Multiple Kernel Learning and Feature Space Denoising

Fei Yan, Josef Kittler and Krystian Mikolajczyk

Overview of the talk

- Kernel methods
  - Kernel methods: an overview
  - Three examples: kernel PCA, SVM, and kernel FDA
  - Connection between SVM and kernel FDA
- Multiple kernel learning
  - MKL: motivation
  - ℓ_p regularised multiple kernel FDA
  - The effect of regularisation norm in MKL
- MKL and feature space denoising
- Conclusions

Kernel Methods: an overview

- Kernel methods: one of the most active areas in ML
- Key idea of kernel methods:
  - Embed data in input space into high dimensional feature space
  - Apply linear methods in feature space
- Input space can be: vector, string, graph, etc.
- Embedding is implicit via a kernel function $k(\cdot, \cdot)$, which defines dot product in feature space
- Any algorithm that can be written with only dot products is "kernelisable"

What is PCA

- Principal component analysis (PCA): an orthogonal basis transformation
- Transform correlated variables into uncorrelated ones (principal components)
- Can be used for dimensionality reduction
- Retains as much variance as possible when reducing dimensionality

How PCA works

- Given m centred vectors: $\tilde{X} = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \cdots, \tilde{\mathbf{x}}_m)$
  - X: $\tilde{d} \times m$ data matrix,
- Eigen decomposition of covariance $\tilde{C} = \tilde{X}\tilde{X}^T$: $\tilde{C} = \tilde{V}\tilde{\Omega}\tilde{V}^T$
  - Diagonal matrix $\tilde{\Omega}$: eigenvalues
  - $\tilde{V} = (\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \cdots)$: eigenvectors, orthogonal basis sought
- Data can now be projected onto orthogonal basis
- Projecting only onto leading eigenvectors ⇒ dimensionality reduction with minimum variance loss

Kernelising PCA

- If we knew explicitly the mapping from input space to feature space $\mathbf{x}_i = \phi(\tilde{\mathbf{x}}_i)$:
- we could map all data: $X = \phi(\tilde{X})$, where X is $d \times m$
- diagonalise the covariance in feature space $C = XX^T$: $X^T CV = X^T V\Omega$: $KA = A\Delta$
  - Diagonal matrix Δ: eigenvalues
  - $V = (\mathbf{v}_1, \mathbf{v}_2, \cdots)$: orthogonal basis in feature space
- However... we have $\phi(\cdot)$ only implicitly via: $< \phi(\tilde{\mathbf{x}}_i), \phi(\tilde{\mathbf{x}}_j) >= k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$
- Kernelised PCA

Kernelising PCA

- Kernel matrix K: evaluation of kernel function on all pairs of samples; symmetric, positive semi-definite (PSD)
- Connection between C and K:
  - $C = XX^T$ and $K = X^T X$
  - C is $d \times d$ and K is $m \times m$
- C is not explicitly available but K is
- So we diagonalise K instead of C: $K = A \Delta A^T$
  - $A = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots)$: eigenvectors

Kernelising PCA

- Using the connection between C and K, we have:
    - C and K have the same eigenvalues
    - Their i^{th} eigenvectors are related by: $\mathbf{v}_i = X\boldsymbol{\alpha}_i$
- $\mathbf{v}_i$ is still not explicitly available: $\boldsymbol{\alpha}_i$ is, but X is not
- However... we are interested in projection onto the orthogonal basis, not the basis itself
- Projection onto $\mathbf{v}_i$: $X^T\mathbf{v}_i = X^T X\boldsymbol{\alpha}_i = K\boldsymbol{\alpha}_i$
- Both K and $\boldsymbol{\alpha}_i$ are available.

Support Vector Machine

- SVM: supervised learning as opposed to (kernel) PCA
- In binary classification setting: maximise the margin
- Integrating misclassification ⇒ soft margin svm:

$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+ \qquad (1)$$

  - $\mathbf{w}$: multiplicative inverse of the margin
  - $(x)_+ = \max(x, 0)$: hinge loss penalising empirical error
  - C: parameter controlling the tradeoff
  - $y_i \in \{+1, -1\}$: label of training sample i
  - Goal: seeking the hyperplane with maximum soft margin

Support Vector Machine

- SVM primal (1) is equivalent to its Lagrangian dual:

$$\begin{array}{rl} \max\limits_{\boldsymbol{\alpha}} & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K_{ij} \\ \text{subject to} & \sum_{i=1}^{m} y_i \alpha_i = 0, \quad \mathbf{0} \le \boldsymbol{\alpha} \le C\mathbf{1} \end{array} \tag{2}$$

