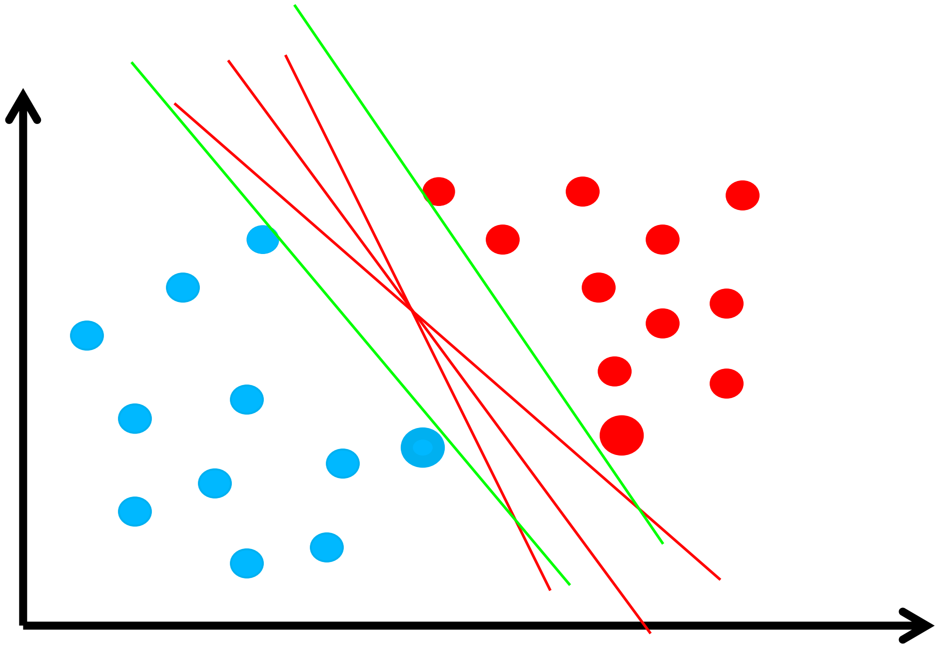
AARHUS
UNIVERSITET

Optimization
and
Data Analytics

Alexandros Iosifidis
@
Aarhus University, Department of Engineering

Support Vector Machine



Infinite decision functions possible

Support Vector Machine

That is, given a set of N samples, each represented by a vector $\mathbf{x}_i \in \mathbb{R}^D$, and the corresponding labels $l_i = \{-1,1\}$ we want to optimize the parameters of $g(\cdot)$ in order to define a discriminant hyperplane discriminating the two classes.

Support Vector Machine (SVM) assumes the data $\mathbf{x}$ is mapped to $\boldsymbol{\varphi}$ using a function $\varphi(\cdot)$

$$\mathbf{x}_i \in \mathbb{R}^D \xrightarrow[\phi(\cdot)]{} \phi_i \in \mathcal{F}$$

Note that the above is a generic mapping. For example a linear function $\varphi(\mathbf{x}) = \mathbf{x}$ can also be used.

Then, we define the decision function $$g(\phi_i) = \mathbf{w}^T \phi_i - b$$

Support Vector Machine

If the parameters of the decision function are optimized, then

$$l_i g(\phi_i) \geq 0 \Rightarrow l_i(\mathbf{w}^T \phi_i - b) \geq 0$$

or

$$\begin{aligned} \mathbf{w}^T \phi_i - b &\geq q, \text{for } l_i = 1 \text{ and} \\ \mathbf{w}^T \phi_i - b &\leq -q, \text{for } l_i = -1. \end{aligned}$$

q expresses the minimal distance between the decision hyperplane and the closest to it training samples. That is, q is the margin appearing between the two classes.

