Support Vector Machine

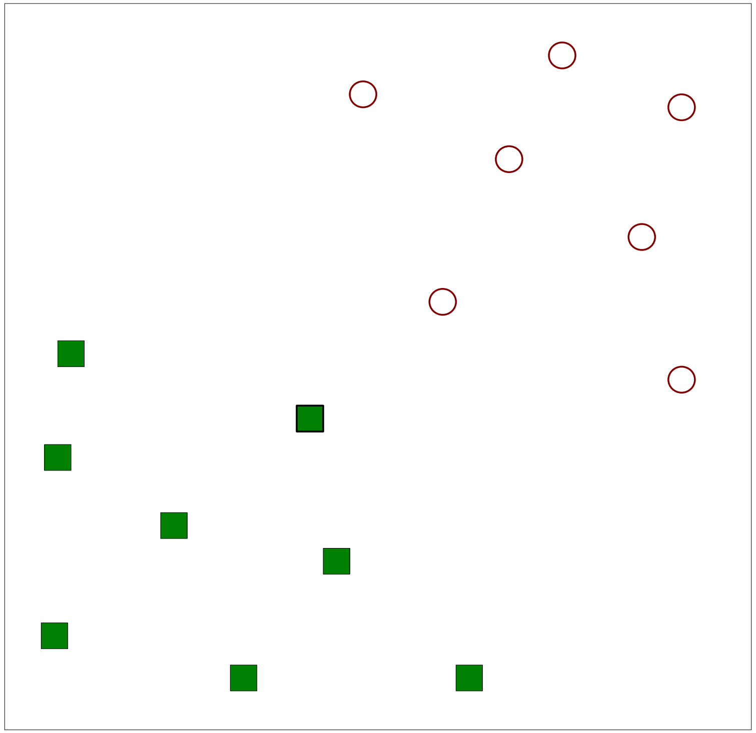
Saerom Park
Department of Industrial Engineering
srompark@unist.ac.kr

Support Vector Machine

- (Separable) Linear SVM
- (Non-separable) Soft-margin SVM

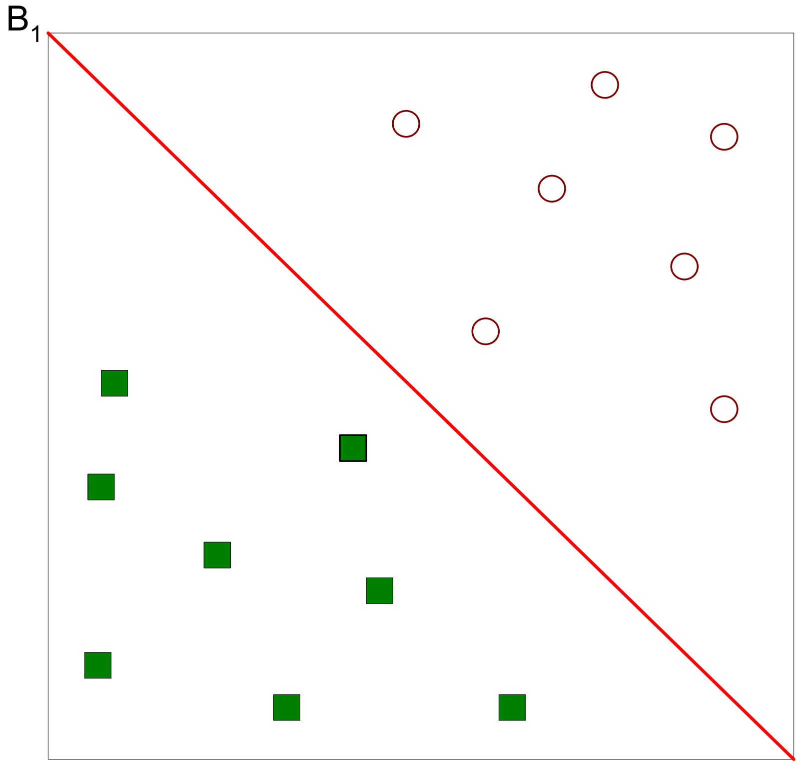
Data Mining
Prof. Saerom Park

Support Vector Machines



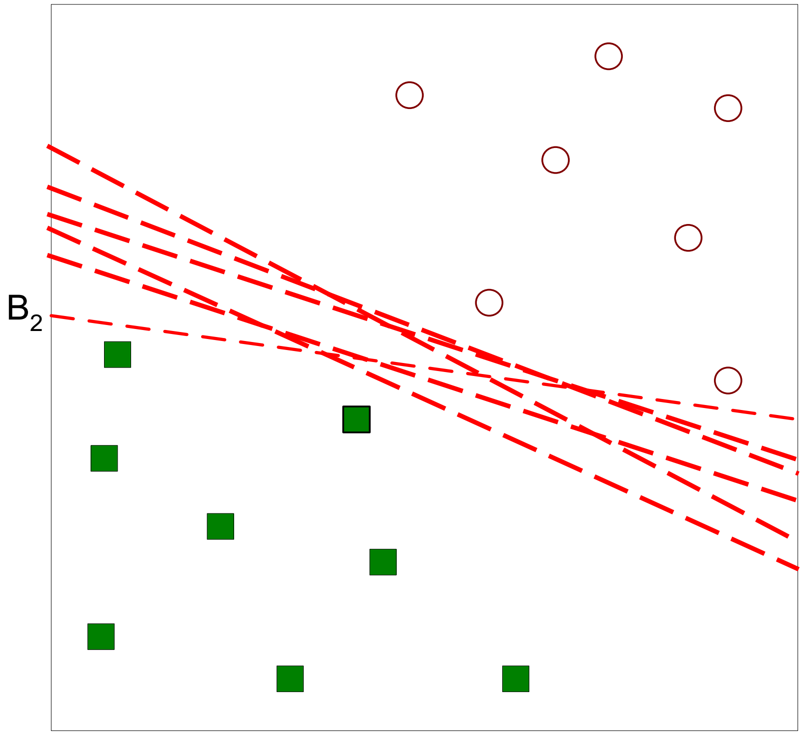
Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