- (2) depends only on kernel matrix K (and labels)
- Explicit mapping $\phi(\cdot)$ into feature space not needed
- SVM can be kernelised

Kernel FDA

- Kernel Fisher discriminant analysis: another supervised learning technique
- Seeking the projection **w** maximising Fisher criterion

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T \frac{m}{m^+ m^-} S_B \mathbf{w}}{\mathbf{w}^T (S_T + \lambda I)\mathbf{w}} \tag{3}$$

  - m: numbers of samples
  - m^+ and m^-: numbers of positive and negative samples
  - S_B and S_T: between class and total scatters
  - λ: regularisation parameter

Kernel FDA

- It can be proved that (3) is equivalent to

$$\min_{\mathbf{w}} ||(XP)^T\mathbf{w} - \mathbf{a}||^2 + \lambda||\mathbf{w}||^2 \tag{4}$$

  - P and $\mathbf{a}$: constants determined by labels
- (4) is equivalent to its Lagrangian dual:

$$\min_{\boldsymbol{\alpha}} \frac{1}{4}\boldsymbol{\alpha}^T(I + \frac{1}{\lambda}K)\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{a} \tag{5}$$

- (5) depends only on K (and labels): FDA can be kernelised

Connection between SVM and kernel FDA

- Like SVM, kernel FDA is a special cases of Tikhonov regularisation
- Goals of Tikhonov regularisation:
    - Small empirical error (loss function may vary)
    - At the same time small norm $\mathbf{w}^T\mathbf{w}$ (for good generalisation)
- λ controls the tradeoff between error and good generalisation
- Instead of SVM's hinge loss for empirical error, FDA uses squared loss

MKL: motivation

- A recap on kernel methods:
  - Embed (implicitly) into (very high dimensional) feature space
  - Implicitly: only need dot product in feature space, i.e., the kernel function $k(\cdot, \cdot)$
  - Apply linear methods in the feature space
  - Easy balance of capacity (empirical error) and generalisation (norm $\mathbf{w}^T\mathbf{w}$)
- These sound nice but what kernel function to use?
  - This choice is critically important, for it completely determines the embedding

MKL: motivation

- Ideal case: learn kernel function from data
- If that is hard, can we learn a good combination of given kernel matrices: the multiple kernel learning problem
- Given n $m \times m$ kernel matrices, $K_1, \cdots, K_n$
- Most MKL formulations consider linear combination:

$$K = \sum_{j=1}^{n} \beta_j K_j, \quad \beta_j \geq 0 \tag{6}$$

- Goal of MKL: learn the "optimal" weights $\boldsymbol{\beta} \in \mathbb{R}^n$

MKL: motivation

- Kernel matrix K_j: pairwise dot products in feature space j
- Geometrical interpretation of unweighted sum $K = \sum_{j=1}^{n} K_j$:
  - Cartesian product of the feature spaces
- Geometrical interpretation of weighted sum $K = \sum_{j=1}^{n} \beta_j K_j$:
  - Scale feature spaces with $\sqrt{\beta_j}$, then take Cartesian product
- Learning kernel weights: seeking the "optimal" scaling

MKL: motivation

- Some example definitions of “optimality”:
  - Soft margin ⇒ multiple kernel SVM
  - Fisher criterion ⇒ multiple kernel FDA
  - Other objectives: kernel alignment, KL divergence, etc.
- Next we propose an ℓ_p regularised MK-FDA
  - Learn kernel weights $\boldsymbol{\beta}$ by maximising Fisher Criterion
  - Regularise $\boldsymbol{\beta}$ with a general ℓ_p norm for any $p \geq 1$
  - Better performance than single kernel and fixed norm MK-FDA

ℓ_p MK-FDA: min-max formulation

- We rewrite the kernel FDA primal problem:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T \frac{m}{m^+ m^-} S_B \mathbf{w}}{\mathbf{w}^T (S_T + \lambda I)\mathbf{w}} \qquad (7)$$

- And its Lagrangian dual:

$$\min_{\boldsymbol{\alpha}} \frac{1}{4} \boldsymbol{\alpha}^T (I + \frac{1}{\lambda} K) \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{a} \qquad (8)$$

- For multikernel FDA, K can be chosen from a kernel set $\mathcal{K}$:

$$\max_{K \in \mathcal{K}} \min_{\boldsymbol{\alpha}} \frac{1}{4} \boldsymbol{\alpha}^T (I + \frac{1}{\lambda} K) \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathbf{a} \qquad (9)$$