Notice that there are multiple w's (see next slide).
But we only care about when w is the unit vector

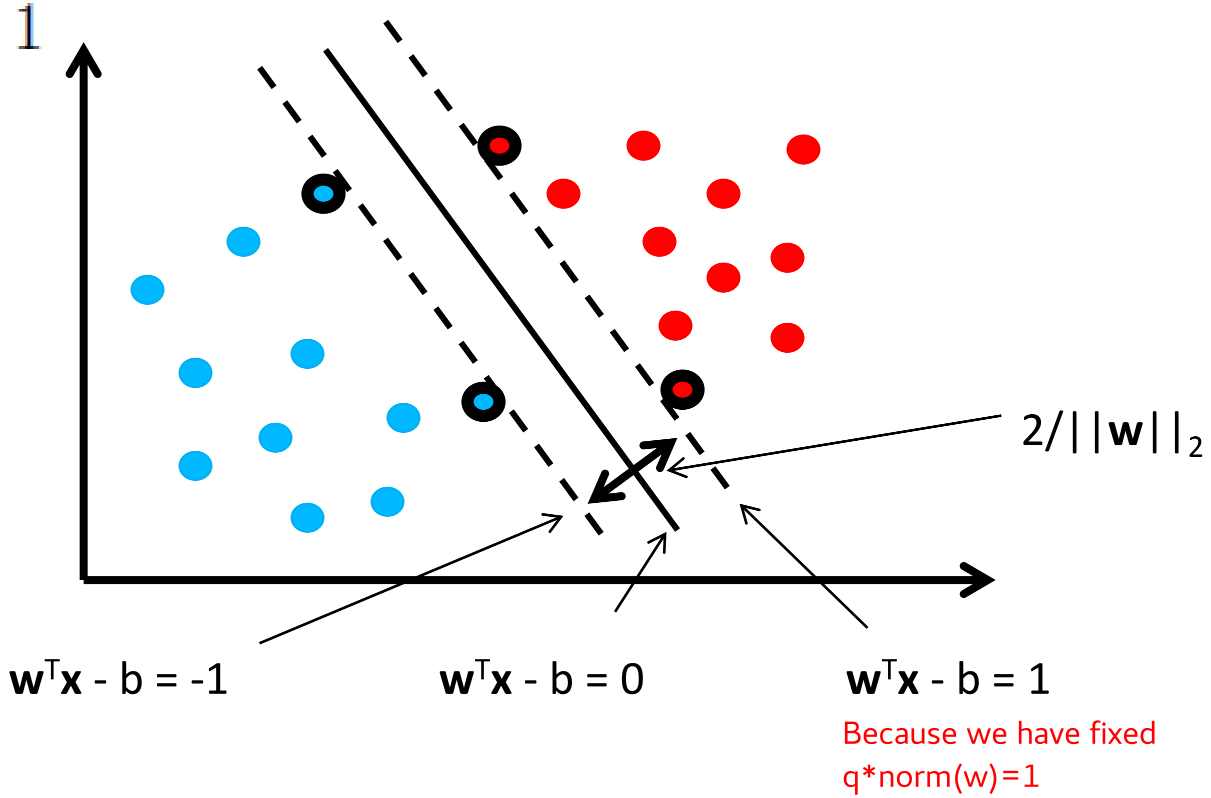
Remember that the decision function expresses distance of a sample from the hyper-plane.

Thus for $\boldsymbol{\varphi}_m$ (the closest point) $$\frac{l_m g(\phi_m)}{\|\mathbf{w}\|_2} = q$$

Support Vector Machine

We fix the right side of the equation in the preivous slide

We use: $q\|\mathbf{w}\|_2 = 1$



Support Vector Machine

In order to define the weights w and the margin b, SVM optimizes for

Minimise the inverse of w which maximises the margin

$$\mathcal{J}_{SVM} = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i,$$

subject to the constraints:

$$\begin{aligned} l_i(\mathbf{w}^T\phi_i - b) &\geq 1 - \xi_i,\ i = 1, \ldots, N \\ \xi_i &\geq 0. \end{aligned}$$

When the Ci is zero
then the corresponding x
in the margin.

Between 0 to 1, it is on the
correct side of the decision
function but inside the margin.

