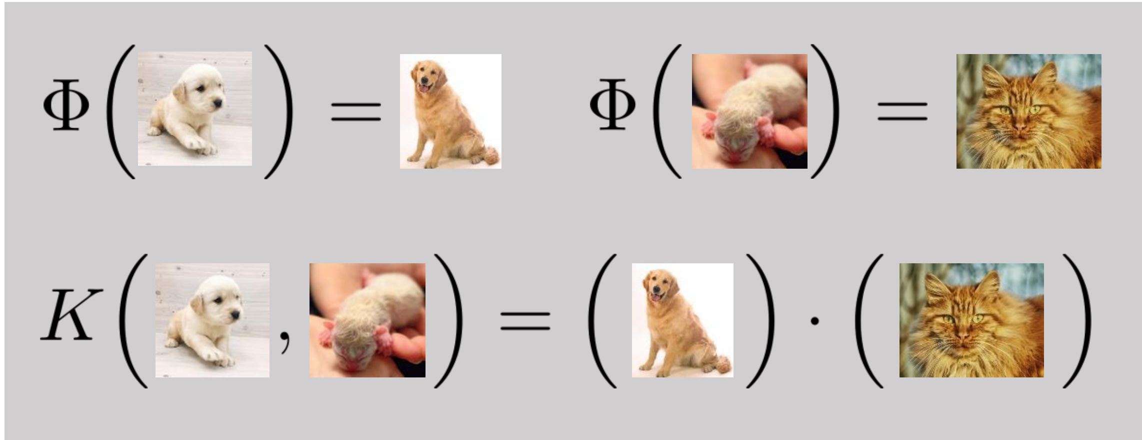
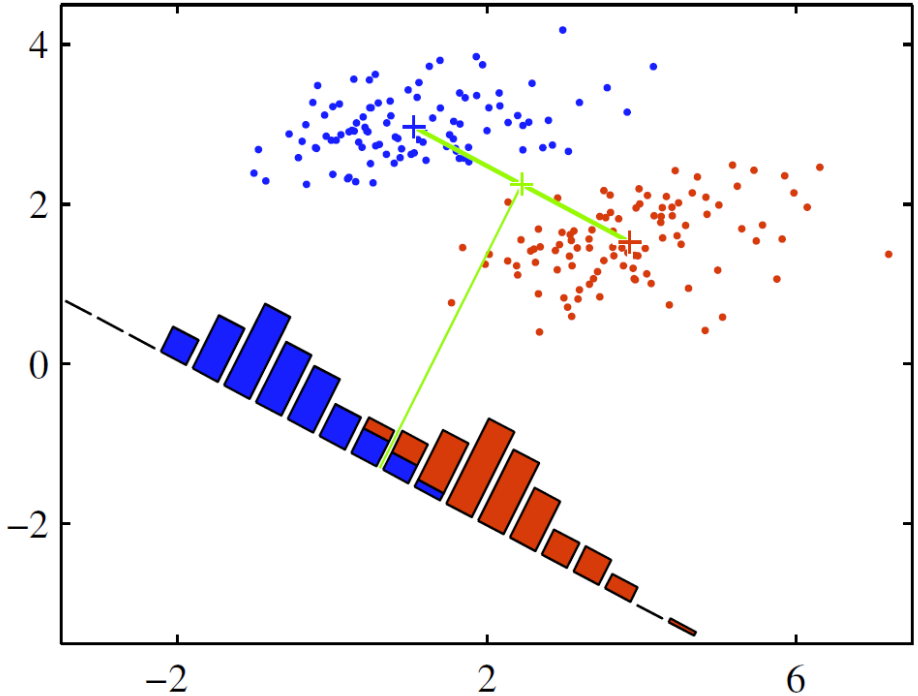


