Kernel Learning for Novelty Detection

John Shawe-Taylor

University College London

NIPS Workshop Kernel Learning: Automatic Selection of Optimal Kernels, December 2008

Joint work with Zakria Hussain

Outline

Outline

Outline

1. Introduction
   - Motivating problem
   - 1-class SVMs
2. Multiple Kernel Learning
   - Method 1: constraining the 1-norm of the weight vectors
   - Method 2: constraining a convex combination of the 1-norm and 2-norm of the weight vectors
3. Experiments
   - Assessing impact of μ
   - Including negative examples
4. Conclusions

Outline

Outline

Content based image retrieval

- Consider problem of content based image retrieval (CBIR) using relevance feedback.
- There are many metrics under which we can compare images: colour, texture, objects included, etc.
- Learning to identify the target of the search is improved if we can identify the metric that best characterises the type of search: eg
  - sunset scene $\Rightarrow$ colour,
  - Sunset over waterfall $\Rightarrow$ colour & texture, etc.
- Baseline system is PicSOM – uses 11 self-organising maps (SOMs) to represent database of images in 11 metrics – estimates a density of relevant vs irrelevant to weight the vertices of each SOM.
- Implicitly reweights the metrics via the density.

Content based image retrieval

- Consider problem of content based image retrieval (CBIR) using relevance feedback.
- There are many metrics under which we can compare images: colour, texture, objects included, etc.
- Learning to identify the target of the search is improved if we can identify the metric that best characterises the type of search: eg
  - sunset scene ⇒ colour,
  - Sunset over waterfall ⇒ colour & texture, etc.
- Baseline system is PicSOM – uses 11 self-organising maps (SOMs) to represent database of images in 11 metrics – estimates a density of relevant vs irrelevant to weight the vertices of each SOM.
- Implicitly reweights the metrics via the density.

Content based image retrieval

- Consider problem of content based image retrieval (CBIR) using relevance feedback.
- There are many metrics under which we can compare images: colour, texture, objects included, etc.
- Learning to identify the target of the search is improved if we can identify the metric that best characterises the type of search: eg
  - sunset scene $\Rightarrow$ colour,
  - Sunset over waterfall $\Rightarrow$ colour & texture, etc.
- Baseline system is PicSOM – uses 11 self-organising maps (SOMs) to represent database of images in 11 metrics – estimates a density of relevant vs irrelevant to weight the vertices of each SOM.
- Implicitly reweights the metrics via the density.

Content based image retrieval

- Consider problem of content based image retrieval (CBIR) using relevance feedback.
- There are many metrics under which we can compare images: colour, texture, objects included, etc.
- Learning to identify the target of the search is improved if we can identify the metric that best characterises the type of search: eg
  - sunset scene $\Rightarrow$ colour,
  - Sunset over waterfall $\Rightarrow$ colour & texture, etc.
- Baseline system is PicSOM – uses 11 self-organising maps (SOMs) to represent database of images in 11 metrics – estimates a density of relevant vs irrelevant to weight the vertices of each SOM.
- Implicitly reweights the metrics via the density.

Content based image retrieval

- Consider problem of content based image retrieval (CBIR) using relevance feedback.
- There are many metrics under which we can compare images: colour, texture, objects included, etc.
- Learning to identify the target of the search is improved if we can identify the metric that best characterises the type of search: eg
  - sunset scene $\Rightarrow$ colour,
  - Sunset over waterfall $\Rightarrow$ colour & texture, etc.
- Baseline system is PicSOM – uses 11 self-organising maps (SOMs) to represent database of images in 11 metrics – estimates a density of relevant vs irrelevant to weight the vertices of each SOM.
- Implicitly reweights the metrics via the density.

1-class SVMs

- Negative data is sparse so consider 1-class learning initially.
- Metrics correspond to kernels: so task is about using combination of kernels to solve a retrieval task.
- If we include 'learning the kernel', we can automatically identify the relevant metrics for the particular search.
- Potential to scale to very large numbers of metrics/submetrics.

