Kernel Methods in ACASVA

Fei Yan

ACASVA Project Meeting
June 27, 2011
Queen Mary, University of London, London

Kernel Methods

- Powerful tool for machine learning
- In a nutshell:
  - Embed data implicitly in high dimensional space
  - Apply linear methods
- Why do they work?
  - In high dimensional space, linear functions extremely rich while remain easy to regularise
  - Theoretically well founded in statistical learning theory
- Numerous applications

Multiple Kernel Learning (MKL)

- Kernel function plays critical role
  - Embedding determined by the kernel function
- Ideally should be learnt from data
- A relaxed version: learning combination of kernels
  - Given n kernels $K_1, \cdots, K_n$, each $m \times m$
  - Find $\boldsymbol{\beta}$ such that $K = \sum_{j=1}^{n} \beta_j K_j$ optimal
  - Constraints: $\beta_j \geq 0, ||\boldsymbol{\beta}||_p \leq 1$
- p controls the sparsity of weights
  - $p \to 1$: most weights 0
  - $p \to \infty$: most weights 1

Large Scale MKL

- MKL can be expensive in memory and time
  - n kernels stored in memory: $O(nm^2)$
  - $n = 50, m = 8000$, double precision: 24G memory
  - Multiclass problems could take days to train
- Reason: wrapper algorithm
  - α step: solve a single kernel problem
    - Slow to solve the complete problem
    - Need access to complete kernel matrices
  - β step: find optimal kernel weights

Large Scale MKL

- We propose an interleaved algorithm
- Inspired by sequential minimal optimisation (SMO)
  - α step: optimise over a minimal subset
  - Generate enough gradient in the β direction
  - β step as before
- Do not need access to whole kernel matrices, only 2 rows
  - Compute kernel on-the-fly, much less memory needed
- Each α step much cheaper, but much more steps
  - Overall improvement: 1-2 orders of magnitude faster

Experiments on Player Action Data

- MKL for feature selection
  - One kernel each dimension, more flexibility
  - Large scale MKL makes it tractable
  - Recall that p controls sparsity
- Player action recognition data
  - Training: single's game, 1277 examples
  - Test: double's game, 1582 examples
  - Feature dimension $= 960$: 960 kernels
  - 3 (unbalanced) classes: idle, hit, serve
  - Speed improvement: several days vs. $\sim$ 1 hour