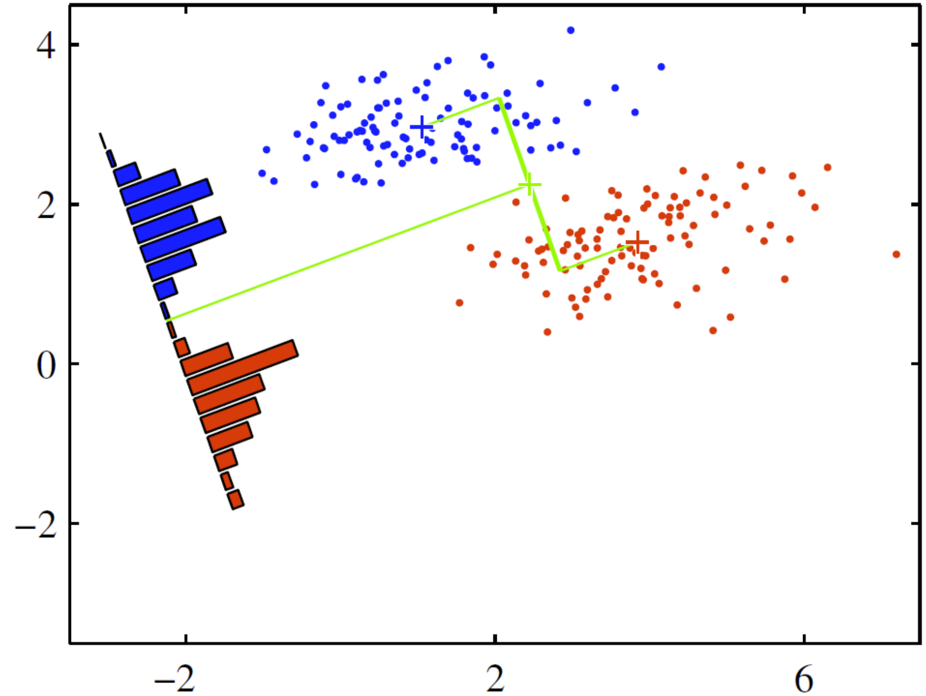
Kernel-based Learning:
Kernel Fisher Discriminant Analysis

Pilsung Kang
School of Industrial Management Engineering
Korea University

Linear Discriminant Analysis

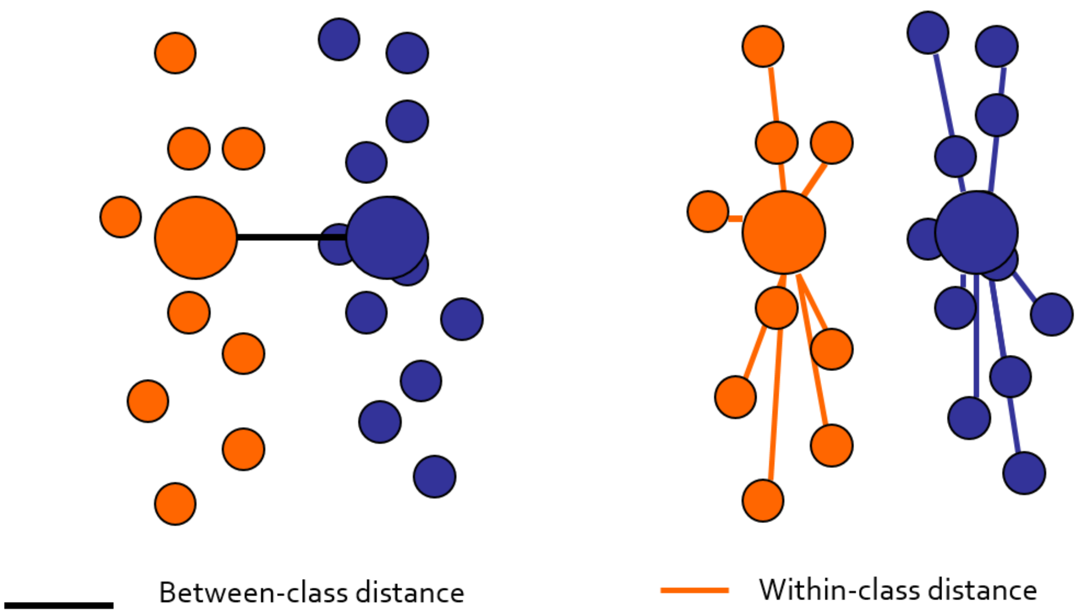
- LDA
  - ✓ Find a line to which two classes are well separated after projection





