Domain Adaptation with Domain Specific Class Means Classifiers

Gabriela Csurka, Boris Chidlovskii and Florent Perronnin

Xerox Research Centre Europe, France

TASK-CV, Friday 12th September 2014

DA methods transforming the feature space

- Using unsupervised transformation between domains:
  - generally based on PCA projections (Gopalan *et al*. ICCV11, Gong *et al*. CVPR12, Fernando *et al*. ICCV13, Baktashmotlagh *et al*. ICCV13)
- Learning transformation by exploiting class labels:
  - generally based on metric learning (Zha *et al*. IJCAI09, Saeko *et al*. ECCV10, Kulis *et al*. CVPR11, Hoffman *et al*. ECCV12)

In addition, DA methods might exploit unlabeled target instances (*e.g*. Duan *et al*. CVPR09, Saha *et al*. ECML11, Tomassi and Caputo ICCV13).

Our contribution

Exploiting class labels to learn the transformation:

- we propose the Domain Specific Class Means Classifier[1] (DSCM) that extends the Nearest Class Mean (NCM) classifier to DA by considering domain-specific class means and weights.

Exploiting unlabeled target instances to adapt it to the target:

- we propose the Self-adaptive Metric Learning Domain Adaptation[2] (SaMLDa) framework, that iteratively refines the metric using increasing target set and predicted labels.

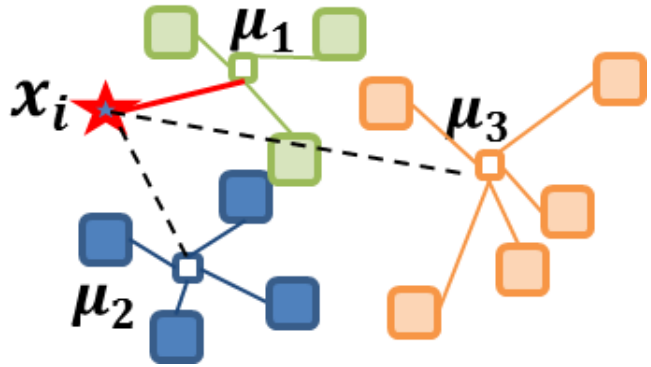
[1] Inspired by the NCMC of Mensink *et al*. PAMI13

[2] Inspired by the NBNN of Tomassi and Caputo ICCV13,

Outline

1. Domain Specific Class Means Classifier
2. Self-adaptive Metric Learning for Domain Adaptation
3. Experimental results
4. Conclusion

The Nearest Class Mean (NCM) classifier[3]



The NCM assigns an image to the closest class mean:

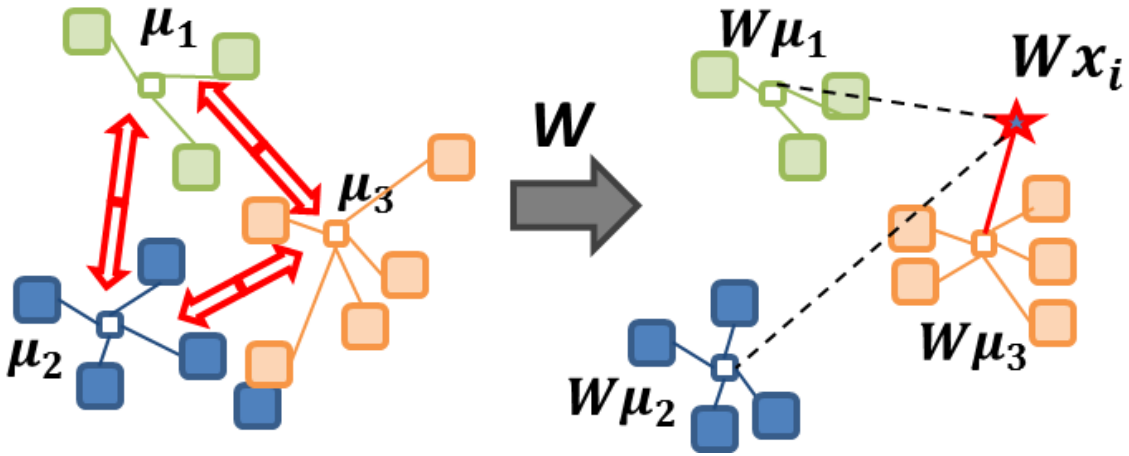
$$\boldsymbol{\mu}_c = \frac{1}{|\{\boldsymbol{x}_i | y_i = c\}|} \sum_{\boldsymbol{x}_i \in \{\boldsymbol{x}_i | y_i = c\}} \boldsymbol{x}_i$$