Experiments on Player Action Data

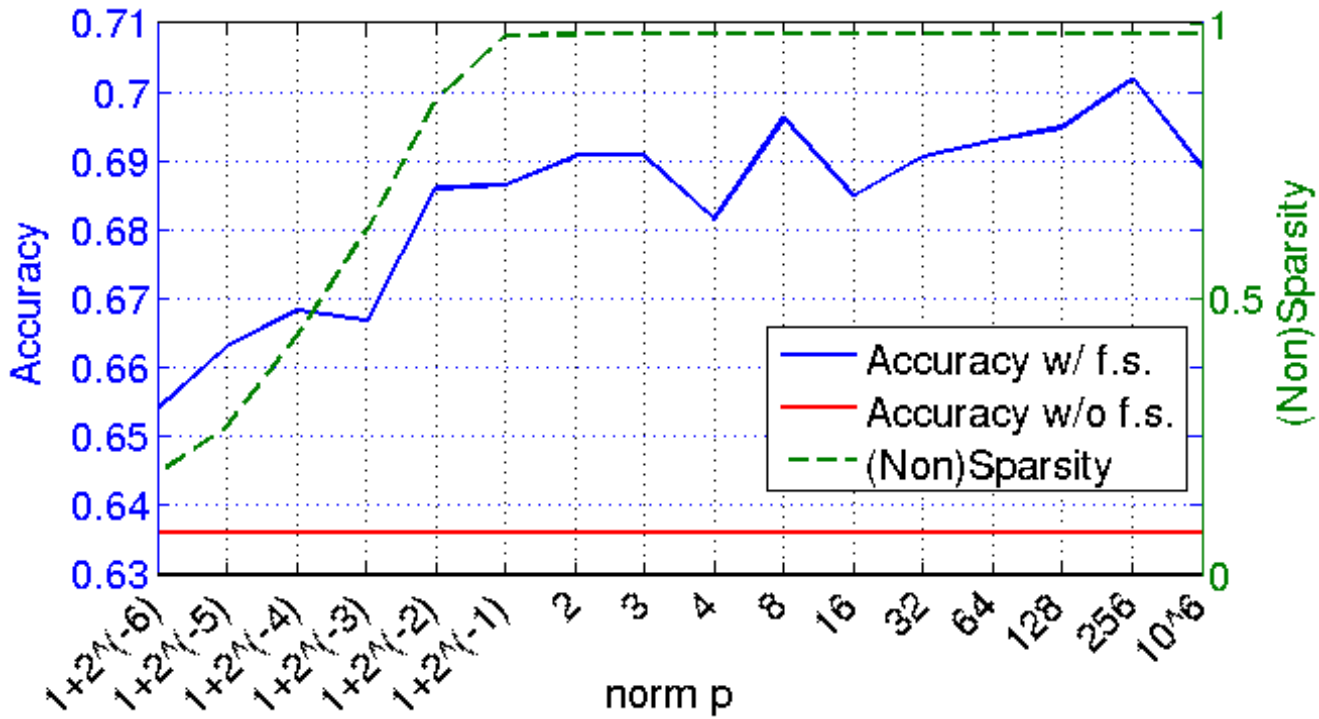

Figure: Average accuracy and sparsity of learnt weights

Experiments on Player Action Data

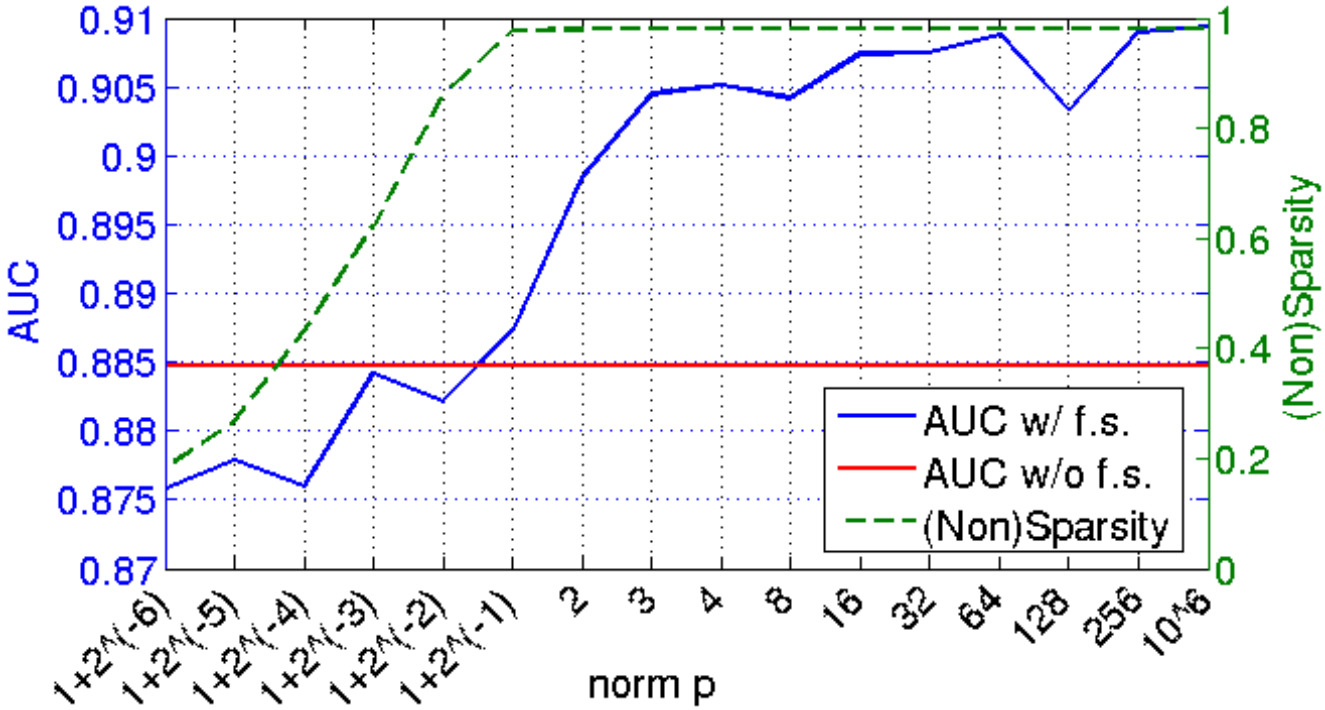

Figure: AUC and sparsity of learnt weights

Other Potential Applications of Kernel Methods

- Apply almost everywhere where learning takes place
- In particular, structured output learning (SOL)
  - Generalises kernel methods to structured output
  - ... such as sequences, graphs, rankings
  - Natural language processing, computational biology, computer vision
  - E.g. $\mathcal{X}$: sentences $\rightarrow$ $\mathcal{Y}$: parse trees / sentences in another language
  - Sports video annotation as SOL?