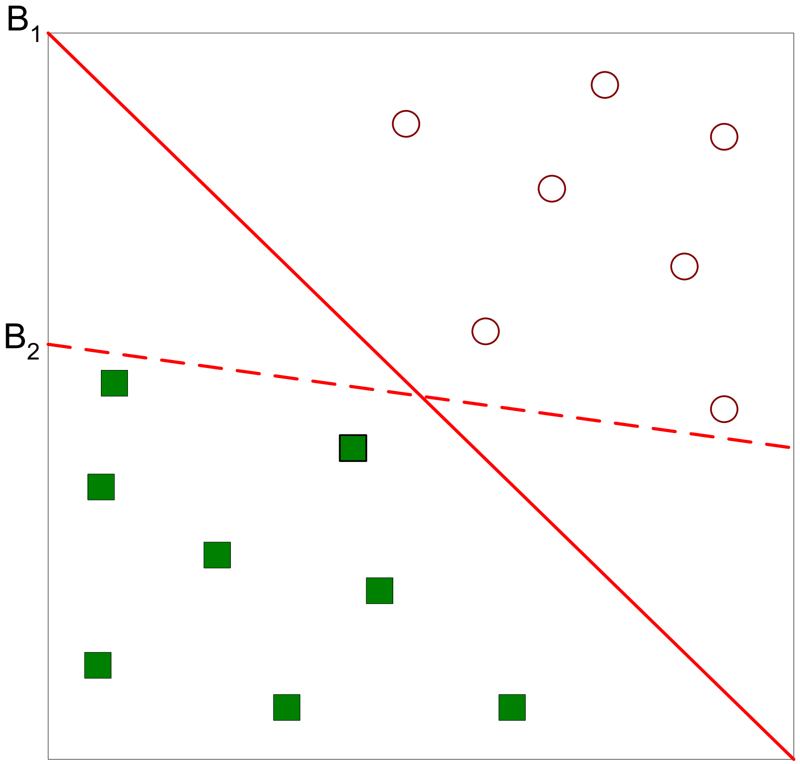
One Possible Solution

Support Vector Machines



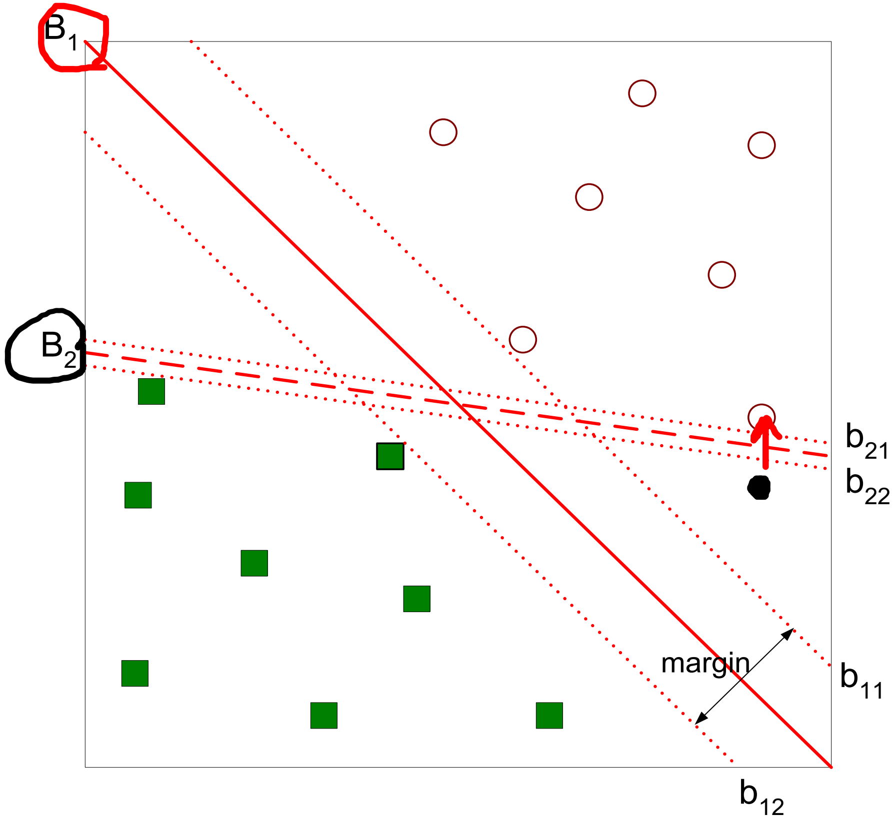
Other possible solutions

Support Vector Machines



Which one is better? B1 or B2?
How do you define better?