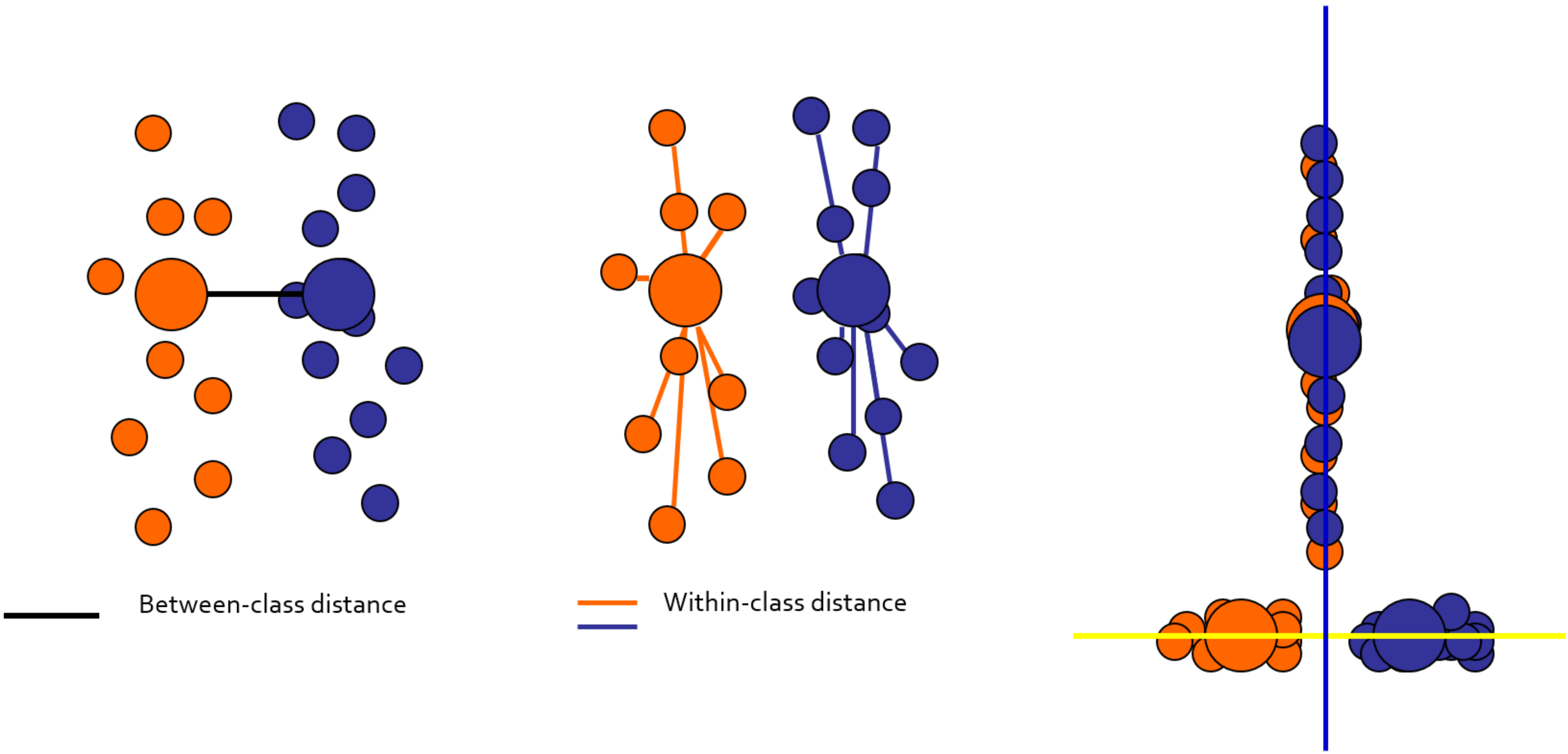
Linear Discriminant Analysis

- Two type of class distances
  - ✓ Between-class distance
    - Distance between the centroids of different classes
  - ✓ Within-class distance
    - Accumulated distance of an instance to the centroid of its class