1-class SVMs

- Negative data is sparse so consider 1-class learning initially.
- Metrics correspond to kernels: so task is about using combination of kernels to solve a retrieval task.
- If we include 'learning the kernel', we can automatically identify the relevant metrics for the particular search.
- Potential to scale to very large numbers of metrics/submetrics.

1-class SVMs

- Negative data is sparse so consider 1-class learning initially.
- Metrics correspond to kernels: so task is about using combination of kernels to solve a retrieval task.
- If we include 'learning the kernel', we can automatically identify the relevant metrics for the particular search.
- Potential to scale to very large numbers of metrics/submetrics.

1-class SVMs

- Negative data is sparse so consider 1-class learning initially.
- Metrics correspond to kernels: so task is about using combination of kernels to solve a retrieval task.
- If we include 'learning the kernel', we can automatically identify the relevant metrics for the particular search.
- Potential to scale to very large numbers of metrics/submetrics.

Optimisation problem

$$\begin{array}{ll} \min_{\mathbf{w},\xi} & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\,\|\xi\|_1 \\ \text{subject to} & \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle \geq 1 - \xi_i \\ & \xi_i \geq 0, \;\; i = 1, \ldots, m \end{array}$$

Outline

A linear combination of kernels

Let κ_k denote the kth kernel from a set $\mathbb{K} = \{\kappa_1, \ldots, \kappa_{|\mathbb{K}|}\}$ of kernels. We define a weighted combination of kernels like so:

$$\kappa_{\mathbf{z}} = \sum_{k=1}^{|\mathbb{K}|} z_k \kappa_k$$

where $\mathbf{z} = \left(z_1, \ldots, z_{|\mathbb{K}|}\right)$, $z_i \in \mathbb{R}^+$.

MKL Optimisation: Constraining the 1-norm (Recap)

Let $K = |\mathbb{K}|$, then we have the following 1-class SVM for MKL when regularising over the weight vector using the 1-norm (primal):

$$\begin{array}{ll} \min_{\mathbf{w}_k,\xi} & \frac{1}{2}\left(\sum_{k=1}^{K}\|\mathbf{w}_k\|_2\right)^2 + C\,\|\xi\|_1 \\ \text{subject to} & \sum_{k=1}^{K}\langle \mathbf{w}_k, \phi_k(\mathbf{x}_i)\rangle \geq 1-\xi_i \\ & \xi_i \geq 0,\ \ i=1,\ldots,m \end{array}$$

Constraining the 1-norm continued

The dual becomes:

$$\begin{array}{ll} \min_\beta \max_{\boldsymbol{\alpha}} & \beta \\ \text{subject to} & \sum_{i,j=1}^m \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j) \leq \beta, \\ & \sum_{i=1}^m \alpha_i = 1, \\ & 0 \leq \alpha_i \leq C,\ i = 1, \ldots, m. \end{array}$$

Experimental issues

- We ran experiments with a set of Gaussian kernels composed of different width parameters.
- Problem: only chose 1 kernel for learning, namely the Gaussian kernel with the largest width parameter. This is also true for experiments conducted with the VOC data sets (cat, cow, dog).
- Our conjecture: except in degenerate cases will only choose 1 kernel.
- A solution: Constrain a convex combination of the 1-norm and 2-norm in the optimisation problem.

Experimental issues

- We ran experiments with a set of Gaussian kernels composed of different width parameters.
- Problem: only chose 1 kernel for learning, namely the Gaussian kernel with the largest width parameter. This is also true for experiments conducted with the VOC data sets (cat, cow, dog).
- Our conjecture: except in degenerate cases will only choose 1 kernel.
- A solution: Constrain a convex combination of the 1-norm and 2-norm in the optimisation problem.