Support Vector Machines

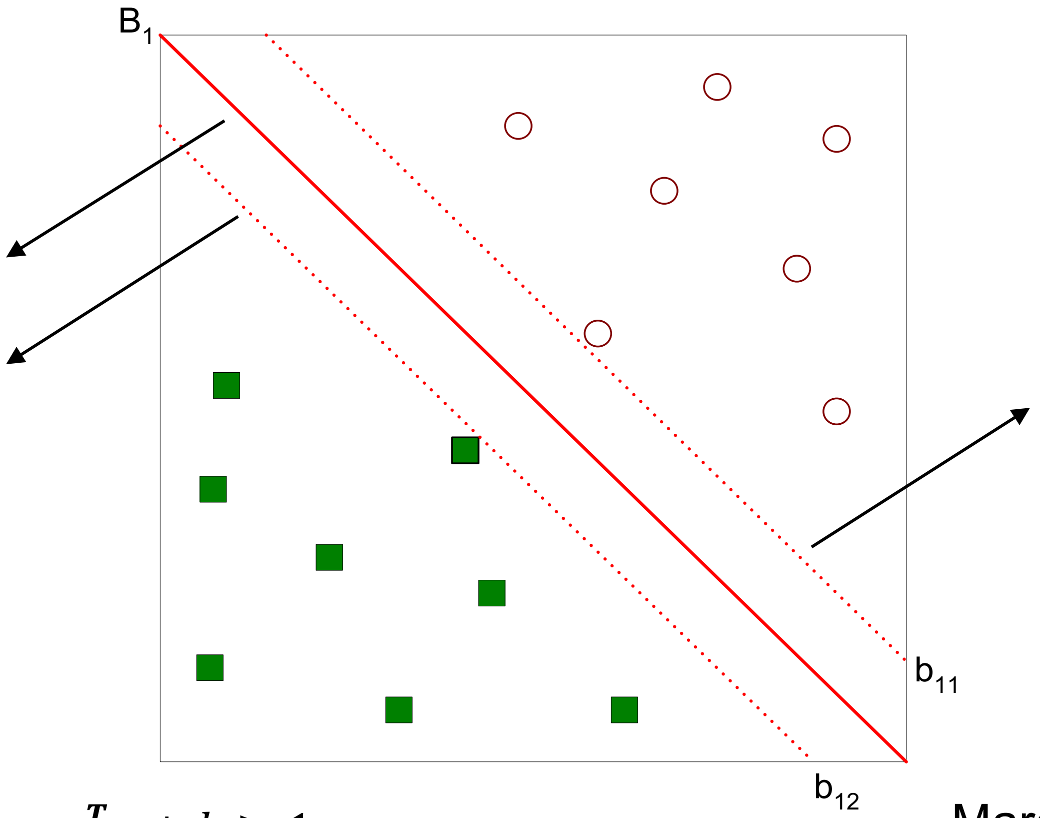


Find hyperplane maximizes the margin => B1 is better than B2

Support Vector Machines

$$\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b = 0$$

$$\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b = -1$$



$$\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b = +1$$

Classifier $c(\boldsymbol{x}) = \begin{cases} 1. & \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b \geq 1 \\ -1. & \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + b \leq -1 \end{cases}$

Margin = $\dfrac{2}{||\boldsymbol{w}||}$

Linear SVM

- Learning the model is equivalent to determining the values of $\boldsymbol{w}$ and b
  - How to find $\boldsymbol{w}$ and b from training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \subset \mathbb{R}^d \times \{-1,1\}$?
- Objective is to maximize: Margin= $\frac{2}{\|\boldsymbol{w}\|}$
  - Which is equivalent to minimizing: $L(\boldsymbol{w}) = \frac{\|w\|^2}{2}$
  - Constraints: $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$ for $i = 1, \dots, n$
  - This is a ***constrained optimization problem:*** Quadratic objective function and linear constraints → Quadratic Programming (QP) → Lagrangian multipliers

$$Minimize_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$$
$$subject\ to\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1\ , \forall i$$

Lagrange Multiplier Method

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{n} \alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\}$$

where $\alpha_i \geq 0$ for all $i = 1, \cdots, n$

Linear SVM

Lagrange Multiplier Method

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{n} \alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\}$$

where $\alpha_i \geq 0$ for all $i = 1, \cdots, n$

- $\frac{\partial L}{\partial \boldsymbol{w}} = 0, \frac{\partial L}{\partial b} = 0 \rightarrow \boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}_i, \sum_i \alpha_i y_i = 0$

- Dual problem (quadratic programming)

Lagrangian dual function:
$L(\alpha) = \min_{\boldsymbol{w},b} L(w, b, \alpha)$

$$\max L(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j + \sum_{i=1}^{n} \alpha_i$$

$$subject\ to.\ \sum_i \alpha_i y_i = 0, \qquad \alpha_i \geq 0, \qquad \forall i$$

  - Convex quadratic optimization → Global optimum is guaranteed

- Use KKT (Karush-Kuhn-Tucker) conditions (for optimal solution)

1. Stationary: $\frac{\partial L}{\partial \boldsymbol{w}} = 0, \frac{\partial L}{\partial b} = 0$
2. Primal feasibility: $y_i(w^T\boldsymbol{x}_i + b) \geq 1, \forall i$
3. Dual feasibility: $\alpha_i \geq 0, \forall i$
4. **Complementary slackness**

Linear SVM

- Use KKT (Karush-Kuhn-Tucker) conditions (for optimal solution)

1. Stationary: $\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial b} = 0$
2. Primal feasibility: $y_i(w^T \boldsymbol{x}_i + b) \geq 1, \forall i$
3. Dual feasibility: $\alpha_i \geq 0, \forall i$
4. **Complementary slackness**

