DKSSVM

- SVM classifier

$$class_{SVM}(\boldsymbol{x}) = sign(f(\boldsymbol{x}))$$

Dual version DKSSVM

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \, \phi(\boldsymbol{x}^i)^T \phi(\boldsymbol{x}) + b(\alpha_i)$$

$$= \sum_{i \in S} \alpha_i y_i \, K\left(\boldsymbol{x}^i, \boldsymbol{x}\right) + \bar{b}$$

Kernel matrix

- To make sure that $K_{ij} = K(\boldsymbol{x}^i, \boldsymbol{x}^j)$ is the inner product of $\phi(\boldsymbol{x}^i)$ and $\phi(\boldsymbol{x}^j)$ in the feature space, such that

  (1) DKSSVM is an easily solved convex QP,

  (2) there is a chance to solve KSSVM,

  we need K to be symmetric and positive semidefinite (Mercer's condition).

- Commonly used kernels:

  1. Polynomial kernel of degree $d = 1, 2, \ldots$

  $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = \left(\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j + r\right)^d$ (homogeneous, if $r = 0$)

  (inhomogeneous, if $r > 0$)

  ** popular in image processing*

Polynomial kernels

Example 1: (inhomogeneous degree 2)

For $\boldsymbol{x} \in \mathbb{R}^1, K(x^i, x^j) = (x^i x^j + 1)^2$ for $r = 1, d = 2,$

we have $\phi(x)^T = (1, \sqrt{2}x, x^2) \in \mathbb{R}^3$ such that

$\phi(x^i)^T \phi(x^j) = 1 + 2x^i x^j + (x^i)^2 (x^j)^2 = (x^i x^j + 1)^2$

Example 2: (homogeneous degree 2)

For $\boldsymbol{x} \in \mathbb{R}^2, K(\boldsymbol{x}^i, \boldsymbol{x}^j) = ((\boldsymbol{x}^i)^T \boldsymbol{x}^j)^2$ for $r = 0, d = 2$,

we have $\phi(\boldsymbol{x})^T = (x_1^2, \sqrt{2}x_1 x_2, x_2^2) \in \mathbb{R}^3$ such that

$\phi(\boldsymbol{x}^i)^T \phi(\boldsymbol{x}^j) = (x_1^i)^2 (x_1^j)^2 + (x_2^i)^2 (x_2^j)^2 + 2(x_1^i x_2^i x_1^j x_2^j) = ((\boldsymbol{x}^i)^T \boldsymbol{x}^j)^2$

**General form $\phi(x)$: contains all polynomial terms up to degree d.

Kernel matrix

Commonly used kernels:

2. Gaussian kernel with $\sigma \in \mathbb{R}\backslash\{0\}$

$$K(\boldsymbol{x}^i, \boldsymbol{x}^j) = exp\left(- \frac{\left\|\boldsymbol{x}^i - \boldsymbol{x}^j\right\|_2^2}{2\sigma^2} \right)$$

* no prior information, general purpose

***General form* $\phi(x)$ *in infinite dimensional feature space.*

3. Gaussian Radial basis function (RBF) kernel with $\gamma > 0$

$$K(\boldsymbol{x}^i, \boldsymbol{x}^j) = exp\left(-\gamma \left\|\boldsymbol{x}^i - \boldsymbol{x}^j\right\|_2^2 \right)$$

* no prior information, general purpose

***General form* $\phi(x)$: see https://en.wikipedia.org/wiki/Radial_basis_function_kernel

Kernel matrix

Commonly used kernels:

4. Laplace RBF kernel with $\sigma > 0$

$$K(\boldsymbol{x}^i, \boldsymbol{x}^j) = exp\left(-1/\sigma \left\|\boldsymbol{x}^i - \boldsymbol{x}^j\right\|_2\right)$$

* no prior information, general purpose

5. Sigmoid kernel with $\beta > 0$, $\theta \in \mathbb{R}$

$$K(\boldsymbol{x}^i, \boldsymbol{x}^j) = tanh\left(\beta\left(\boldsymbol{x}^i\right)^T \boldsymbol{x}^j + \theta\right)$$

* proxy for neural networks

Quality of kernel-based SVM

- Two major factors:

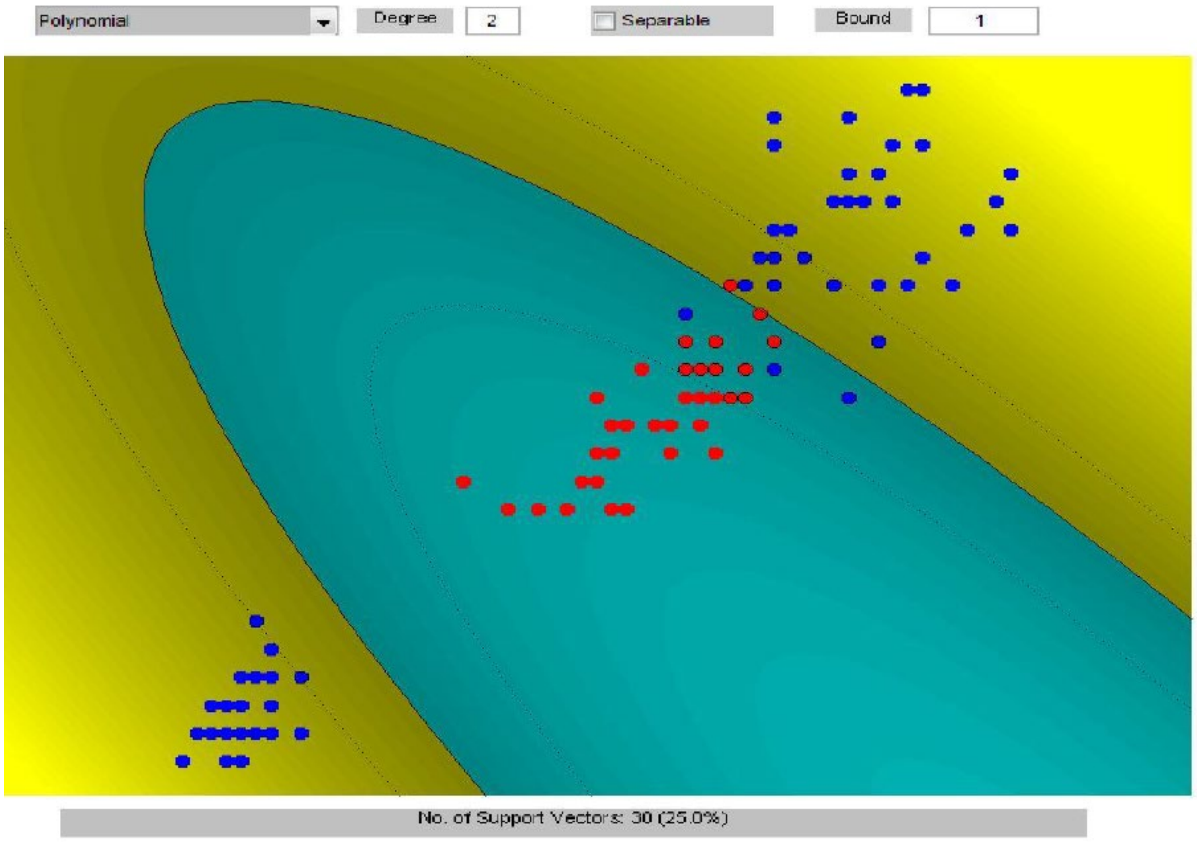
1. Like LSSVM, the parameter C plays a role.