Experimental issues

- We ran experiments with a set of Gaussian kernels composed of different width parameters.
- Problem: only chose 1 kernel for learning, namely the Gaussian kernel with the largest width parameter. This is also true for experiments conducted with the VOC data sets (cat, cow, dog).
- Our conjecture: except in degenerate cases will only choose 1 kernel.
- A solution: Constrain a convex combination of the 1-norm and 2-norm in the optimisation problem.

Experimental issues

- We ran experiments with a set of Gaussian kernels composed of different width parameters.
- Problem: only chose 1 kernel for learning, namely the Gaussian kernel with the largest width parameter. This is also true for experiments conducted with the VOC data sets (cat, cow, dog).
- Our conjecture: except in degenerate cases will only choose 1 kernel.
- A solution: Constrain a convex combination of the 1-norm and 2-norm in the optimisation problem.

MKL Optimisation: Constraining a combination of the 1-norm and 2-norm

Constraining using both these norms gives us the following primal problem, with μ controlling the sparsity trade-off,

$$\begin{array}{ll} \min_{\mathbf{w}_k,\xi} & \frac{\mu}{2}\left(\sum_{k=1}^{K}\|\mathbf{w}_k\|_2\right)^2 + \frac{1-\mu}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|_2^2 + C\,\|\xi\|_1 \\ \text{subject to} & \sum_{k=1}^{K}\langle \mathbf{w}_k, \phi_k(\mathbf{x}_i)\rangle \geq 1-\xi_i \\ & \xi_i \geq 0,\ \ i=1,\ldots,m \end{array}$$

Constraining a combination of the 1-norm and 2-norm continued

Let $D = \sum_{k=1}^{K} \|\mathbf{w}_k\|_2$, then the dual is:

$$\begin{array}{ll} \max_{\boldsymbol{\alpha}} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{A}{2} \sum_{k=1}^{K} \beta_k + \frac{B}{2} \left(\sum_{k \in J} \sqrt{\beta_k} \right)^2 \\ \text{subject to} & \beta_k = \sum_{i,j=1}^{m} \alpha_i \alpha_j \kappa_k(\mathbf{x}_i, \mathbf{x}_j) \\ & 0 \leq \alpha_i \leq C, \;\; i = 1, \ldots, m \end{array}$$

where $A = 1/(1-\mu)$ and
$B = ((|J|-1)\mu^2 + \mu)/((1-\mu)(1-\mu+\mu|J|)^2)$, where
$J = \{k : z_k \neq 0\}$, is the set of indices k, for which

$$\beta_k > \mu^2 D^2,$$

Combination of 1- and 2-norm continued

From the Lagrangian we get,

$$z_k = \max\left\{0, \frac{1}{1-\mu}\left(\frac{\sqrt{\beta_k}}{D} - \mu\right)\right\},$$

Also,

$$D = \frac{\sum_{k\in J}\sqrt{\beta_k}}{1-\mu+\mu|J|}$$

Algorithm for combination of 1- and 2-norm

We perform coordinate-wise descent in the $\boldsymbol{\alpha}$ vector. Writing

$$g_i(\alpha_i) = \frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_i},$$

where α_i is the i-th coordinate of $\boldsymbol{\alpha}$ in the argument of $W(\cdot)$, we seek the solution of $g_i(\alpha_i) = 0$ as the new value for α_i.

Algorithm for combination of 1- and 2-norm

We expand $g_i(\alpha_i)$ in a Taylor series around the current values α^0:

$$g_i(\alpha_i) \approx \frac{\partial W(\alpha^0)}{\partial \alpha_i} + \frac{\partial W^2(\alpha^0)}{\partial \alpha_i^2}(\alpha_i - \alpha_i^0) = 0$$

and solve for α_i. Hence our update rule for each α_i becomes:

$$\alpha_i = \alpha_i^0 - \frac{\frac{\partial W(\alpha^0)}{\partial \alpha_i}}{\frac{\partial W^2(\alpha^0)}{\partial \alpha_i^2}}.$$

Algorithm for combination of 1- and 2-norm

Initialise $\boldsymbol{\alpha}^0$ vector to zero with one element, say $\alpha_1^0 > 0$.