ℓ_p MK-FDA: min-max formulation

- Consider linear combination: $\mathcal{K} = \{K = \sum_{i=1}^{n} \beta_i K_i : \boldsymbol{\beta} \geq \mathbf{0}\}$
- $\boldsymbol{\beta}$ must be regularised in order for (9) to be meaningful
- We propose a general ℓ_p regularisation for any $p \geq 1$: $\mathcal{K} = \{K = \sum_{i=1}^{n} \beta_i K_i : \boldsymbol{\beta} \geq \mathbf{0}, ||\boldsymbol{\beta}||_p \leq 1\}$
- Substituting into (9), the ℓ_p MK-FDA problem becomes:

$$\begin{aligned} \max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{4\lambda}\boldsymbol{\alpha}^T \sum_{i=1}^{n} \beta_i K_i \boldsymbol{\alpha} + \frac{1}{4}\boldsymbol{\alpha}^T\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{a} \\ \text{s.t.} \quad & \boldsymbol{\beta} \geq \mathbf{0}, \quad ||\boldsymbol{\beta}||_p \leq 1 \end{aligned} \tag{10}$$

ℓ_p MK-FDA: SIP formulation

- Semi-infinite program (SIP):
    - Finite number of variables, infinite many constraints
    - Efficient algorithms exist for solving SIP
- Min-max formulation (10) can be reformulated as a SIP:

$$\max_{\theta,\boldsymbol{\beta}} \quad \theta \tag{11}$$

$$\text{s.t.} \quad \boldsymbol{\beta} \geq \mathbf{0}, \quad ||\boldsymbol{\beta}||_p \leq 1, \quad S(\boldsymbol{\alpha},\boldsymbol{\beta}) \geq \theta \quad \forall \boldsymbol{\alpha} \in \mathbb{R}^m$$

where

$$S(\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{4\lambda}\boldsymbol{\alpha}^T \sum_{i=1}^{n} \beta_i K_i \boldsymbol{\alpha} + \frac{1}{4}\boldsymbol{\alpha}^T\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\mathbf{a} \tag{12}$$

ℓ_p MK-FDA: solving the SIP with column generation

- Column generation:
  - Divide SIP into inner and outer subproblems
  - Alternate between the two subproblems till convergence
- Inner subproblem:
  - unconstrained quadratic program
- Outer subproblem:
  - quadratically constrained linear program
- Very efficient, and convergence is guaranteed

Effect of regularisation norm: simulation

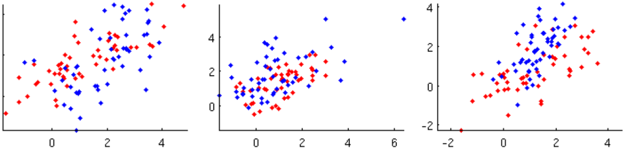

Figure: Distributions of two classes: 3 examples.

- Sample from two heavily overlapping Gaussian distributions
- Error rate of single kernel FDA with RBF kernel: $\sim$0.43
- Generate n kernels, apply ℓ_1 and ℓ_2 MK-FDAs, i.e. set $p = 1$ and $p = 2$ in ℓ_p MK-FDA

Effect of regularisation norm: simulation

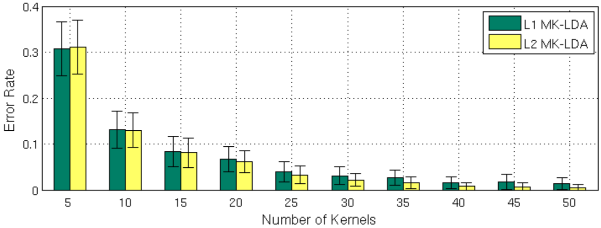

Figure: Error rate of ℓ_1 MK-FDA and ℓ_2 MK-FDA

- Both outperform single kernel, more kernels ⇒ lower error:
    - More kernels means more dimensions, better separability
- More kernels ⇒ more advantageous ℓ_2 is over ℓ_1. Why?

Effect of regularisation norm: simulation

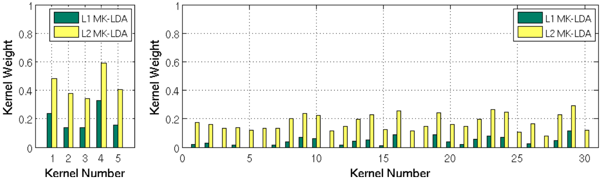

Figure: Leant kernel weights. Left: $n = 5$. Right: $n = 30$.