2. The choice of an appropriate kernel matrix (and its parameters) is important.

Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020,CMU

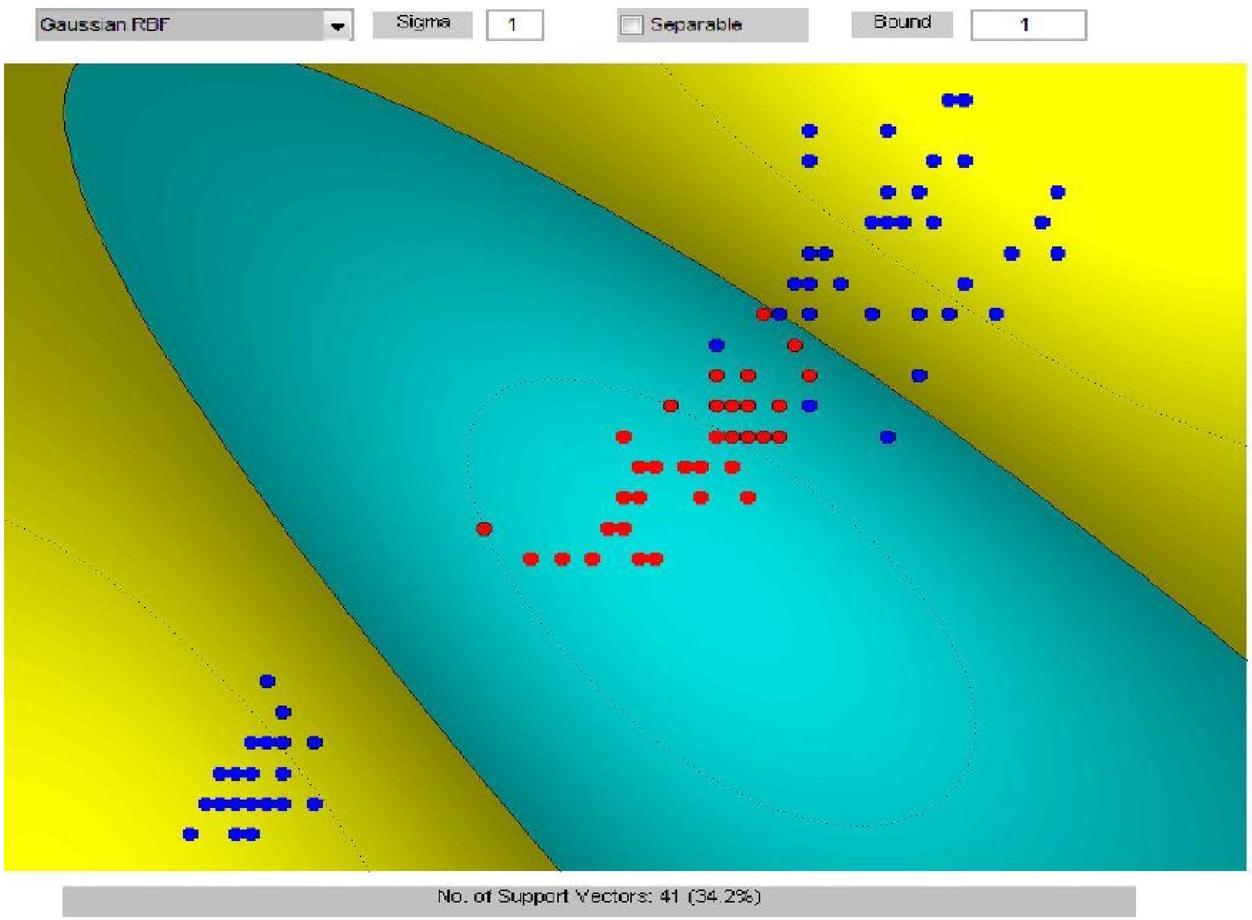
  Iris dataset, 1 vs 23, Polynomial Kernel degree 2 $(C = 1)$



Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020,CMU

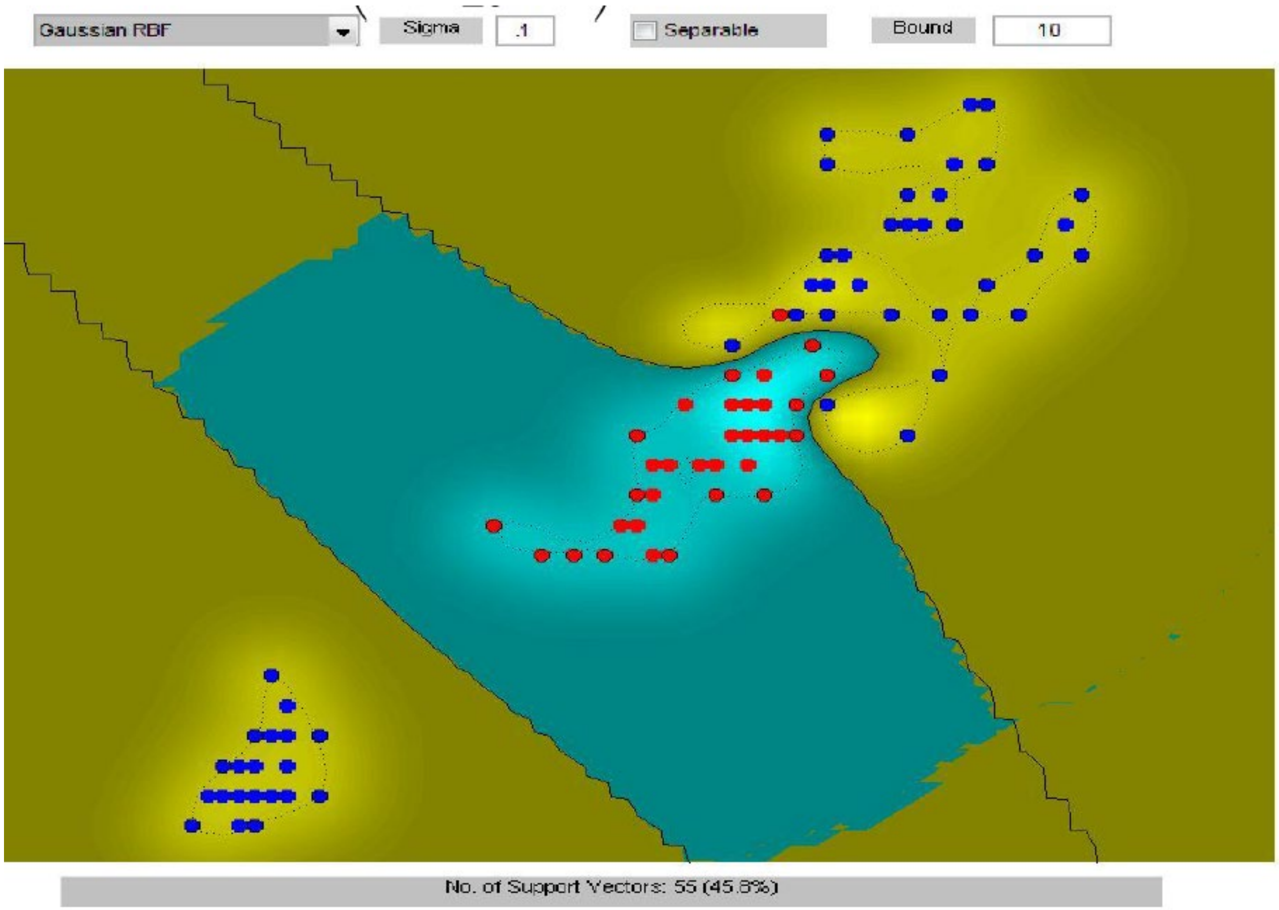
Iris dataset, 1 vs 23, Gaussian RBF kernel ($C = 1, \sigma = 1$)



Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020,CMU

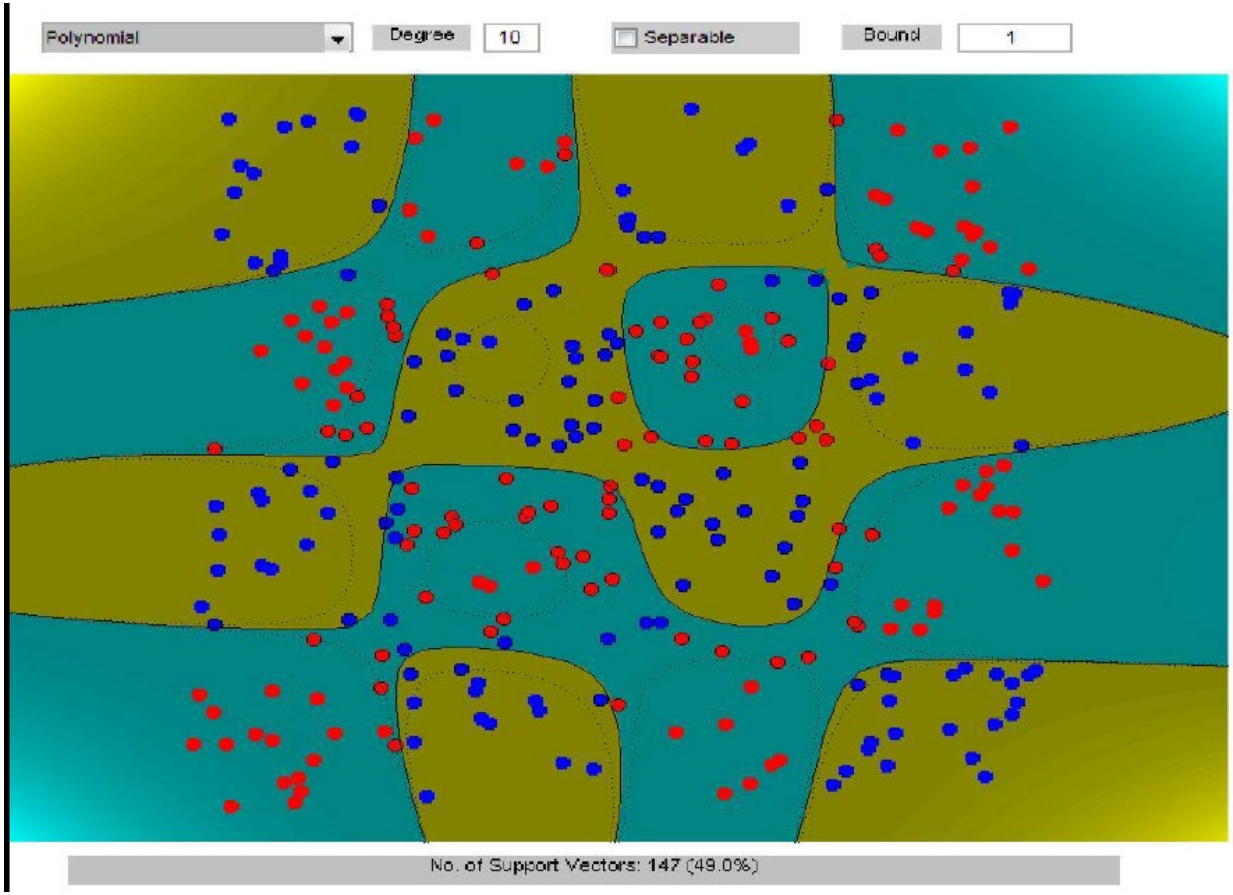
Iris dataset, 1 vs 23, Gaussian RBF kernel $(C = 10, \sigma = 1)$



Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020,CMU

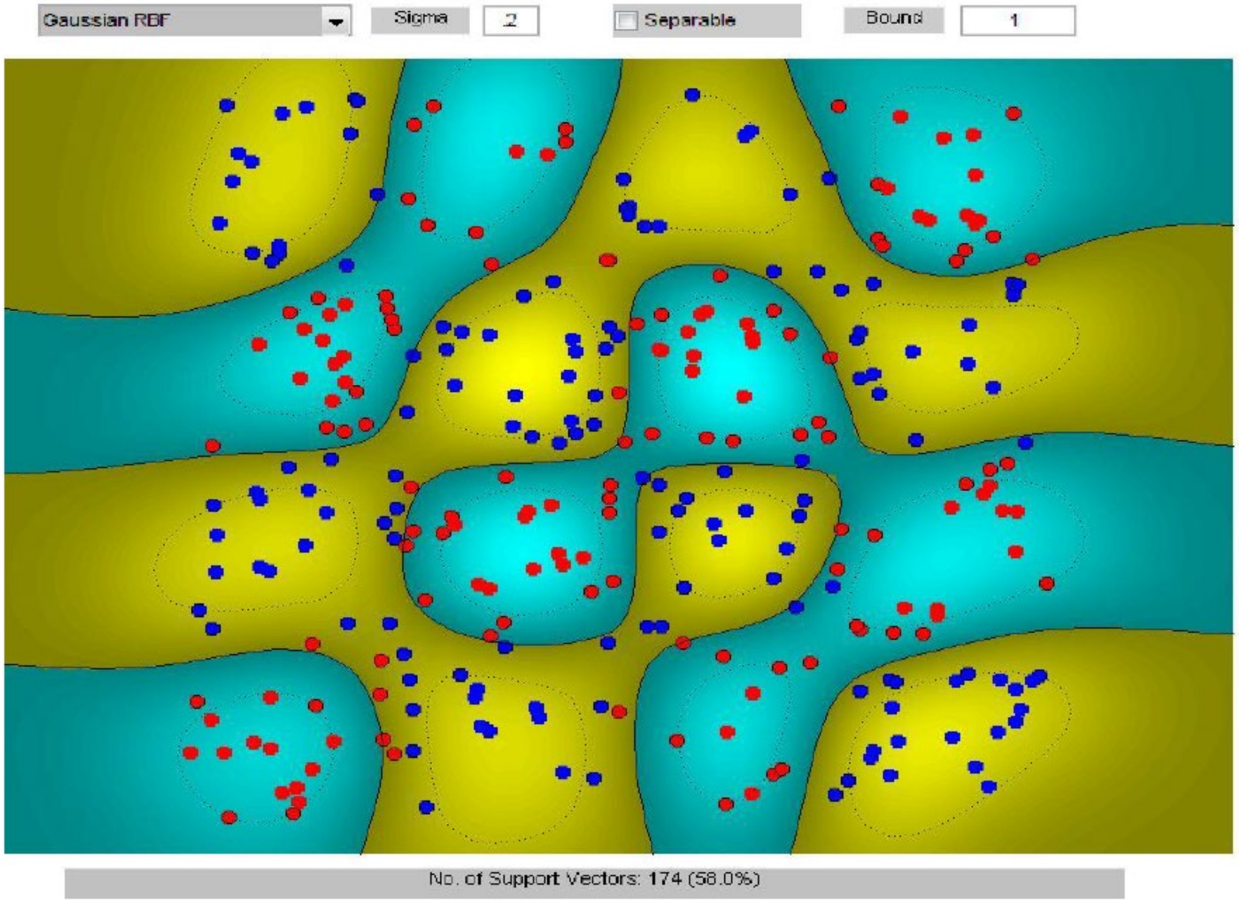
Chessboard dataset, Polynomial kernel $(d = 10, C = 1)$