If Ci > 1 then the sample will
be on the wrong side of the
decision function

Support Vector Machine

To optimize J_{SVM} s.t. the constraints, we define the Lagrangian

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\beta_i\xi_i - \sum_{i=1}^{N}\alpha_i[l_i(\mathbf{w}^T\phi_i - b) - 1 + \xi_i]$$

It is possible to use a fixed alpha and beta instead of alphas and betas for each sample.

Support Vector Machine

To optimize J_{SVM} s.t. the constraints, we define the Lagrangian

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^T\mathbf{w} + c\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\beta_i\xi_i - \sum_{i=1}^{N}\alpha_i[l_i(\mathbf{w}^T\phi_i - b) - 1 + \xi_i]$$

The derivatives of L w.r.t. all variables are

$$\frac{\theta\mathcal{L}}{\theta\mathbf{w}} = 0 \;\; \Rightarrow \;\; \mathbf{w} = \sum_{i=1}^{N}\alpha_i l_i\phi_i,$$

$$\frac{\theta\mathcal{L}}{\theta b} = 0 \;\; \Rightarrow \;\; \sum_{i=1}^{N}\alpha_i l_i = 0,$$

$$\frac{\theta\mathcal{L}}{\theta\xi_i} = 0 \;\; \Rightarrow \;\; c - \alpha_i - \beta_i = 0.$$

Support Vector Machine

If we substitute these equations to L, we obtain

We compute the results and get a quadratic function

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j l_i l_j \phi_i^T \phi_j + \sum_{i=1}^{N} \alpha_i$$

subject to the constraints

$$0 \leq \alpha_i \leq c,\ i = 1, \ldots, N$$

Support Vector Machine

If we substitute these equations to L, we obtain

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j l_i l_j \phi_i^T \phi_j + \sum_{i=1}^{N} \alpha_i$$

subject to the constraints

$$0 \leq \alpha_i \leq c,\ i = 1, \ldots, N$$

Which can also be written as

$$\max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T (\mathrm{l}\mathrm{l}^T \circ \mathbf{K})\boldsymbol{\alpha} + \mathbf{1}^T \boldsymbol{\alpha}$$

where $\mathbf{K} = \Phi^T \Phi$

Support Vector Machine

The problem

$$\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}^T(\mathrm{ll}^T \circ \mathbf{K})\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\alpha}$$

is a quadratic problem having one global solution when K is positive semi-definite.

After obtaining **α**, **w** is calculated by $\mathbf{w} = \sum_{i=1}^{N} \alpha_i l_i \phi_i$

b can be calculated by selecting a training sample for which $\alpha_i > 0$ and computing

$$b = \mathbf{w}^T\phi_i - l_i$$

Kernels

We can use any function κ(⋅,⋅) defined on vector-pairs in order to calculate the elements of a matrix $\mathbf{K}_{ij}$ as long as the resulting matrix **K** is positive semi-definite.

Some example functions are

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2}$$

There is an equevalent sigma =1/(2sigma^2)

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^d$$

Using a generic function κ(⋅,⋅), we have

$$g(\mathbf{x}_*) = \mathbf{w}^T \phi_* - b = \sum_{i=1}^{N} l_i \alpha_i \phi_i^T \phi_* - b = \boldsymbol{\alpha}^T \mathbf{L} \mathbf{k}_* - b$$

Kernel Least-Means Square Regression

We can also define a linear regression using the data $\phi_i \in \mathcal{F},\ i = 1, \ldots, N$

$$\mathbf{W}^T \phi_i = \mathbf{t}_i,\ i = 1, \ldots, N$$

In order to determine the matrix W, we optimize for

$$\begin{aligned} \mathcal{J}_{LSE} &= \|\mathbf{W}^T \Phi - \mathbf{T}\|_F^2 \\ &= Tr\left(\mathbf{W}^T \Phi \Phi^T \mathbf{W} - 2\mathbf{W}^T \Phi \mathbf{T} + \mathbf{T}\mathbf{T}^T\right) \end{aligned}$$

where $\Phi = [\phi_1, \ldots, \phi_N]$, $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]$ and $Tr(\cdot)$ is the trace operator of a matrix.