Linear Discriminant Analysis

- (Fisher's) Linear Discriminant Analysis
  - ✓ Find most discriminant projection by maximizing between-class distance (variance) and minimizing within-class distance (variance)



Linear Discriminant Analysis

- Fisher's LDA (cont')
  - ✓ Take the D-dimensional input vector x and project it down to one dim.

$$y = \mathbf{w}^T \mathbf{x}$$

  - ✓ Consider a two-class problem in which there are N_1 & N_2 observations in C_1 and C_2, respectively.

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in C_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in C_2} \mathbf{x}_n$$

  - ✓ Objective 1: Choose w to maximize the separation of the projected class means (between class variance)

$$m_2 - m_1 = \mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1), \quad m_k = \mathbf{w}^T \mathbf{m}_k$$

Linear Discriminant Analysis

- Fisher's LDA (cont')
  - ✓ Objective 2: Choose w to minimize the variance in each class after projection (within class variance)

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$$

  - ✓ Fisher's criterion
    - The ratio of the between-class variance to the within-class variance

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

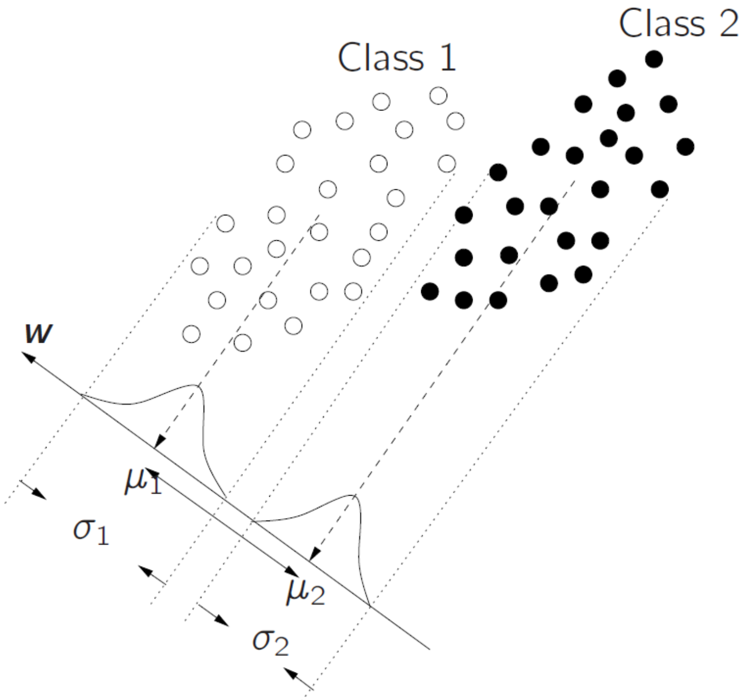
Linear Discriminant Analysis

- Fisher's LDA (cont')
  - ✓ Find w
    - Differentiating the Fisher's criterion w.r.t. w, then $J(\mathrm{w})$ is maximized when

$$(\mathbf{w}^T\mathbf{S}_B\mathbf{w})\mathbf{S}_W\mathbf{w} = (\mathbf{w}^T\mathbf{S}_W\mathbf{w})\mathbf{S}_B\mathbf{w}$$

    - $\mathrm{S_B w}$ is always in the direction of $(\mathrm{m_2}\text{-}\mathrm{m_1})$
    - Can drop the scalar factor $(\mathrm{w^T S_B w})$ and $(\mathrm{w^T S_W w})$
    - Then, obtain *Fisher's linear discriminant*

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$



Kernel Fisher Discriminant (KFD)

Mika (2002)