- Complementary slackness

$$\alpha_i(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1) = 0$$

1) $\alpha_i > 0$ and $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) = 1 \rightarrow x_i$ lies on the margin boundary (+1 or -1) **Support Vectors (SVs)**

2) $\alpha_i = 0$ and $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) > 1 \rightarrow x_i$ lies outside the margin boundary

The decision function of SVM (primal solution)

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

The decision function of SVM (dual solution)

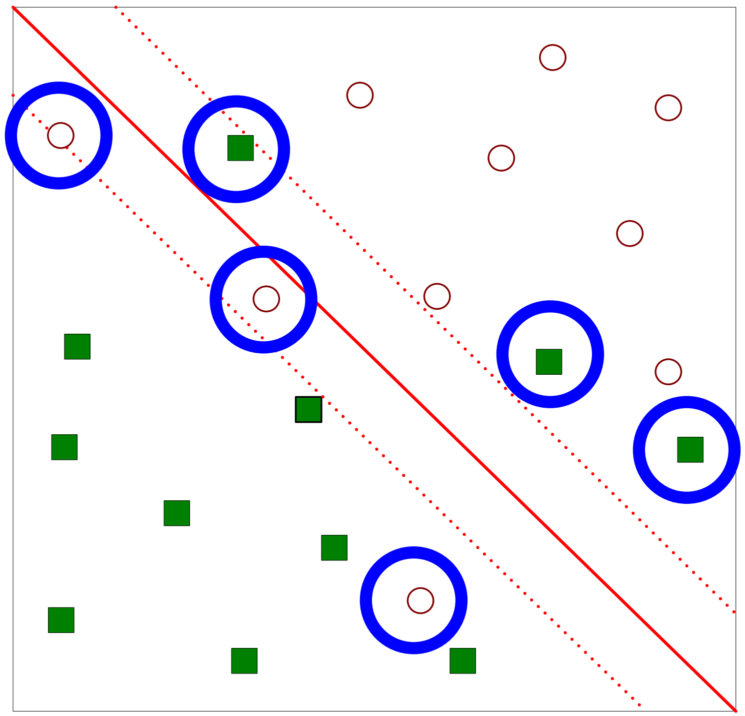
$$f(\boldsymbol{x}) = \sum_i \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x} + b^*$$

When we classify a test data x, only support vectors contribute to $\mathrm{f}(x)$

C.f. $b^* = y_i - \boldsymbol{w}^T \boldsymbol{x}_i$ *for any* x_i *(support vector) such that* $\alpha_i > 0$

Non-separable Case

- What if the problem is not linearly separable?



Feasible solution does not exist any more

Soft-margin SVM

Lagrange Multiplier Method

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1\} - \sum_{i=1}^{n}\beta_i\xi_i$$

where $\alpha_i, \beta_i \geq 0$ for all $i = 1, \cdots, n$

- Soft-margin formulation (Primal)
  - Allow some errors by introducing slack variables $\xi_i \geq 0$

$L(y, \hat{y}) = \max(0, 1 - y\hat{y})$

Maximize the margin

Minimize empirical risk (hinge loss)
trade-off hyperparameter C

$$\min L(\boldsymbol{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i}\xi_i$$

$$\textit{subject to} \quad y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \forall i$$

Most training points are outside the margin,
but some are not

how much a data point violates
the margin

  - Quadratic optimization problem
    - Quadratic objective function
    - Linear constraints

Soft-margin SVM

Lagrange Multiplier Method

$$L(\boldsymbol{w}, b, \xi, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i\} - \sum_{i=1}^{n}\beta_i\xi_i$$

where $\alpha_i, \beta_i \geq 0$ for all $i = 1, \cdots, n$

- Dual problem (quadratic programming)

$$\max L(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j$$

$$subject\ to.\ \sum_i \alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq C, \qquad \forall i$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow w = \sum_i \alpha_i y_i \boldsymbol{x}_i$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_i \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$

  - Convex optimization → Global optimum is guaranteed

- Use KKT (Karush-Kuhn-Tucker) conditions (for optimal solution)

1. Stationary: $\frac{\partial L}{\partial \boldsymbol{w}} = 0, \frac{\partial L}{\partial b} = 0, \frac{\partial L}{\partial \xi} = 0$
2. Primal feasibility: $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0$
3. Dual feasibility: $0 \leq \alpha_i \leq C$
4. **Complementary slackness**

Soft-margin SVM

The decision function of SVM (dual solution)

$$f(\boldsymbol{x}) = \sum_i \alpha_i y_i \boldsymbol{x}_i^T \boldsymbol{x} + b^*$$

c.f. $b^* = y_i - w^T x_i$ *for any* x_i *(support vector) such that* $0 < \alpha_i < C$

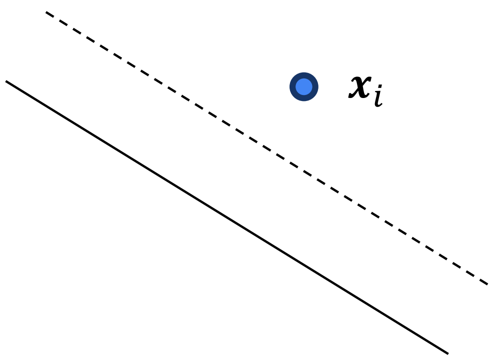
- The classifier only depends on the support vectors in training data

$$D_{SV} = \{(\boldsymbol{x}_i, y_i) \in D_{tr} | \alpha_i > 0\}$$

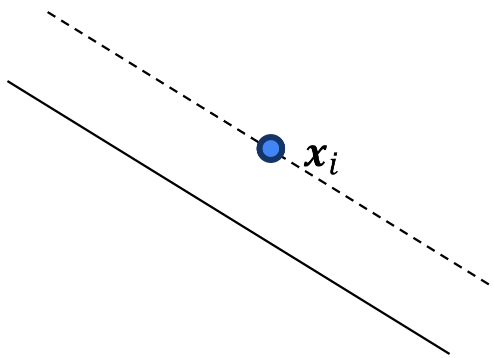
- Complementary slackness

$$\alpha_i(y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) - 1 + \xi_i) = 0, (C - \alpha_i)\xi_i = 0$$