Can be seen as the posterior of a GMM with $w_c = \frac{1}{N_c}$ and $\boldsymbol{\Sigma} = \boldsymbol{I}$:

$$p(c|\boldsymbol{x}_i) = \frac{w_c p(\boldsymbol{x}_i|c)}{\sum_{c'=1}^{N_c} w_{c'} p(\boldsymbol{x}_i|c')} = \frac{w_c \mathcal{N}(\boldsymbol{x}_i, \boldsymbol{\mu}_c, \boldsymbol{I})}{\sum_{c'=1}^{N_c} w_{c'} \mathcal{N}(\boldsymbol{x}_i, \boldsymbol{\mu}_{c'}, \boldsymbol{I})}$$

[3] **T. Mensink, J. Verbeek, F. Perronnin and G. Csurka, Distance-based image classification: Generalizing to new classes at near zero cost. PAMI 35(11), 2013**

ML for NCM[4]



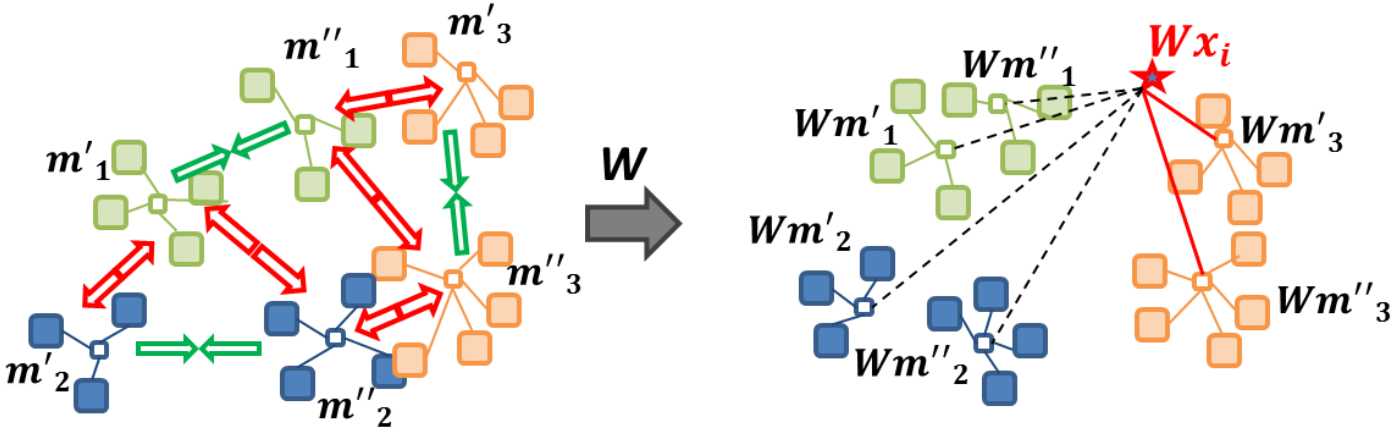
Learning a projection $\boldsymbol{W}$ that maximizes the NCM accuracy:

$$p(c|\boldsymbol{x}_i) = \frac{w_c \mathcal{N}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{\mu}_c, \boldsymbol{\Sigma})}{\sum_{c'} w_{c'} \mathcal{N}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{\mu}_{c'}, \boldsymbol{\Sigma})} = \frac{\exp\left(-\frac{1}{2} d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_c)\right)}{\sum_{c'} \exp\left(-\frac{1}{2} d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_{c'})\right)}$$

where $d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_c) = \|\boldsymbol{W}(\boldsymbol{x}_i - \boldsymbol{\mu}^c)\|^2$ and $\boldsymbol{\Sigma} = (\boldsymbol{W}^\top \boldsymbol{W})^{-1}$.

