1 Title:
Learning Transferable Features With Deep Adaptation Networks

2 Summary:
This paper addresses unsupervised domain adaptation in the context of deep neural networks by minimizing Multi-Kernel MMD between source and target distributions at multiple higher/task-specific layers of the DNN along with empirical risk. Further, an optimal kernel selection mechanism is proposed to maximize capacity while minimizing Type II error. These two objectives are alternatively optimized and experiments are presented on Office-31, Office-10 and Caltech-10 datasets against various benchmarks and ablations.

3 Strengths:
i) Domain alignment is performed at multiple layers instead of a single layer. This is well suited for deep networks and works well even when multiple layers learn source domain specific features pre-adaptation.
ii) O(n) approximation is presented for kernel MMD computation which is useful to reduce training time, especially for deeper networks.
iii) Multiple kernels of different band-width are combined and an optimal kernel is learnt. This reduces the sensitivity to kernel choice.
iv) Experiments are well-designed to highlight the effect of each proposition over baseline.

4 Weaknesses:
i) The experimental results overall only show a slight improvement over regular CNN (70.1% vs 72.9% average on Office-31).
ii) The training process is complicated and compute-intensive involving alternating a mini-batch SGD step and a QPP step for kernel selection.

5 Analysis of Experiments:
i) The proposed model shows good improvement over CNN for a few combinations of source and target domains (e.g. Amazon->Webcam: 61.6% to 68.5%), while on most other combinations the improvement is marginal.
ii) CNN without adaptation shows huge improvement over shallow networks (27.8% to 70.1% on average)
iii) Multi-layer adaptation outperforms single layer adaptation and multi-kernel outperforms single kernel for almost all the tasks.
iv) T-SNE visualizations show improvement in discriminativeness and domain alignment.

6 Possible Extensions:
Since we have a kernel, we can find which image are close, across domains, by evaluating the kernel. Using this we can frame loss functions such as, a) Multiple layers should not give contradicting information. If layer l deems images 1 and 2 are similar, say, $K(\phi_l(x1), \phi_l(x2)) > 0.5$, then layer $l+1$ should give similar output. b) Multi-task Learning: If we have two different tasks with labels in source domain, then we can impose an additional loss that image pairs deemed close for task 1 should also be deemed close for task 2 and vice-versa.