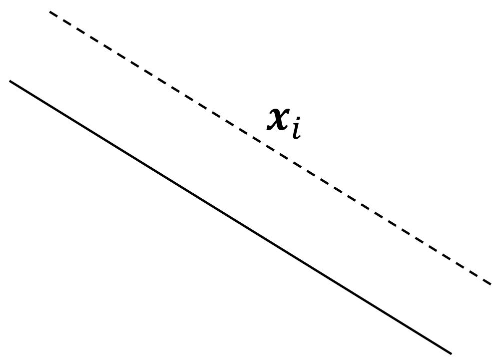
1) $\alpha_i = 0$ ($\xi_i = 0$) and $y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) > 1 \rightarrow \boldsymbol{x}_i$ lies outside the margin and is not a support vector
2) $0 < \alpha_i < C$ and $y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) = 1 \rightarrow \boldsymbol{x}_i$ lies exactly on the margin boundary **Support Vectors (SVs)**
3) $\alpha_i = C$, $\xi_i > 0$ and $\mathrm{y}_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \leq 1 - \xi_i \rightarrow \boldsymbol{x}_i$ lies inside the margin or is misclassified **Sparse solution!**



Non-support vectors
$\{(\boldsymbol{x}_i, y_i) | \alpha_i = 0\}$



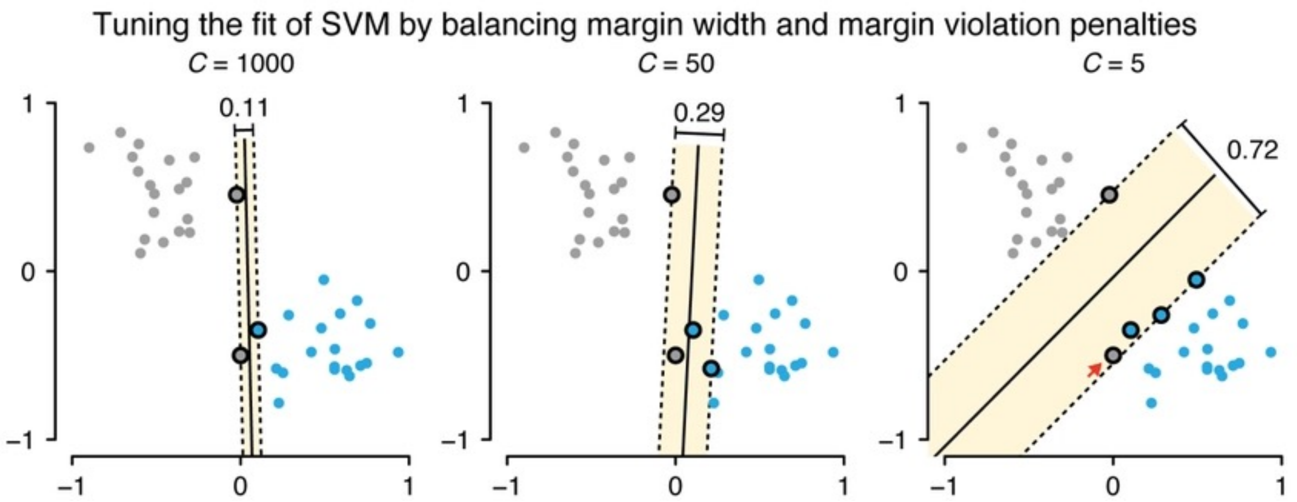
(Margin) support vectors
$\{(\boldsymbol{x}_i, y_i) | 0 < \alpha_i < C\}$



Error support vectors
$\{(\boldsymbol{x}_i, y_i) | \alpha_i = C\}$

Regularization of SVM

- The trade-off hyperparameter (the strength of the regularization): C
  - Lower values of C correspond to more regularization
    - The model puts more emphasis on finding a coefficient vector $\mathbf{w}$ that is close to zero **→ underfitting**
  - Higher values of C correspond to less regularization
    - The model tries to fit the training set as best as possible **→ overfitting**