- Extend the LDA formulation by introducing kernels
  - ✓ KFD formulation
    - The full covariance of a dataset **Z** in the feature space by

$$\mathbf{C}^{\Phi} = \frac{1}{N}\sum_{n=1}^{N}(\Phi(\mathbf{x}_n) - \mathbf{m}^{\Phi})(\Phi(\mathbf{x}_n) - \mathbf{m}^{\Phi})^T, \quad \mathbf{m}^{\Phi} = \frac{1}{N}\sum_{n=1}^{N}\Phi(\mathbf{x}_n)$$

    - The within-class variance and the between-class variance in the feature space are given by

$$\mathbf{S}_W^{\Phi} = \sum_{i=1,2}\sum_{n=1}^{N_i}(\Phi(\mathbf{x}_n^i) - \mathbf{m}_i^{\Phi})(\Phi(\mathbf{x}_n^i) - \mathbf{m}_i^{\Phi})^T$$

$$\mathbf{S}_B = (\mathbf{m}_2^{\Phi} - \mathbf{m}_1^{\Phi})(\mathbf{m}_2^{\Phi} - \mathbf{m}_1^{\Phi})^T \qquad \mathbf{m}_i^{\Phi} = \frac{1}{N_i}\sum_{j=1}^{N_i}\Phi(\mathbf{x}_j^i)$$

Kernel Fisher Discriminant (KFD)

Mika (2002)

- Extend the LDA formulation by introducing kernels
  - ✓ Objective functions

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B^{\Phi} \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W^{\Phi} \mathbf{w}}$$

  - ✓ Projected vector

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n \Phi(\mathbf{x}_n), \quad \alpha_n \in R$$

  - ✓ Projected mean

$$\mathbf{w}^T \mathbf{m}_i^{\Phi} = \frac{1}{N_i} \sum_{n=1}^{N} \sum_{k=1}^{N_i} \alpha_n (\Phi(\mathbf{x}_n) \cdot \Phi(\mathbf{x}_k^i)) = \frac{1}{N_i} \sum_{n=1}^{N} \sum_{k=1}^{N_i} \alpha_n \mathbf{K}(\mathbf{x}_n, \mathbf{x}_k^i) = \boldsymbol{\alpha}^T \boldsymbol{\mu}_i$$

$$(\boldsymbol{\mu}_i)_n = \frac{1}{N_i} \sum_{k=1}^{N_i} \mathbf{K}(\mathbf{x}_n, \mathbf{x}_k^i)$$

Kernel Fisher Discriminant (KFD)

Mika (2002)

- Extend the LDA formulation by introducing kernels
  - ✓ Objective function (Numerator)

$$\mathbf{w}^T \mathbf{S}_B^{\Phi} \mathbf{w} = \mathbf{w}^T (\mathbf{m}_2^{\Phi} - \mathbf{m}_1^{\Phi})(\mathbf{m}_2^{\Phi} - \mathbf{m}_1^{\Phi})^T \mathbf{w}$$

$$= \boldsymbol{\alpha}^T (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{\alpha}$$

$$= \boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha}, \quad \text{where} \quad \mathbf{M} = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T$$

Kernel Fisher Discriminant (KFD)

Mika (2002)

- Extend the LDA formulation by introducing kernels
  - ✓ Objective function (Denominator)

$$\mathbf{w}^T \mathbf{S}_W^{\Phi} \mathbf{w} = \left(\sum_{i=1}^{N} \alpha_i \Phi(\mathbf{x}_i)\right) \left(\sum_{j=1,2} \sum_{n=1}^{N_i} (\Phi(\mathbf{x}_n^j) - \mathbf{m}_j^{\Phi})(\Phi(\mathbf{x}_n^j) - \mathbf{m}_j^{\Phi})^T \right) \sum_{k=1}^{N} \alpha_k \Phi(\mathbf{x}_k)$$

