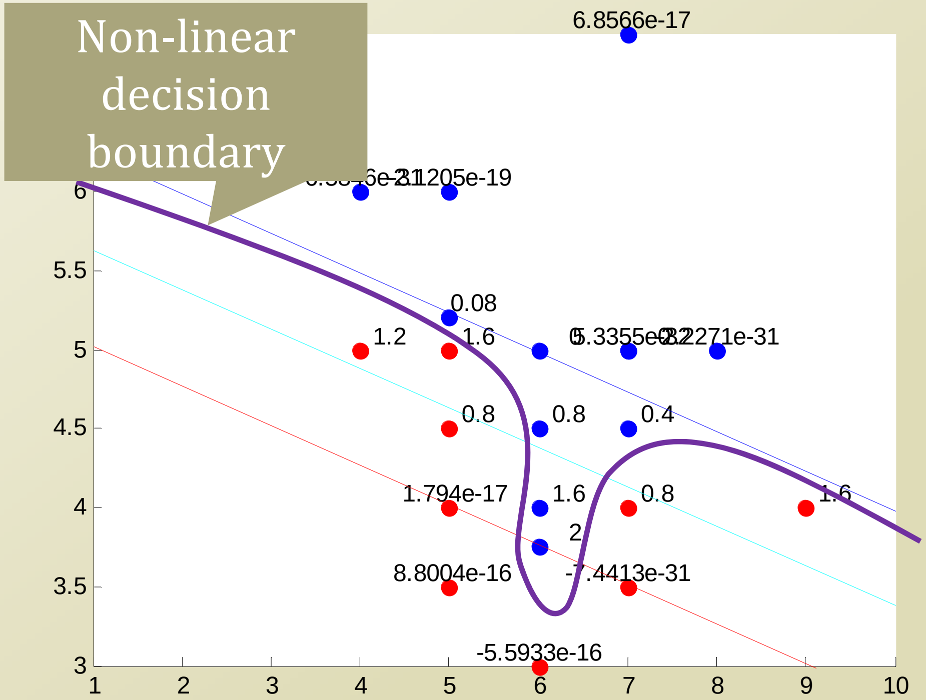
Support Vector Machine

Il-Chul Moon
Dept. of Industrial and Systems Engineering
KAIST
icmoon@kaist.ac.kr

KERNEL TRICK

Enough of Studying SVM?

- You can train the SVM when you even have "error" cases
  - Use a soft-margin to handle such errors
- However, this does not change the complexity of the decision boundary
- In the real world, there are situations which require complex decision boundary…
  - Option 1
    - Make decision boundary more complex
    - Go to non-linear
  - Option 2
    - Admit there will be an "error"
    - Represent the error in our problem formulation.



KAIST

Feature Mapping to Expand Dim.

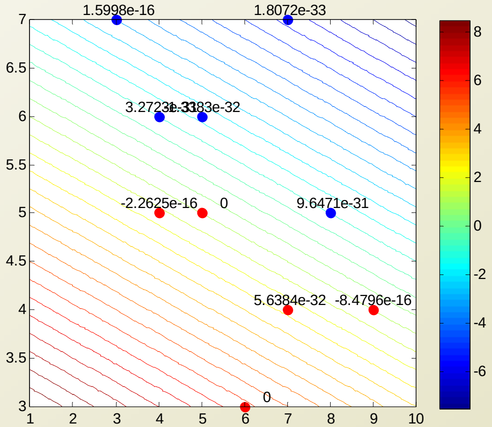
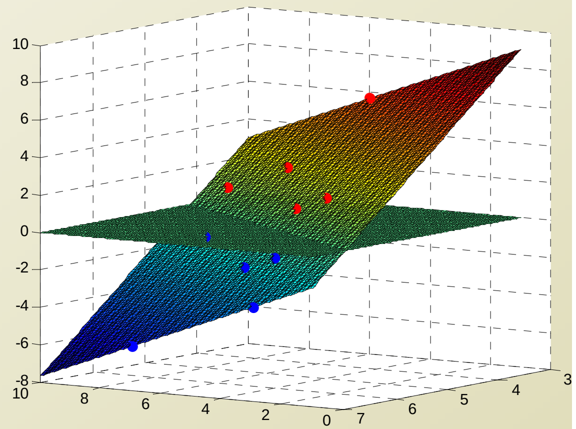
- $min_{w,b,\xi_j} ||w|| + C \sum_j \xi_j$