Kernel Least-Means Square Regression

Setting the derivative w.r.t. **W** equal to zero, we have

$$\nabla \mathcal{J}_{LSE} = 0 \Rightarrow 2\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{W} = 2\mathbf{\Phi}\mathbf{T}^T$$

leading to

$$\mathbf{W} = \left(\mathbf{\Phi}\mathbf{\Phi}^T\right)^{-1}\mathbf{\Phi}\mathbf{T}^T = \mathbf{\Phi}^{\dagger}\mathbf{T}^T$$

when the mapping **x** → **φ** is defined, we can use the above equation to calculate **W**.

Kernel Least-Means Square Regression

When the mapping **x** → **φ** is defined through the function κ(∙,∙), we express W as a linear combination of the training samples

Representer theorem

$$\mathbf{W} = \mathbf{\Phi}\mathbf{A}$$

Substituting W to J_{LSE}, we obtain

$$\begin{aligned}\mathcal{J}_{LSE} &= \|\mathbf{A}^T\mathbf{\Phi}^T\mathbf{\Phi} - \mathbf{T}\|_F^2 = \|\mathbf{A}^T\mathbf{K} - \mathbf{T}\|_F^2 \\ &= Tr\left(\mathbf{A}^T\mathbf{K}\mathbf{K}^T\mathbf{A} - 2\mathbf{A}^T\mathbf{K}\mathbf{T} + \mathbf{T}\mathbf{T}^T\right)\end{aligned}$$

Why does K need to be positive semi-definite? Suppose we have KK=USU^T, then we can write US(1/2)U^T U^T S(1/2) U^T

Kernel Least-Means Square Regression

When the mapping $\mathbf{x} \rightarrow \boldsymbol{\varphi}$ is defined through the function $\kappa(\cdot,\cdot)$, we express W as a linear combination of the training samples

$$\mathbf{W} = \mathbf{\Phi}\mathbf{A}$$

Substituting W to J_{LSE}, we obtain

$$\begin{aligned} \mathcal{J}_{LSE} &= \|\mathbf{A}^T\mathbf{\Phi}^T\mathbf{\Phi} - \mathbf{T}\|_F^2 = \|\mathbf{A}^T\mathbf{K} - \mathbf{T}\|_F^2 \\ &= Tr\left(\mathbf{A}^T\mathbf{K}\mathbf{K}^T\mathbf{A} - 2\mathbf{A}^T\mathbf{K}\mathbf{T} + \mathbf{T}\mathbf{T}^T\right) \end{aligned}$$

Setting the derivative to zero we have $\nabla\mathcal{J}_{LSE} = 0 \Rightarrow 2\mathbf{K}\mathbf{K}^T\mathbf{A} = 2\mathbf{K}\mathbf{T}^T$

$$\mathbf{A} = \left(\mathbf{K}\mathbf{K}^T\right)^{-1}\mathbf{K}\mathbf{T}^T = \mathbf{K}^{\dagger}\mathbf{T}^T$$

Kernel Discriminant Analysis

We can also define a linear projection using the data $\phi_i \in \mathcal{F},\ i = 1, \ldots, N$

In this case, the two scatter matrices are

In order to determine the matrix W, we optimize for

$$\mathbf{S}_w = \sum_{k=1}^{K} \sum_{i, l_i = k} (\phi_i - \boldsymbol{\mu}_k)(\phi_i - \boldsymbol{\mu}_k)^T \qquad \mathbf{S}_b = \sum_{k=1}^{K} N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T$$

where

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i, l_i = k} \phi_i = \frac{1}{N_k} \boldsymbol{\Phi} \mathbf{1}_k \qquad \boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \phi_i = \frac{1}{N} \boldsymbol{\Phi} \mathbf{1}$$