$$= \sum_{j=1,2} \sum_{i=1}^{N} \sum_{n=1}^{N_j} \sum_{k=1}^{N} \left(\alpha_i \Phi(\mathbf{x}_i) \left(\Phi(\mathbf{x}_n^j) - \mathbf{m}_j^{\Phi}\right) \left(\Phi(\mathbf{x}_n^j) - \mathbf{m}_j^{\Phi}\right)^T \alpha_k \Phi(\mathbf{x}_k) \right)$$

$$= \sum_{j=1,2} \sum_{i=1}^{N} \sum_{n=1}^{N_j} \sum_{k=1}^{N} \left(\alpha_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}_n^j) - \frac{1}{N_j} \sum_{p=1}^{N_j} \alpha_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}_p^j) \right) \times$$

$$\left(\alpha_k \mathbf{K}(\mathbf{x}_k, \mathbf{x}_n^j) - \frac{1}{N_j} \sum_{q=1}^{N_j} \alpha_k \mathbf{K}(\mathbf{x}_k, \mathbf{x}_q^j) \right)$$

Kernel Fisher Discriminant (KFD)

Mika (2002)

- Extend the LDA formulation by introducing kernels
  - ✓ Objective function (Denominator)

$$\sum_{j=1,2}\sum_{i=1}^{N}\sum_{n=1}^{N_j}\sum_{k=1}^{N}\left(\alpha_i\mathbf{K}(\mathbf{x}_i,\mathbf{x}_n^j)-\frac{1}{N_j}\sum_{p=1}^{N_j}\alpha_i\mathbf{K}(\mathbf{x}_i,\mathbf{x}_p^j)\right)\left(\alpha_k\mathbf{K}(\mathbf{x}_k,\mathbf{x}_n^j)-\frac{1}{N_j}\sum_{q=1}^{N_j}\alpha_k\mathbf{K}(\mathbf{x}_k,\mathbf{x}_q^j)\right)$$

$$=\sum_{j=1,2}\left(\sum_{i=1}^{N}\sum_{n=1}^{N_j}\sum_{k=1}^{N}\left(\alpha_i\alpha_k\mathbf{K}(\mathbf{x}_i,\mathbf{x}_n^j)\mathbf{K}(\mathbf{x}_k,\mathbf{x}_n^j)-\frac{2\alpha_i\alpha_k}{N_j}\sum_{p=1}^{N_j}\mathbf{K}(\mathbf{x}_i,\mathbf{x}_n^j)\mathbf{K}(\mathbf{x}_k,\mathbf{x}_p^j)\right.\right.$$

$$\left.\left.+\frac{\alpha_i\alpha_k}{N_j^2}\sum_{p=1}^{N_j}\sum_{q=1}^{N_j}\mathbf{K}(\mathbf{x}_i,\mathbf{x}_p^j)\mathbf{K}(\mathbf{x}_k,\mathbf{x}_q^j)\right)\right)$$

$$=\sum_{j=1,2}\left(\sum_{i=1}^{N}\sum_{n=1}^{N_j}\sum_{k=1}^{N}\left(\alpha_i\alpha_k\mathbf{K}(\mathbf{x}_i,\mathbf{x}_n^j)\mathbf{K}(\mathbf{x}_k,\mathbf{x}_n^j)-\frac{\alpha_i\alpha_k}{N_j}\sum_{p=1}^{N_j}\mathbf{K}(\mathbf{x}_i,\mathbf{x}_n^j)\mathbf{K}(\mathbf{x}_k,\mathbf{x}_p^j)\right)\right)$$

Kernel Fisher Discriminant (KFD)

Mika (2002)