- Repeat until KKT conditions satisfied or $\|\boldsymbol{\alpha}^n - \boldsymbol{\alpha}^{n-1}\|_2 < \epsilon$, where ϵ is a small positive real number
  - Compute update rule for each component of $\boldsymbol{\alpha}$ using:

$$\alpha_i = \alpha_i^0 - \frac{\frac{\partial W(\alpha^0)}{\partial \alpha_i}}{\frac{\partial W^2(\alpha^0)}{\partial \alpha_i^2}}.$$

  - update $\mathbf{z}$ and D

Decision function:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \sum_{k \in J} \frac{z_k}{\mu + (1-\mu) z_k} \kappa_k(\mathbf{x}_i, \mathbf{x})$$

Outline

Datasets considered

- Considered the PASCAL VOC data: cat, cow, dog
- 11 feature sets extracted for PICSOM

Feature	dimensions
DCT coefficients of average colour in rectangular grid	12
CIE L*a*b* colour of two dominant colour clusters	6
Histogram of local edge statistics	80
Haar transform of quantized HSV colour histogram	256
Histogram of interest point SIFT features	256
Average CIE L*a*b* colour	15
Three central moments of CIE L*a*b* colour distribution	45
Histogram of four Sobel edge directions	20
Co-occurrence matrix of four Sobel edge directions	80
Magnitude of the 16×16 FFT of Sobel edge image	128
Histogram of relative brightness of neighboring pixels	40

Datasets considered

- Considered the PASCAL VOC data: cat, cow, dog
- 11 feature sets extracted for PICSOM

Feature	dimensions
DCT coefficients of average colour in rectangular grid	12
CIE L*a*b* colour of two dominant colour clusters	6
Histogram of local edge statistics	80
Haar transform of quantized HSV colour histogram	256
Histogram of interest point SIFT features	256
Average CIE L*a*b* colour	15
Three central moments of CIE L*a*b* colour distribution	45
Histogram of four Sobel edge directions	20
Co-occurrence matrix of four Sobel edge directions	80
Magnitude of the 16×16 FFT of Sobel edge image	128
Histogram of relative brightness of neighboring pixels	40