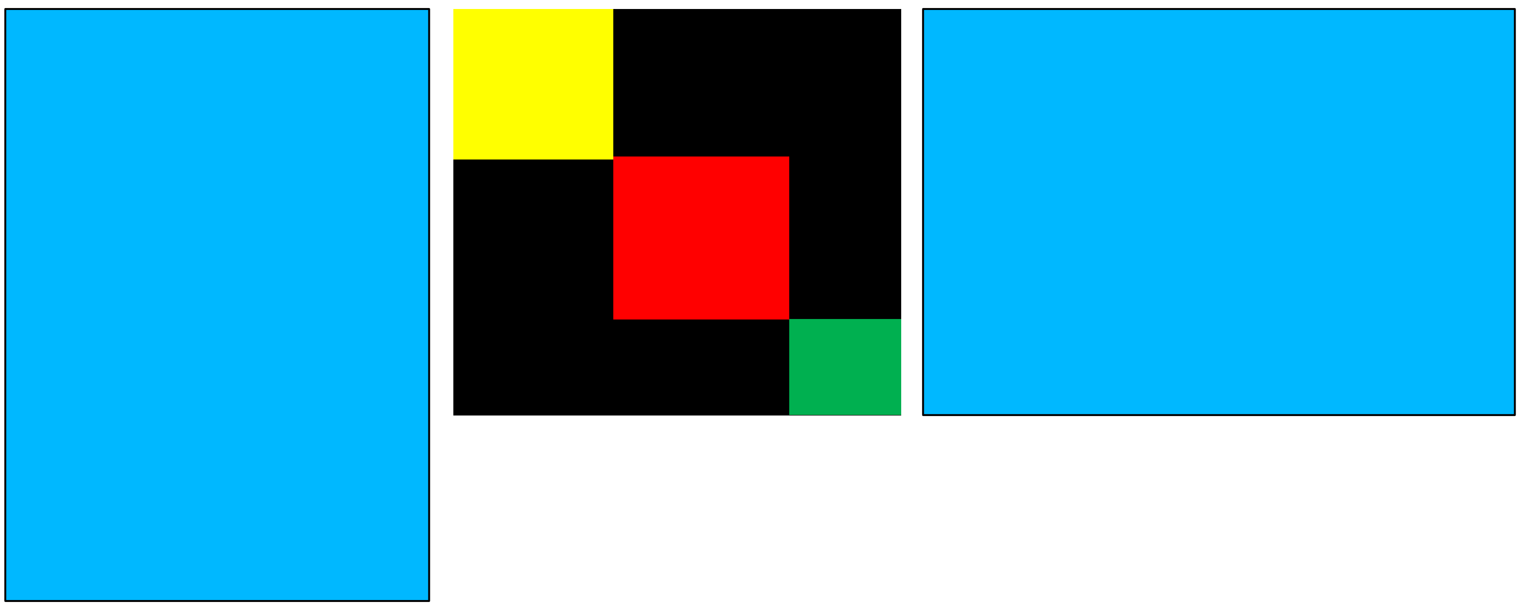
Kernel Discriminant Analysis

Substituting $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}$ in the scatter matrices, we get

$$\begin{aligned}
\mathbf{S}_w &= \sum_{k=1}^{K} \left(\boldsymbol{\Phi}\mathbf{J}_k - \frac{1}{N_k}\boldsymbol{\Phi}\mathbf{1}_k\mathbf{1}_k^T \right) \left(\boldsymbol{\Phi}\mathbf{J}_k - \frac{1}{N_k}\boldsymbol{\Phi}\mathbf{1}_k\mathbf{1}_k^T \right)^T \\
&= \sum_{k=1}^{K} \boldsymbol{\Phi} \left(\mathbf{J}_k - \frac{1}{N_k}\mathbf{1}_k\mathbf{1}_k^T \right) \left(\mathbf{J}_k - \frac{1}{N_k}\mathbf{1}_k\mathbf{1}_k^T \right)^T \boldsymbol{\Phi}^T \\
&= \boldsymbol{\Phi} \left(\sum_{k=1}^{K} \left(\mathbf{J}_k - \frac{1}{N_k}\mathbf{1}_k\mathbf{1}_k^T \right) \left(\mathbf{J}_k - \frac{1}{N_k}\mathbf{1}_k\mathbf{1}_k^T \right)^T \right) \boldsymbol{\Phi}^T \\
&= \boldsymbol{\Phi}\mathbf{L}_w\boldsymbol{\Phi}^T,
\end{aligned}$$