- Reason: when n is large, ℓ_1 regularisation gives sparse solution, resulting in loss of information

Effect of regularisation norm: Pascal VOC 2008

- Pascal VOC 2008 development set:
    - 20 object classes ⇒ 20 binary problems
    - Mean average precision (MAP) as performance metric
- 30 “informative” kernels:
    - Colour SIFTs as local descriptors
    - Bag-of-words model for kernel construction
- Mix informative kernels with 30 random kernels
    - 31 runs in total
    - 1st run: 0 informative + 30 random
    - 31st run: 30 informative + 0 random

Effect of regularisation norm: Pascal VOC 2008

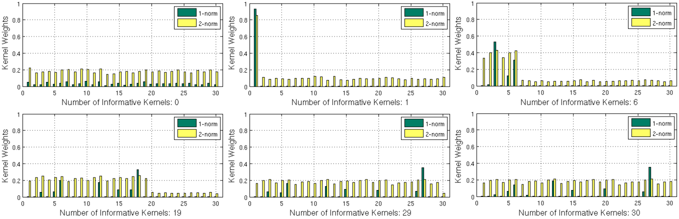

Figure: Learnt kernel weights with various kernel mixture.

- Again, ℓ_1 gives sparse solution and ℓ_2 non-sparse
- A hypothesis: when most kernels are informative sparsity is a bad thing and vice versa

Effect of regularisation norm: Pascal VOC 2008

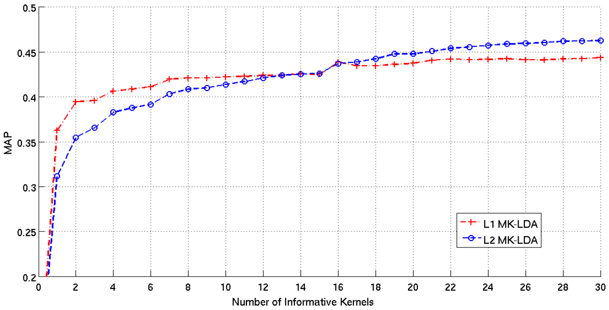

Figure: MAP vs. number of informative kernels

Effect of regularisation norm: Pascal VOC 2007

- We have seen the behaviour of ℓ_1 and ℓ_2 MK-FDAs
- A principle for selecting regularisation norm:
  - High intrinsic sparsity in base kernels: use small norm
  - Low intrinsic sparsity: use large norm
- But how do we know the intrinsic sparsity?
- Simple idea: try various norms, choose the best on validation
- ℓ_p MK-FDA allows us to do this

Effect of regularisation norm: Pascal VOC 2007

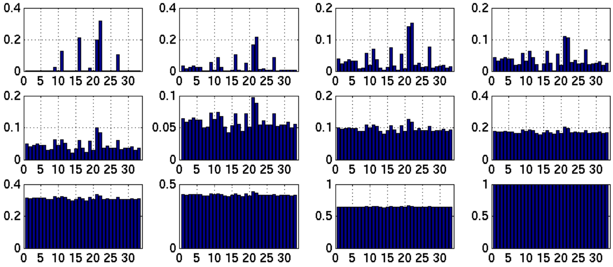

Figure: Learnt kernel weights on validation set with various p value. $p = \{1, 1+2^{-6}, 1+2^{-5}, 1+2^{-4}, 1+2^{-3}, 1+2^{-2}, 1+2^{-1}, 2, 3, 4, 8, 10^6\}$, and increases from left to right, top to bottom.

Effect of regularisation norm: Pascal VOC 2007

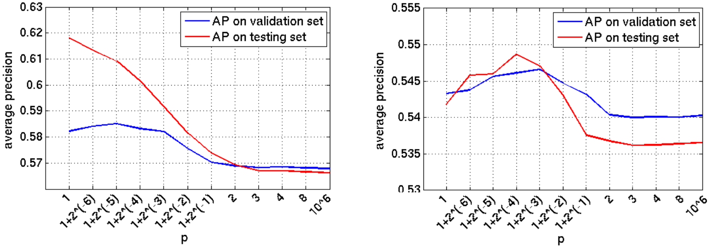

Figure: APs on validation set and test set with various p value. Left column: "dinningtable" class. Right column: "cat" class.