Effect of μ on sparsity

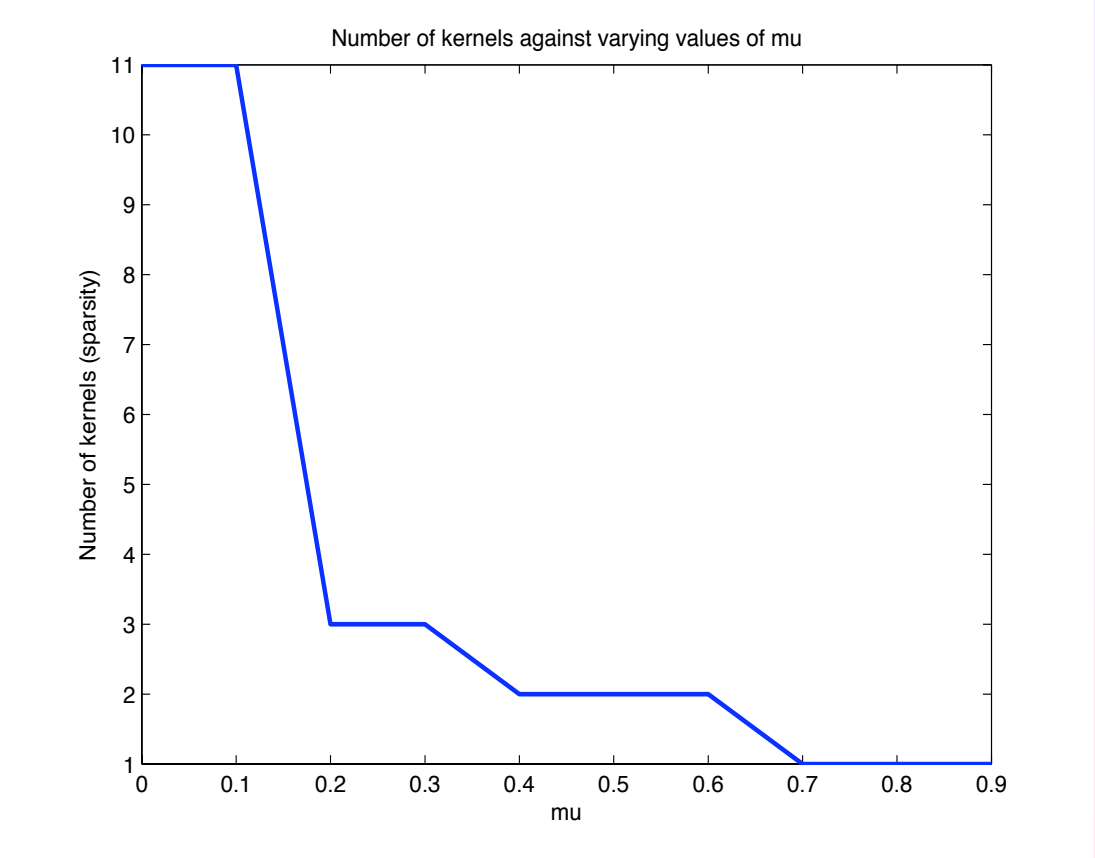

Figure: Sparsity as function of μ for cats

Effect of μ on retrieval

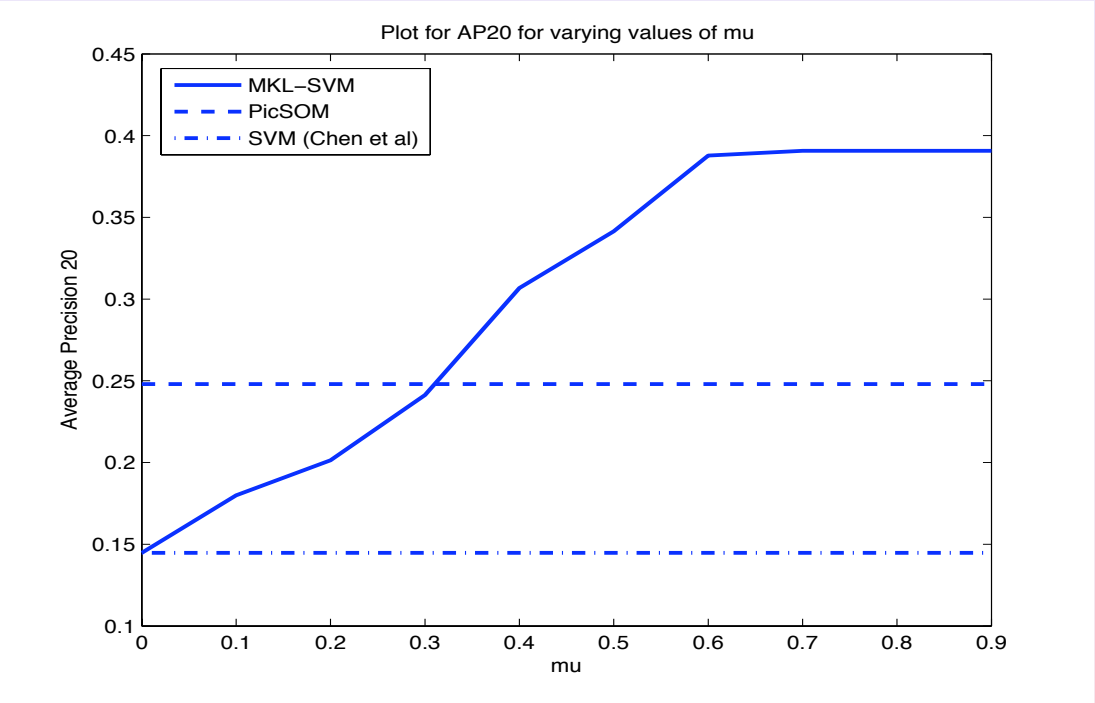

Figure: Average precision 20 against μ for cats

- Note that PicSOM uses negative examples

Effect of μ on retrieval