  $s.t.$

  $(w\varphi(x_j) + b)y_j \geq 1 - \xi_j, \forall j$

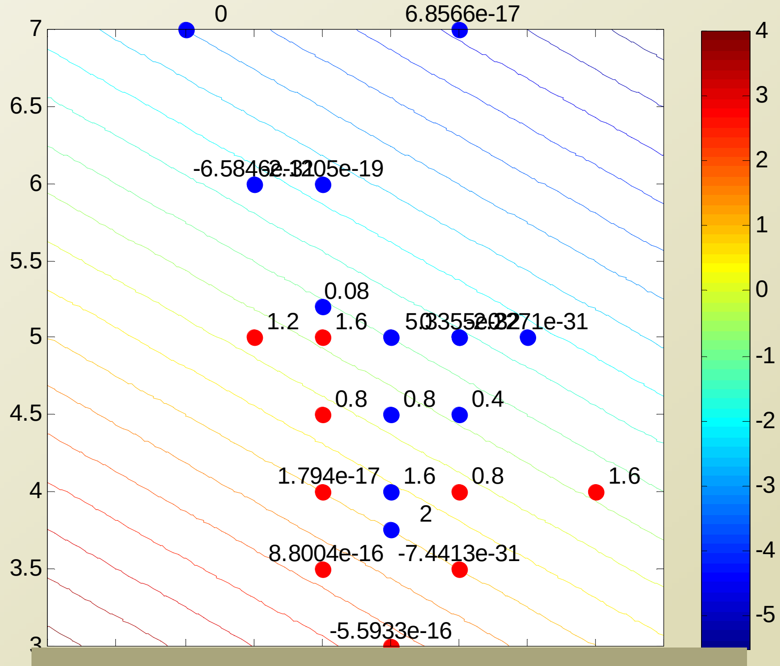
  $\xi_j \geq 0, \forall j$

- $\varphi(< x_1, x_2 >) =$
  $< x_1, x_2, x_1{}^2, x_2{}^2, x_1x_2, x_1{}^3, x_2{}^3, x_1{}^2x_2, x_1x_2{}^2 >$

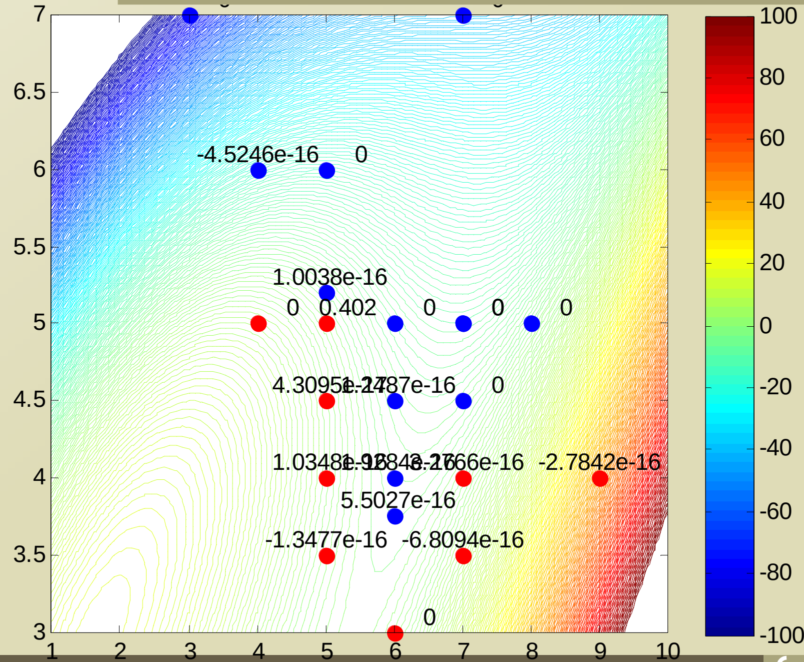



Linearly Separable Dataset

Any problem???
# of Params, Representation, Computation….



Linear Decision Boundary on



Rethinking the Formulation

- SVM turns
  - Classification → Constrained quadratic programming
- Constrained optimization
  - $min_x f(x)$
  - $s.t.\ g(x) \leq 0, h(x) = 0$
- Lagrange method
  - Lagrange Prime Function: $L(x, \alpha, \beta) = f(x) + \alpha g(x) + \beta h(x)$
  - Lagrange Multiplier: $\alpha \geq 0, \beta$
  - Lagrange Dual Function: $d(\alpha, \beta) = inf_{x \in X} L(x, \alpha, \beta) = min_x L(x, \alpha, \beta)$
  - $max_{\alpha \geq 0, \beta} L(x, \alpha, \beta) = \begin{cases} f(x): if\ x\ is\ feasible \\ \infty: otherwise \end{cases}$
  - $min_x f(x) \rightarrow min_x max_{\alpha \geq 0, \beta} L(x, \alpha, \beta)$
- Take advantage of the formulation technique of the constrained optimization
  - Primal and Dual Problems!

inf: infimum "Greatest Lower Bound"
inf{1,2,3} = 1