Effect of regularisation norm: Pascal VOC 2007

- As expected, the smaller the p, the more sparse the learnt weights
- $p = 10^6$ is practically ℓ_∞, i.e. uniform weighting
- Performance on validation and test sets matches well
  - A good p value on validation set is also good on test set
  - This means the optimal p, or the intrinsic sparsity, can be learnt

Effect of regularisation norm: Pascal VOC 2007

Table: Comparing ℓ_p MK-FDA and fixed norm MK-FDAs

	ℓ_1 MK-FDA	ℓ_2 MK-FDA	ℓ_∞ MK-FDA	ℓ_p MK-FDA
MAP	54.85	54.79	54.64	**55.61**

- By learning optimal p (intrinsic sparsity) for each class, ℓ_p MK-FDA outperforms fixed norm MK-FDA
- $\sim 1\%$ improvement is significant: leading methods in VOC challenges differ only by a few tenths of a percent

MKL and Denoising: Experimental setup

- PASCAL VOC07 dataset, same 33 kernels as before
- Use kernel PCA for dimensionality reduction (denoising) in feature space
- Questions to be answered:
  - Can denoising improve single kernel performance?
  - Can denoising improve MKL performance?
  - How MKL behaviour differs on original kernels and denoised kernels?

MKL and Denoising: Single kernel performance

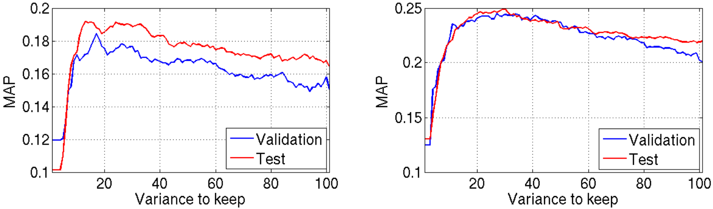

Figure: AP vs. variance kept in kernel PCA. Two kernels as examples.

- Choosing denoising level using a validation set ⇒ better single kernel performance (compared to original kernel)

MKL and Denoising: MKL performance

Table: Comparing ℓ_p MK-FDA and fixed norm MK-FDAs

	ℓ_1 MK-FDA	ℓ_2 MK-FDA	ℓ_∞ MK-FDA	ℓ_p MK-FDA
original kernels	54.85	54.79	54.64	**55.61**
denoised kernels	54.26	56.06	55.82	**56.17**

- In general, denoised kernels are better than original ones
- ℓ_p is better than fixed norm, on both original and denoised
- Advantage of ℓ_p is much smaller with denoised kernels. Why?

MKL and Denoising: Learnt kernel weight vs. noise level

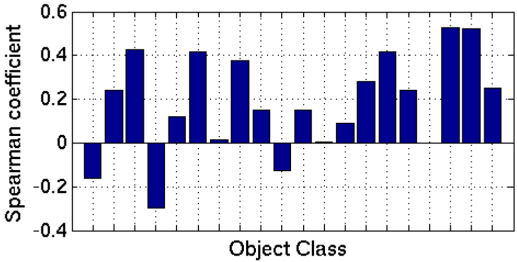

Figure: Spearman's coefficient between learnt kernel weights and variance kept in denoising. All 20 problems in PASCAL VOC07.

- Spearman's coefficient: measure ranking correlation

MKL and Denoising: Learnt kernel weight vs. noise level

- Positive coefficients on most problems (16 out of 20):
    - The more noisy a kernel, the lower weight it gets
    - MKL essentially works by removing noise?
    - Maybe this is why ℓ_p not as advantageous on denoised kernels?
    - Maybe MKL should be done on per dimension basis instead of per kernel basis?
    - Linear combination assigns same weight to all dimensions in a feature space: it cannot remove noise completely
    - Maybe only nonlinear MKL can be optimal?

Conclusions

- A brief introduction to kernel methods
  - The kernel trick
  - Three examples: kernel PCA, SVM, and kernel FDA
  - Connection between SVM and kernel FDA
- Proposed an MKL method: ℓ_p regularised MK-FDA
  - Regularisation norm plays an important role in MKL
  - ℓ_p MK-FDA allows to learn intrinsic sparsity of base kernels $\Rightarrow$ better performance than fixed norm MKL

Conclusions

- Investigated connection between MKL and feature space denoising
  - Denoising improves both single kernel and MKL performance
  - Positive correlation between weights and variance kept: the more noisy a kernel is, the lower its learnt weight
  - Linear kernel combination cannot take care of feature space denoising automatically
  - MKL should be done on per dimension basis instead of per kernel basis?
  - The optimal (non-linear) MKL is yet to be developed