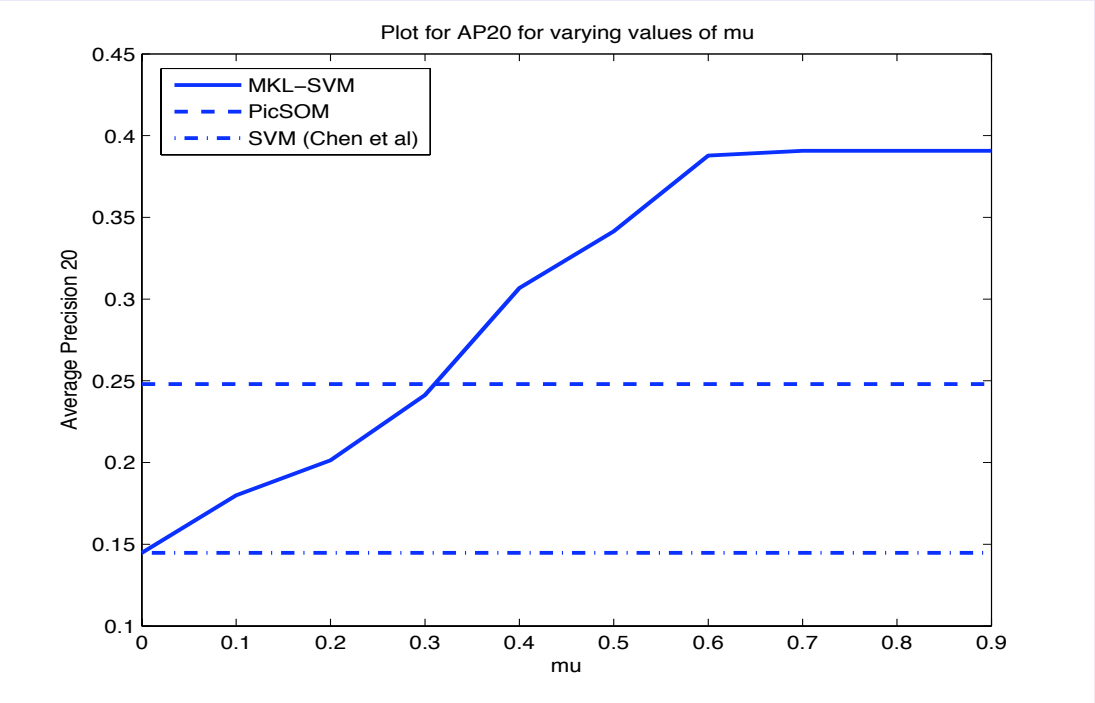

Figure: Average precision 20 against μ for cats

- Note that PicSOM uses negative examples

Including negative examples

- Included negatives by negating features, i.e. negating kernel entries between differently labelled images