Effect of kernel matrix

- Picture from Machine Learning 10-315, Aarti Singh, Oct 28, 2020,CMU

Chessboard dataset, Gaussian RBF kernel $(C = 1, \sigma = 2)$



Quality of kernel-based SVM

- Two major factors:

1. Like LSSVM, the parameter C plays a role.

2. The choice of an appropriate kernel matrix
(and its parameters) is important.

Question: How to choose/design right ones?
- theoretical analysis?
- computational experiments !

Ideas of choosing parameters

- Example: choosing parameter C

1. Define an error or score measure:

   for example, MSE (mean squares error),

   MAPE (mean absolute percentage error),

   $1/\|\boldsymbol{w}\|_2^2$, or $\sum_{i=1}^{N} y_i\left(\boldsymbol{w}^T\boldsymbol{x}^i + b\right)$, …

2. Conduct computational experiments with different value of C :

   - statistically meaningful

3. Plot resulting error measures against C.
4. Find the elbow/ turning point value of C.

** check many other "cross-validation" methods.

Design kernel matrices

Combining kernels:

Kernels $K_1(\boldsymbol{x}^i, \boldsymbol{x}^j), K_2(\boldsymbol{x}^i, \boldsymbol{x}^j), \ldots, K_p(\boldsymbol{x}^i, \boldsymbol{x}^j)$ are given,

1. $K(\boldsymbol{x}^i, \boldsymbol{x}^j) \triangleq K_1(\boldsymbol{x}^i, \boldsymbol{x}^j) + K_2(\boldsymbol{x}^i, \boldsymbol{x}^j)$
   is a kernel matrix
2. $K(\boldsymbol{x}^i, \boldsymbol{x}^j) \triangleq \beta K_1(\boldsymbol{x}^i, \boldsymbol{x}^j), \beta > 0$
   is a kernel matrix
3. Search for the best kernel combination
   $K \triangleq \alpha_1 K_1 + \cdots + \alpha_p K_p$
   for $some\ \alpha_1 > 0, \ldots, \alpha_p > 0.$
4. When $K_1 K_2 = K_2 K_1$ (Commuting)
   $K(\boldsymbol{x}^i, \boldsymbol{x}^j) \triangleq K_1(\boldsymbol{x}^i, \boldsymbol{x}^j) K_2(\boldsymbol{x}^i, \boldsymbol{x}^j)$ is a kernel matrix

Kernel algebra

1. If $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = (\boldsymbol{x}^i)^T A \boldsymbol{x}^j$ with matrix A being symmetric and positive semidefinite,
   then K is a kernel matrix and $\phi(\boldsymbol{x}) = L\boldsymbol{x}$, where $A = LL^T$.

2. If $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = f(\boldsymbol{x}^i) f(\boldsymbol{x}^j) K_1(\boldsymbol{x}^i, \boldsymbol{x}^j)$ with
   function $f(\boldsymbol{x})\colon \mathbb{R}^n \rightarrow \mathbb{R}$ and $K_1(\boldsymbol{x}^i, \boldsymbol{x}^j) = \phi_1(\boldsymbol{x}^i)^T \phi_1(\boldsymbol{x}^j)$,
   then K is a kernel matrix and $\phi(\boldsymbol{x}) = f(\boldsymbol{x}) \phi_1(\boldsymbol{x})$.