- Extend the LDA formulation by introducing kernels
  - ✓ Objective function (Denominator)

$$\sum_{j=1,2}\left(\sum_{i=1}^{N}\sum_{n=1}^{N_j}\sum_{k=1}^{N}\left(\alpha_i\alpha_k\mathbf{K}(\mathbf{x}_i,\mathbf{x}_n^j)\mathbf{K}(\mathbf{x}_k,\mathbf{x}_n^j)-\frac{\alpha_i\alpha_k}{N_j}\sum_{p=1}^{N_j}\mathbf{K}(\mathbf{x}_i,\mathbf{x}_n^j)\mathbf{K}(\mathbf{x}_k,\mathbf{x}_p^j)\right)\right)$$

$$=\sum_{j=1,2}\boldsymbol{\alpha}^T\mathbf{K}_j\mathbf{K}_j^T\boldsymbol{\alpha}-\boldsymbol{\alpha}^T\mathbf{K}_j\mathbf{1}_{N_j}\mathbf{K}_j^T\boldsymbol{\alpha}$$

$$=\boldsymbol{\alpha}^T\mathbf{N}\boldsymbol{\alpha},\quad \text{where}\quad \mathbf{N}=\sum_{j=1,2}\mathbf{K}_j(\mathbf{I}-\mathbf{1}_{N_j})\mathbf{K}_j^T$$

  - ✓ $\mathbf{1}_{N_j}$: Gram matrix with the size of N_j by N_j with all elements equal to $1/N_j$

고려대학교 KOREA UNIVERSITY

DSBA Data Science & Business Analytics

Kernel Fisher Discriminant (KFD)

Mika (2002)

- Extend the LDA formulation by introducing kernels

  ✓ Objective function

$$J(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{N} \boldsymbol{\alpha}}$$

  ✓ Take the first derivative and set it equal to 0

$$(\boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha}) \mathbf{N} \boldsymbol{\alpha} = (\boldsymbol{\alpha}^T \mathbf{N} \boldsymbol{\alpha}) \mathbf{M} \boldsymbol{\alpha}$$

  ✓ Since $\mathbf{M}\boldsymbol{\alpha} = (\mathbf{M}_2 - \mathbf{M}_1)(\mathbf{M}_2 - \mathbf{M}_1)^T \boldsymbol{\alpha} = \lambda (\mathbf{M}_2 - \mathbf{M}_1)$,

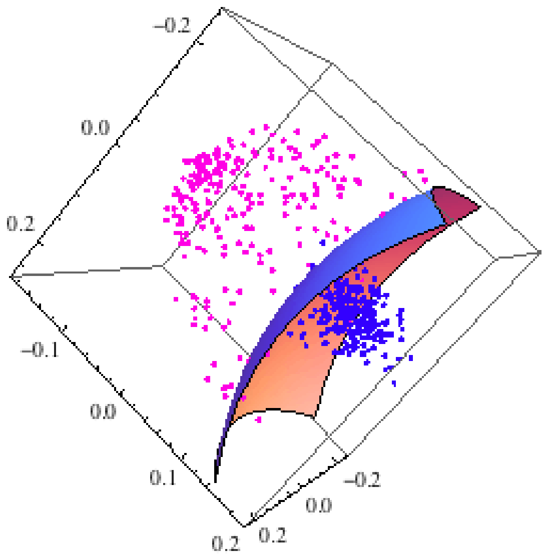
$$\boldsymbol{\alpha} = \mathbf{N}^{-1} (\mathbf{M}_2 - \mathbf{M}_1)$$

  ✓ Given the solution for α, the projection of a new data point is given by

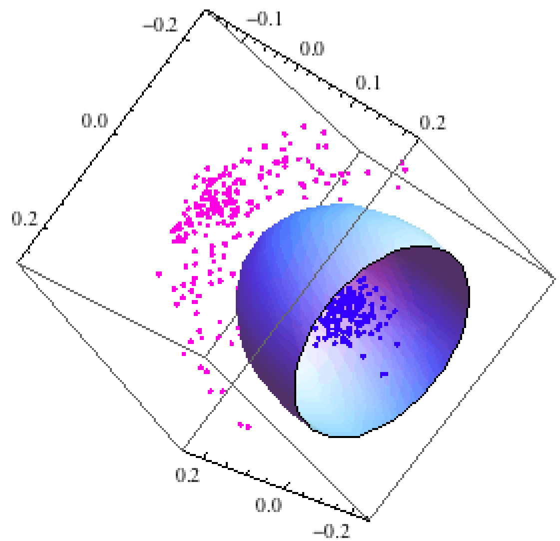
$$y(\mathbf{x}) = (\mathbf{w} \cdot \Phi(\mathbf{x})) = \sum_{n=1}^{N} \alpha_n \mathbf{K}(\mathbf{x}_n, \mathbf{x})$$