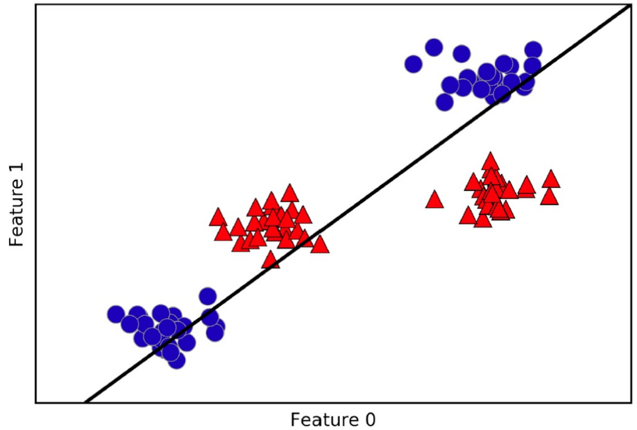
Kernel SVM

Data Mining
Prof. Saerom Park

- Kernel trick
- Kernelized SVM

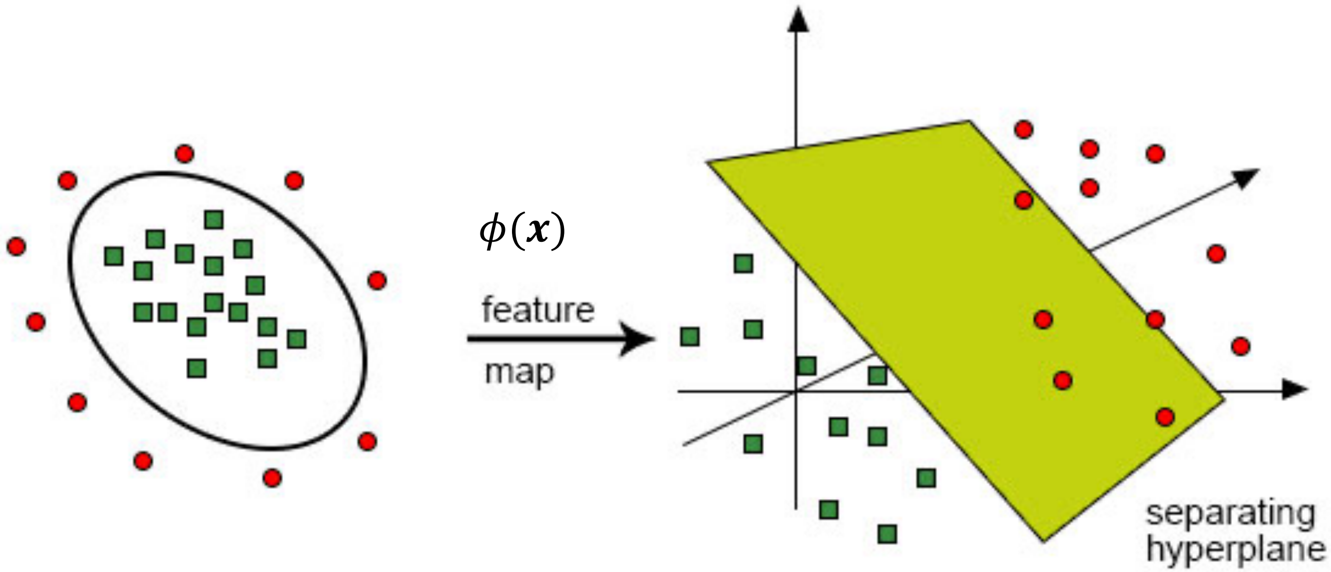
Limitations on Linear SVM

- Linear support vector classification can be quite limiting in low-dimensional spaces, as lines and hyperplanes have limited flexibility
- Kernelized support vector machines are an extension that allows for more complex models that are not defined simply by hyperplanes in the input space.
    - Example: Given a two-class classification dataset in which classes are not linearly separable, the decision boundary found by a linear SVM



Kernelized Trick

- SVM for Non-linear Classification: Kernel Trick
  - Use a function φ that maps the data into a higher dimensional space
    - Replace $\boldsymbol{x}_i$ by $\phi(\boldsymbol{x}_i)$
    - Example: $\phi(\boldsymbol{x}) = \phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$



Kernelized Trick

- Kernel function: in general, it can be considered as a similarity metric
  - If there is a "kernel function" k that defines inner products in the transformed space, such that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$, then we don't have to know ϕ at all, but use k instead.
    - Replace $\boldsymbol{x}_i^T \boldsymbol{x}_j$ by $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
    - Positive definite symmetric (PDS) kernels can preserve the convexity of optimization problem (Mercer's theorem)
- Examples of Kernel Functions
  - (PDS) Linear kernel: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$
  - (PDS) Polynomial kernel: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (a + b\boldsymbol{x}_i^T \boldsymbol{x}_j)^p$
  - (PDS) Radial basis function (RBF) kernel: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$
  - Tanh kernel: $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(a + b\boldsymbol{x}_i^T \boldsymbol{x}_j)$ (non-PSD depends on the choice of the parameters a,b)

PDS can be related to the convergence of SVM problem

Kernelized Support Vector Classification

- Positive definite symmetric (PDS) kernels
  - A kernel $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be positive definite symmetric (PDS) if for any $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathcal{X}$, the matrix $K = \left[k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right]_{ij} \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite (SPSD)
- Techniques for constructing new PDS kernels
  - Given valid PDS kernels $k_1(\boldsymbol{x}, \boldsymbol{x}')$, $k_2(\boldsymbol{x}, \boldsymbol{x}')$, the following new kernels will also be valid
    - $k(\boldsymbol{x}, \boldsymbol{x}') = c k_1(\boldsymbol{x}, \boldsymbol{x}')$ ($c > 0$), $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(k_1(\boldsymbol{x}, \boldsymbol{x}'))$
    - $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}')$, $k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') \cdot k_2(\boldsymbol{x}, \boldsymbol{x}')$
    - $k(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x}) k_1(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}')$ (any function $f(\boldsymbol{x})$),
    - $k(\boldsymbol{x}, \boldsymbol{x}') = q\left(k_1(\boldsymbol{x}, \boldsymbol{x}')\right)$ (polynomial with nonnegative coefficients $q(\boldsymbol{x})$)