3. If $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = \alpha K_1(\boldsymbol{x}^i, \boldsymbol{x}^j)$ with scalar $\alpha > 0$,
   and $K_1(\boldsymbol{x}^i, \boldsymbol{x}^j) = \phi_1(\boldsymbol{x}^i)^T \phi_1(\boldsymbol{x}^j)$,
   then $\phi(\boldsymbol{x}) = \sqrt{\alpha}\, \phi_1(\boldsymbol{x})$.

Kernel algebra

4. If $K(\boldsymbol{x}^i, \boldsymbol{x}^j) = K_1(\boldsymbol{x}^i, \boldsymbol{x}^j) + K_2(\boldsymbol{x}^i, \boldsymbol{x}^j)$,

and $K_1(\boldsymbol{x}^i, \boldsymbol{x}^j) = \phi_1(\boldsymbol{x}^i)^T \phi_1(\boldsymbol{x}^j)$,

$K_2(\boldsymbol{x}^i, \boldsymbol{x}^j) = \phi_2(\boldsymbol{x}^i)^T \phi_2(\boldsymbol{x}^j)$,

then $\phi(\boldsymbol{x}) = \begin{pmatrix} \phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x}) \end{pmatrix}$.

Example: For $\boldsymbol{x} \in \mathbb{R}^2$,

$K_1(\boldsymbol{x}^i, \boldsymbol{x}^j) = ((\boldsymbol{x}^i)^T \boldsymbol{x}^j + 1)^1$ with $\phi_1(\boldsymbol{x}) = (1, x_1, x_2)^{\mathrm{T}}$

$K_2(\boldsymbol{x}^i, \boldsymbol{x}^j) = ((\boldsymbol{x}^i)^T \boldsymbol{x}^j + 0)^2$ with $\phi_2(\boldsymbol{x}) = (x_1^2, 2x_1x_2, x_2^2)^T$

we have $\phi(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, 2x_1x_2, x_2^2)^T$ and

$K(\boldsymbol{x}^i, \boldsymbol{x}^j) = ((\boldsymbol{x}^i)^T \boldsymbol{x}^j + 1)^2$.

Kernel matrix and feature map

- Feature map $\Rightarrow$ kernel matrix is clear.
- How about the other direction?

Recall that we mentioned the Mercer's theorem previously. Here is how the theorem goes. Let T_K be a linear operator such that for $f \in L_2(\mathcal{X})$, $T_K(f)(x) = \int K(x,y)f(y)dy$.

Theorem 6.2 (Mercer's theorem) *Assume that K is a continuous symmetric positive semi-definite kernel over $\mathcal{X} \times \mathcal{X}$, where $\mathcal{X}$ is compact. Then there exists an orthonormal basis $\{e_i(\cdot) : i = 1, \cdots,\}$ of $L_2(\mathcal{X})$ consisting of eigenfunctions of T_K such that*

$$K(x,y) = \sum_{i=1}^{n} \lambda_i e_i(x) e_i(y),$$

where $\lambda_i \geq 0$ are the corresponding eigenvalues.

It also implies another representation under the regular L_2 space:

$$\phi(x) = (\sqrt{\lambda_1} e_1(x), \sqrt{\lambda_2} e_2(x), \cdots).$$

The quantities λ_i and $e_i(x)$ are from Theorem 6.2.

- From: http://faculty.washington.edu/yenchic/19A_stat535/Lec6_kernel.pdf

Kernel tricks of using DKSSVM

- SVM classifier

$$class_{SVM}(\boldsymbol{x}) = sign(f(\boldsymbol{x}))$$

Dual version DKSSVM

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i \, \phi(\boldsymbol{x}^i)^T \phi(\boldsymbol{x}) + b(\alpha_i)$$
$$= \sum_{i \in S} \alpha_i y_i \, K(\boldsymbol{x}^i, \boldsymbol{x}) + \bar{b}$$

- Classifier can be learnt in the higher dimensional feature space without explicitly computing $\phi(\boldsymbol{x})$.
- All that is needed is the kernel $K(\boldsymbol{x}^i, \boldsymbol{x}^j)$.
- Complexity of learning depends on N, not on l.

Comparisons and discussions

- LSSVM vs. KSSVM
  - applicability
  - complexity

- Properties of each commonly used kernel
  - polynomial
  - Gaussian
  - RBF
  - sigmoid

- Drawbacks of kernel-based SVM models

Kernel-free Nonlinear SVM

- Drawbacks of kernel-based SVM models:
  - No universal rule to select a suitable kernel function.
  - Performance depends heavily on kernel parameters.
  - Singularity of kernel matrix may cause computational problems.
- Idea: How about *generating a nonlinear separation surface* directly without using kernel functions?