KFD Example

- KFD with a polynomial kernel



- KFD with a RBF (Gaussian) kernel



http://www.mathematica-journal.com/2011/07/fisher-discrimination-with-kernels/

KFD Performance

- Classification Performance

  ✓ Benchmark data sets

	dimensionality	Size of training set	Size of test set
Banana	2	400	4900
B.Cancer	9	200	77
Diabetes	8	468	300
German	20	700	300
Heart	13	170	100
Ringnorm	20	400	7000
F.Sonar	9	666	400
Thyroid	5	140	75
Titanic	3	150	2051
Waveform	21	400	4600

KFD Performance

- Classification Performance
  - ✓ Classification accuracy

	RBF	AB	AB_R	SVM	KFD
Banana	**10.8±0.06**	12.3±0.07	*10.9±0.04*	11.5±0.07	**10.8±0.05**
B.Cancer	27.6±0.47	30.4±0.47	26.5±0.45	*26.0±0.47*	**25.8±0.46**
Diabetes	24.3±0.19	26.5±0.23	23.8±0.18	*23.5±0.17*	**23.2±0.16**
German	24.7±0.24	27.5±0.25	24.3±0.21	**23.6±0.21**	*23.7±0.22*
Heart	17.6±0.33	20.3±0.34	16.5±0.35	**16.0±0.33**	*16.1±0.34*
Ringnorm	1.7±0.02	1.9±0.03	*1.6±0.01*	1.7±0.01	**1.5±0.01**
F.Sonar	34.4±0.20	35.7±0.18	34.2±0.22	**32.4±0.18**	*33.2±0.17*
Thyroid	4.5±0.21	*4.4±0.22*	4.6±0.22	4.8±0.22	**4.2±0.21**
Titanic	23.3±0.13	*22.6±0.12*	*22.6±0.12*	**22.4±0.10**	23.2±0.20
Waveform	10.7±0.11	10.8±0.06	**9.8±0.08**	*9.9±0.04*	*9.9±0.04*
Average	18.0%	20.2%	17.5%	17.2%	17.2%

고려대학교 KOREA UNIVERSITY

DSBA Data Science & Business Analytics


Q & A

References

Research Papers

- Mika, S. Kernel Fisher Discriminants. Ph.D Thesis. https://opus4.kobv.de/opus4-tuberlin/files/482/mika_sebastian.pdf
- Müller, K., Mika, S., Rätsch, G., Tsuda, K., and Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. IEEE Transactions on Neural Networks 12(2): 181-201.

References

Other materials

- http://www.cs.nyu.edu/~mohri/icml2011-tutorial/
- Suykens, J. (2003). Least Squares Support Vector Machines. IJCNN 2003 Tutorial.
- Abu-Mostafa, Y. (2012). Lecture 14: Support Vector Machines. Learning From Data. Caltech.
- http://www.ci.tuwien.ac.at/~meyer/svm/final.pdf
- http://pubs.rsc.org/en/content/articlehtml/2010/an/b918972f
- Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017b). Understanding deep learning requires rethinking generalization. International Conference on Learning Representations. [Slides] http://pluskid.org/slides/ICLR2017.key
- Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017c). Understanding deep learning requires rethinking generalization. International Conference on Learning Representations. [Poster] http://pluskid.org/slides/ICLR2017-Poster.pdf
- Park, J.S. (2017). Deep learning and data. DSBA Lab Seminar. http://dsba.korea.ac.kr/wp/wp-content/seminar/Paper%20Review/Deep%20Learning%20and%20Data%20-%20%EB%B0%95%EC%9E%AC%EC%84%A0.pdf