Kernel Discriminant Analysis

$$S_w = \Phi \quad L_w \quad \Phi^T$$



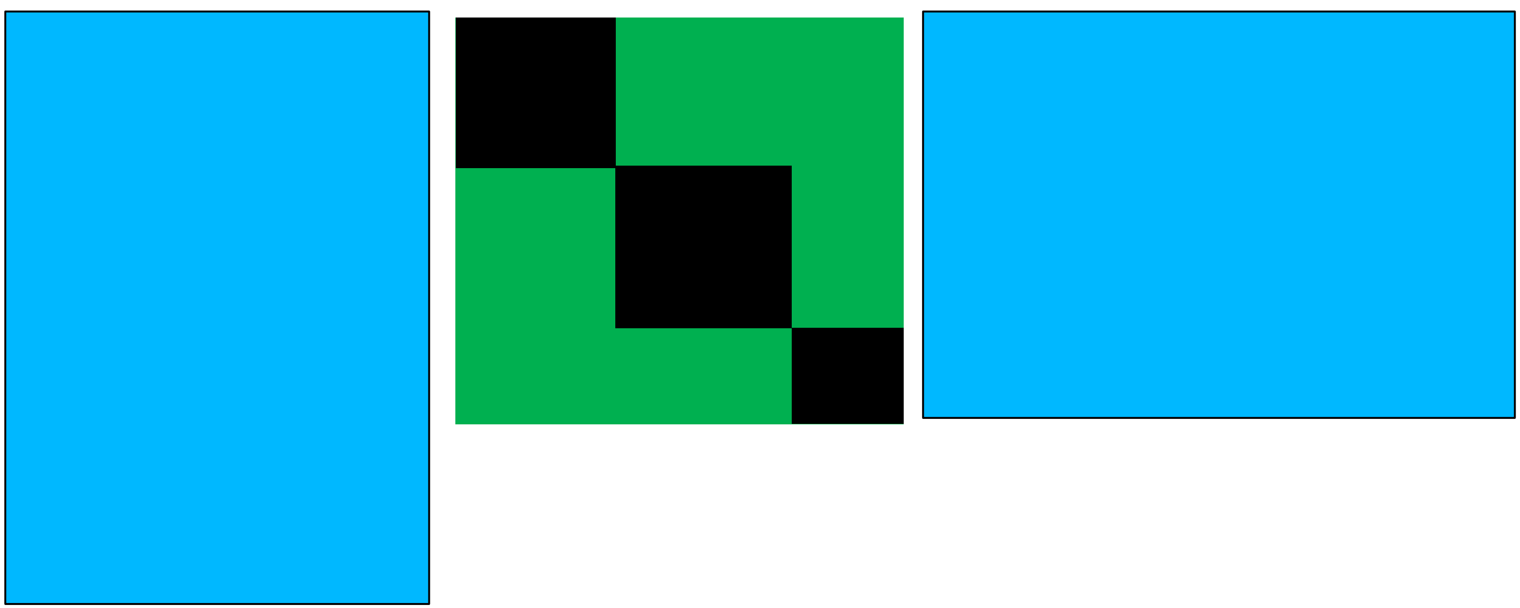
Kernel Discriminant Analysis

Substituting $\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}$ in the scatter matrices, we get