Soft-margin kernel SVM

$$\max L(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

- Soft-margin formulation (Primal)
  - Allow some errors by introducing slack variables $\xi_i \geq 0$

$$\min L(\boldsymbol{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_i \xi_i$$

$$subject\ to\quad y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \forall i$$

- Dual problem (quadratic programming)
  - Allow some errors by introducing slack variables $\xi_i \geq 0$

$$\max L(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

← Use QP solver!

$$subject\ to.\quad \sum_i \alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq C, \qquad \forall i$$

Convex optimization:
Global optimum is guaranteed

PDS kernel

Kernel SVM optimal solution

- Primal solution through dual solution

$$\boldsymbol{w}^* = \sum_i \alpha_i y_i \phi(\boldsymbol{x}_i)$$

$$b^* = y_i - w^{*T}\phi(\boldsymbol{x}_i) = \mathrm{y_i} - \sum_{j=1}^{n} \alpha_j y_j k(\boldsymbol{x}_j, \boldsymbol{x}_i) \text{ where } 0 < \alpha_i < C$$

- For a new test data
  - SVM classifier

$$c(\boldsymbol{x}) = sign(\boldsymbol{w}^{*T}\phi(\boldsymbol{x}) + b^*) = sign\left(\sum_i \alpha_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b^*\right)$$

Sparse solution!

  - The classifier only depends on the support vectors in training data

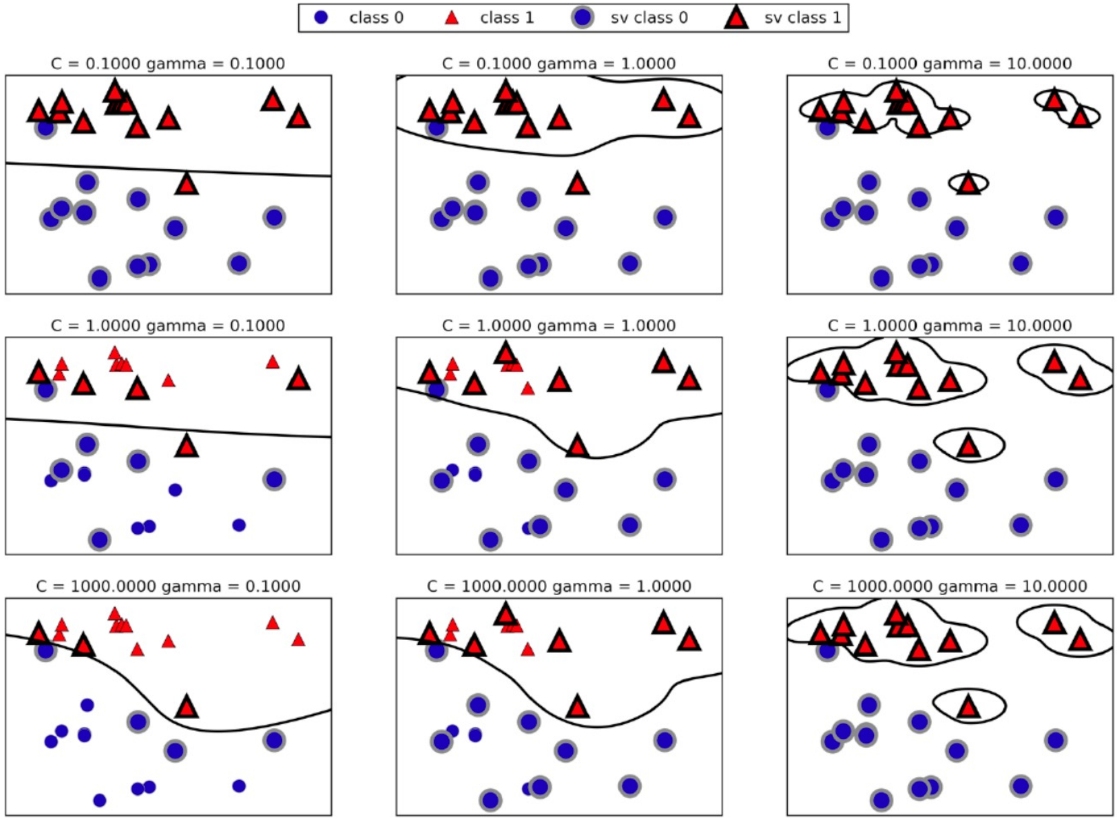
$$D_{SV} = \{(\boldsymbol{x}_i, y_i) \in D_{tr} | \alpha_i > 0\}$$

Hyperparameters for Kernel SVM

- RBF kernel
  - Kernel parameter γ

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$$

  - Smaller parameter means that the exponential term will decay rapidly, resulting in the influence of each data point being more *localized*
- Regularization
  - Parameter C



Appendix

- Optimization
- KKT conditions

Optimization*

- Constrained Optimization problem with one inequality
  - Suppose that x^* is a local solution of constrained optimization problem

$$minimize\ f(\boldsymbol{x})$$
$$subject\ to\ g(x) \leq 0$$
$$\mathcal{L}(x, \mu) = f(x) + \mu g(x), \mu \geq 0$$

  - If the solution lies at the constraint boundary, then the Lagrange condition holds as

$$\nabla_x \mathcal{L}(x^*, \mu^*) = \nabla_x f(x^*) + \mu^* \nabla_x g(x^*) = 0$$
$$\frac{\partial \mathcal{L}(x^*, \mu^*)}{\partial \mu} = g(x^*) = 0$$

Optimal point is g(x) =0
$\mu > 0$

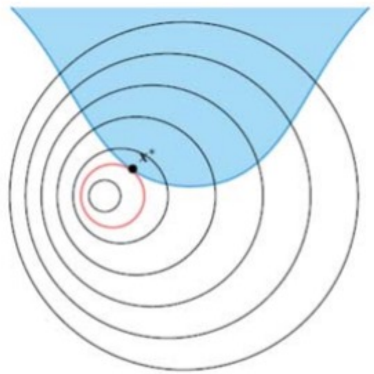


Figure 10.5. An active inequality constraint. The corresponding contour line is shown in red.