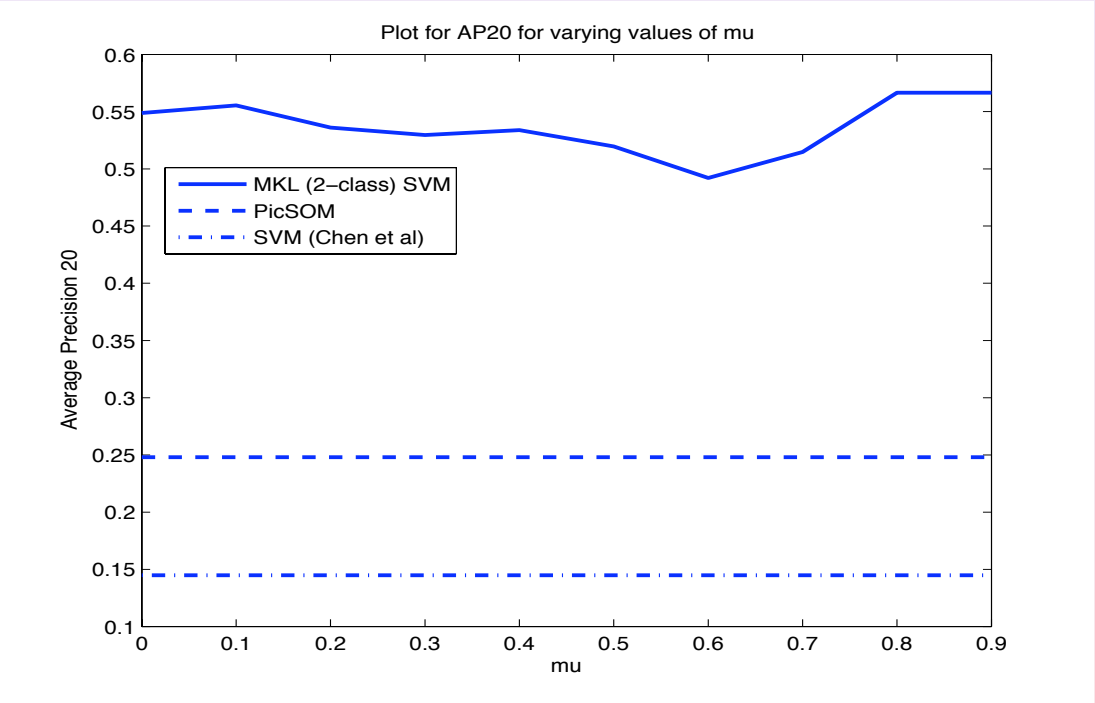

Figure: Average precision 20 against μ for cats

Complete precision/recall curves

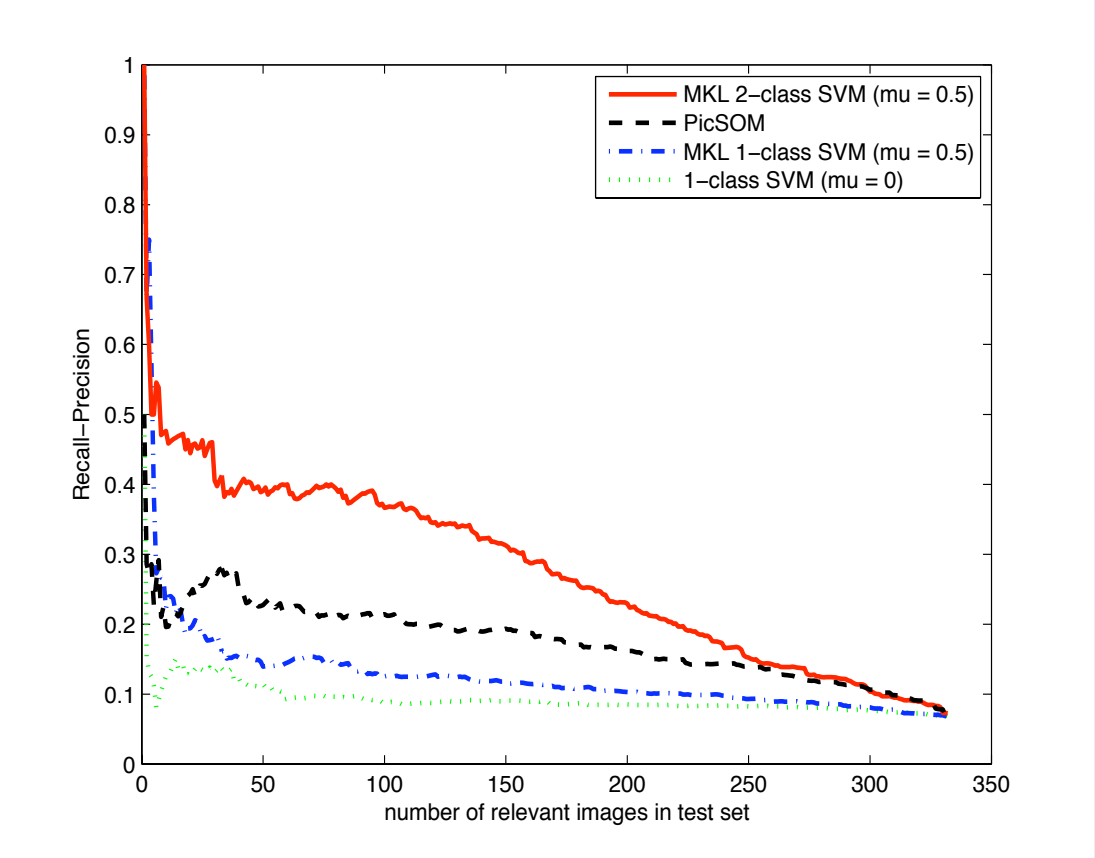

Figure: Precision/recall curve for cats

Average Precision scores

Obj.	MKL 2-class (μ : 0.5)			MKL 1-class (μ : 0.5)			1-class SVM		PicSOM	
	AP20	AP50	#ker	AP20	AP50	#ker	AP20	AP50	AP20	AP50
Cat	**0.52**	**0.46**	3	0.34	0.24	2	0.14	0.13	0.25	0.25
Cow	**0.29**	**0.20**	5	0.17	0.14	3	0.14	0.12	0.25	**0.20**
Dog	**0.37**	**0.36**	11	0.11	0.13	2	0.11	0.12	0.28	0.28

Outline

Conclusions

- Considered CBIR task in which learning the search metric corresponds to learning the kernel
- In 1-class MKL don't get variable sparsity by varying C
- Flexible mix of 1-norm and 2-norm regularisation gives natural control of sparsity with good performance against PicSOM and 1-class SVM on VOC cats
- Using negative (non-relevant) examples improves performance.
- SOM uses density learning to weight metrics – should compare with same approach in kernel methods

Conclusions

- Considered CBIR task in which learning the search metric corresponds to learning the kernel
- In 1-class MKL don't get variable sparsity by varying C
- Flexible mix of 1-norm and 2-norm regularisation gives natural control of sparsity with good performance against PicSOM and 1-class SVM on VOC cats
- Using negative (non-relevant) examples improves performance.
- SOM uses density learning to weight metrics – should compare with same approach in kernel methods

Conclusions

- Considered CBIR task in which learning the search metric corresponds to learning the kernel
- In 1-class MKL don't get variable sparsity by varying C
- Flexible mix of 1-norm and 2-norm regularisation gives natural control of sparsity with good performance against PicSOM and 1-class SVM on VOC cats
- Using negative (non-relevant) examples improves performance.
- SOM uses density learning to weight metrics – should compare with same approach in kernel methods

Conclusions

- Considered CBIR task in which learning the search metric corresponds to learning the kernel
- In 1-class MKL don't get variable sparsity by varying C
- Flexible mix of 1-norm and 2-norm regularisation gives natural control of sparsity with good performance against PicSOM and 1-class SVM on VOC cats
- Using negative (non-relevant) examples improves performance.
- SOM uses density learning to weight metrics – should compare with same approach in kernel methods

Conclusions

- Considered CBIR task in which learning the search metric corresponds to learning the kernel
- In 1-class MKL don't get variable sparsity by varying C
- Flexible mix of 1-norm and 2-norm regularisation gives natural control of sparsity with good performance against PicSOM and 1-class SVM on VOC cats
- Using negative (non-relevant) examples improves performance.
- SOM uses density learning to weight metrics – should compare with same approach in kernel methods