$$
\begin{aligned}
\mathbf{S}_b &= \sum_{k=1}^{K} N_k \left(\frac{1}{N_k}\boldsymbol{\Phi}\mathbf{1}_k\mathbf{1}_k^T - \frac{1}{N}\boldsymbol{\Phi}\mathbf{1}\mathbf{1}_k^T\right)\left(\frac{1}{N_k}\boldsymbol{\Phi}\mathbf{1}_k\mathbf{1}_k^T - \frac{1}{N}\boldsymbol{\Phi}\mathbf{1}\mathbf{1}_k^T\right)^T \\
&= \boldsymbol{\Phi}\left(\sum_{k=1}^{K} N_k \left(\frac{1}{N_k}\mathbf{1}_k\mathbf{1}_k^T - \frac{1}{N}\mathbf{1}\mathbf{1}_k^T\right)\left(\frac{1}{N_k}\mathbf{1}_k\mathbf{1}_k^T - \frac{1}{N}\mathbf{1}\mathbf{1}_k^T\right)^T\right)\boldsymbol{\Phi}^T \\
&= \boldsymbol{\Phi}\mathbf{L}_b\boldsymbol{\Phi}^T.
\end{aligned}
$$

Kernel Discriminant Analysis

$$\mathbf{S}_b = \mathbf{\Phi} \quad \mathbf{L}_b \quad \mathbf{\Phi}^T$$



Kernel Discriminant Analysis

Thus, the optimization criterion becomes

$$\mathcal{J}(\mathbf{W}) = \frac{Tr(\mathbf{W}^T \left(\mathbf{\Phi}\mathbf{L}_b\mathbf{\Phi}^T\right) \mathbf{W})}{Tr(\mathbf{W}^T \left(\mathbf{\Phi}\mathbf{L}_w\mathbf{\Phi}^T\right) \mathbf{W})}$$

or using $\mathbf{W} = \mathbf{\Phi}\mathbf{A}$

$$\mathcal{J}(\mathbf{A}) = \frac{Tr(\mathbf{A}^T \left(\mathbf{K}\mathbf{L}_b\mathbf{K}^T\right) \mathbf{A})}{Tr(\mathbf{A}^T \left(\mathbf{K}\mathbf{L}_w\mathbf{K}^T\right) \mathbf{A})} = \frac{Tr(\mathbf{A}^T\mathbf{S}_b^{(A)}\mathbf{A})}{Tr(\mathbf{A}^T\mathbf{S}_w^{(A)}\mathbf{A})}$$

Kernel Discriminant Analysis

Thus, the optimization criterion becomes

$$\mathcal{J}(\mathbf{W}) = \frac{Tr(\mathbf{W}^T \left(\mathbf{\Phi} \mathbf{L}_b \mathbf{\Phi}^T\right) \mathbf{W})}{Tr(\mathbf{W}^T \left(\mathbf{\Phi} \mathbf{L}_w \mathbf{\Phi}^T\right) \mathbf{W})}$$

or using $\mathbf{W} = \mathbf{\Phi} \mathbf{A}$

$$\mathcal{J}(\mathbf{A}) = \frac{Tr(\mathbf{A}^T \left(\mathbf{K} \mathbf{L}_b \mathbf{K}^T\right) \mathbf{A})}{Tr(\mathbf{A}^T \left(\mathbf{K} \mathbf{L}_w \mathbf{K}^T\right) \mathbf{A})} = \frac{Tr(\mathbf{A}^T \mathbf{S}_b^{(A)} \mathbf{A})}{Tr(\mathbf{A}^T \mathbf{S}_w^{(A)} \mathbf{A})}$$

which is solved by solving $\mathbf{S}_b^{(A)} \mathbf{a} = \lambda \mathbf{S}_w^{(A)} \mathbf{a}$

Kernel Discriminant Analysis

Demo SVM:

http://cs.stanford.edu/people/karpathy/svmjs/demo/

http://vision.stanford.edu/teaching/cs231n-demos/linear-classify/

Demo MLP:

http://cs.stanford.edu/~karpathy/svmjs/demo/demonn.html