Optimization*

- Constrained Optimization problem with one inequality
  - Suppose that x^* is a local solution of constrained optimization problem

$$minimize\ f(\boldsymbol{x})$$
$$subject\ to\ g(x) \leq 0$$
$$\mathcal{L}(x, \mu) = f(x) + \mu g(x), \mu \geq 0$$

  - If the solution is inside the at the constraint boundary, then the Lagrange condition holds as

$$\nabla_x \mathcal{L}(x^*, \mu^*) = \nabla_x f(x^*) + \mu^* \nabla_x g(x^*) = 0$$

$$\frac{\partial \mathcal{L}(x^*, \mu)}{\partial \mu} = g(x^*) < 0$$

g(x) has no meaning
$\mu = 0$

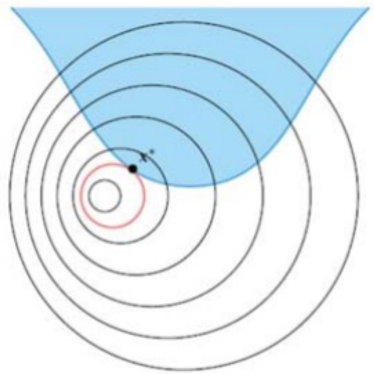


Figure 10.5. An active inequality constraint. The corresponding contour line is shown in red.

Optimization*

- Constrained Optimization problem with one inequality
  - Suppose that x^* is a local solution of constrained optimization problem

$$\mathit{minimize}\ f(\boldsymbol{x})$$
$$\mathit{subject\ to}\ g(x) \le 0$$
$$\mathcal{L}(x, \mu) = f(x) + \mu g(x), \mu \ge 0$$

  - The Lagrange condition holds as

$$\nabla_x \mathcal{L}(x^*, \mu^*) = \nabla_x f(x^*) + \mu^* \nabla_x g(x^*) = 0$$
$$\mu g(x^*) = 0$$
$$\mu \ge 0$$

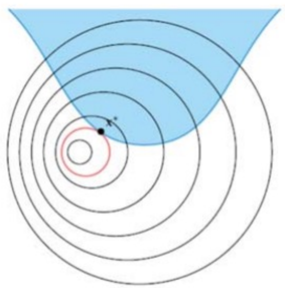


Figure 10.5. An active inequality constraint. The corresponding contour line is shown in red.

g(x) =0
$\mu > 0$

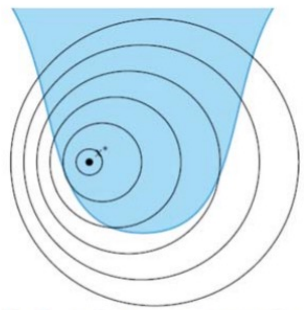


Figure 10.6. An inactive inequality constraint.

g(x) <0
$\mu = 0$

KKT conditions*

- Constrained optimization problem

$$\text{Minimize } f(x)$$
$$\text{Subject to } g_i(x) \leq 0\,, i = 1, .., l$$
$$h_j(x) = 0 \;\;, j = 1, \dots, m$$

  - where f is the objective function, g are the inequality constraints, and h are the equality constraints.
  - Feasible set $\Omega = \{x\colon g_i(x) \leq 0\,, i = 1, \dots, l,\;\; h_j(x) = 0\,, j = 1, \dots, m\}$
  - At a feasible point $x \in \Omega$, the inequality constraint i is said to be active if $g_i(x) = 0$ and inactive if the strict inequality $g_i(x) > 0$ is satisfied.
- Lagrangian for the constrained optimization problem

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_i u_i g_i(x) + \sum_j \lambda_j\, h_j(x)$$

KKT conditions*

- Karush-Kuhn-Tucker conditions
  - Suppose that x^* is a local solution of constrained optimization problem (1) and the LICQ (Linearly Independence Constraint Qualification) holds at x^* .
  - Then there is a Lagrange multiplier vector (λ^*, μ^*) such that the following Karush-Kuhn-Tucker conditions, or KKT conditions for short are satisfied at x^*, λ^*, ν^* :

$$\nabla \mathcal{L}(x^*, \lambda^*, \mu^*) = \nabla f(x^*) + \sum_i u_i^* \nabla g_i(x^*) + \sum_j \lambda_j^* \nabla h_j(x^*) = 0 \text{ (Stationarity)}$$

$$g_i(x^*) \leq 0 , i = 1, \dots , l \text{ (Primal Feasibility)}$$

$$h(x^*) = 0 , j = 1, \dots , m$$

$$\mu_i^* \geq 0 , i = 1, \dots , l \text{ (Dual Feasibility)}$$

$$\mu_i^* g_i(x^*) = 0 = 0 , i = 1, \dots , l \text{ (Complementary slackness)}$$

What's Next?

Data Mining
Prof. Saerom Park

- Principal Component Analysis
- Manifold Learning