Soft Quadratic Surface SVM

- Joint work with Dr. Jian Luo （海南大学罗健老师 2014 ）

- Separate by a quadratic surface: $\{f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathbf{W}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c = 0\}$.
- Adopt the relative geometrical margin based on all data points.

Kernel-free SQSSVM model [5]:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{N} \|\mathbf{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}\|_2^2 + C\sum_{i=1}^{N}\xi_i \\
s.t. \quad & y^{(i)}\left(\frac{1}{2}\boldsymbol{x}^{(i)T}\mathbf{W}\boldsymbol{x}^{(i)} + \boldsymbol{x}^{(i)T}\boldsymbol{b} + c\right) \geqslant 1 - \xi_i \\
& i = 1, \cdots, N, \\
& \mathbf{W} \in \mathbb{S}^n,\ \boldsymbol{b} \in \mathbb{R}^n,\ c \in \mathbb{R},\ \boldsymbol{\xi} \in \mathbb{R}_+^N.
\end{aligned}
\qquad \text{(SQSSVM)}
$$

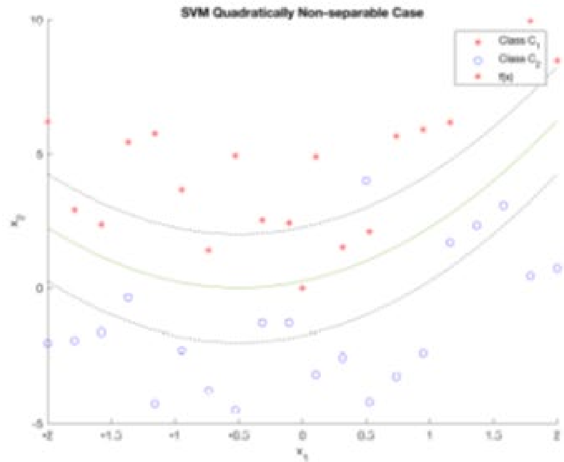


Figure: SQSSVM

where $C > 0$ is the penalty parameter for data points.

Double Well Potential Surface SVM

- Joint work with D. Zheming Gao (东大高哲明老师 2020)
- Ideal: Separate by a *degree 4 polynomial DWP surface*

$$\left\{ F(\boldsymbol{x}) = \frac{1}{2}\left(\frac{1}{2}\|\mathbf{B}\boldsymbol{x} - \boldsymbol{q}\|_2^2 - d\right)^2 + \frac{1}{2}\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x} + c = 0 \right\}.$$

$$\begin{aligned}
\min \quad & \frac{1}{2}\left\| \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \right\|_F^2 + \frac{1}{2}\|\boldsymbol{b}\|_2^2 + C\sum_{i=1}^{N}\zeta_i \\
\text{s.t.} \quad & y^{(i)}\left(\frac{1}{2}\boldsymbol{z}^{(i)T}\mathbf{W}\boldsymbol{z}^{(i)} + \frac{1}{2}\boldsymbol{x}^{(i)T}\mathbf{A}\boldsymbol{x}^{(i)} + \boldsymbol{b}^T\boldsymbol{x}^{(i)} + c\right) \geqslant 1 - \zeta_i, \\
& i = 1, \ldots, N, \\
& \text{rank}(\mathbf{W}) = 1 \\
& \mathbf{W} \in \mathbb{S}^{\frac{n(n+1)}{2}+n+1},\ \mathbf{A} \in \mathbb{S}^n,\ \boldsymbol{b} \in \mathbb{R}^n,\ c \in \mathbb{R},\ \zeta \in \mathbb{R}_+^N.
\end{aligned}$$

(DWPSVM)

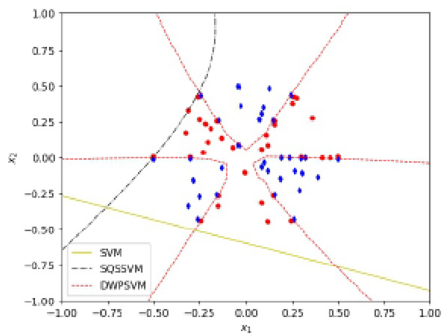


Figure: DWPSVM