[4] **T. Mensink *et al.*, Distance-based image classification, PAMI 2013**

The Nearest Class Multiple Centroids (NCMC)[5]



It extends the NCM by considering multiple centroids $\boldsymbol{m}_c^j$ per class.

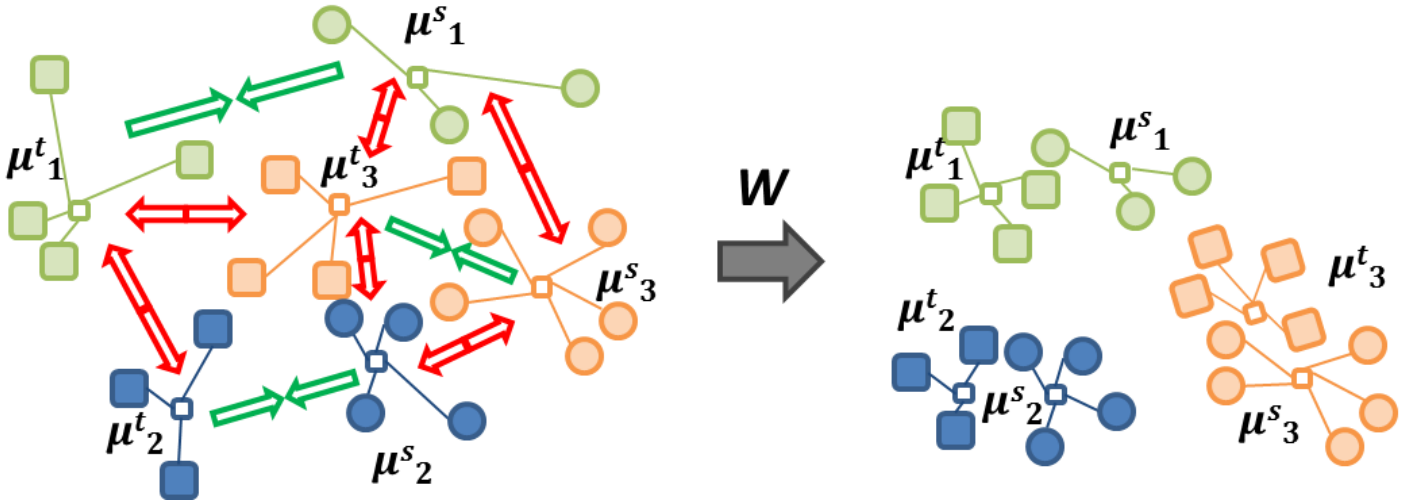
- The model becomes a mixture of GMMs:

$$p(c|\boldsymbol{x}_i) = \frac{w_c \sum_j w_j \mathcal{N}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{m}_c^j, \boldsymbol{\Sigma})}{\sum_{c'} w_{c'} \sum_j w_j \mathcal{N}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{W}\boldsymbol{m}_{c'}^j, \boldsymbol{\Sigma})},$$

with $w_c = \frac{1}{N_c}$ and $w_j = \frac{1}{N_j}$ and shared $\boldsymbol{\Sigma} = (\boldsymbol{W}^\top \boldsymbol{W})^{-1}$.

[5] **T. Mensink *et al.*, Distance-based image classification, PAMI 2013**

Domain Specific Class Means (DSCM)



- Mixture of GMM:

$$p(c|\boldsymbol{x}_i) = \frac{\sum_d w_d \mathcal{N}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{W}\mu_d^c, \boldsymbol{\Sigma})}{\sum_{c'} \sum_d w_d \mathcal{N}(\boldsymbol{W}\boldsymbol{x}_i, \boldsymbol{W}\mu_d^{c'}, \boldsymbol{\Sigma})} = \frac{\sum_d w_d \exp\left(-\frac{1}{2} d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_d^c)\right)}{\sum_{c'} \sum_d w_d \exp\left(-\frac{1}{2} d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_d^{c'})\right)}$$

with

- domain-specific class means μ_d^c, instead of clustering.
- domain-specific weights w_d, instead of $\frac{1}{N_d}$.

The domain specific weights

Allowing to express different importance of the source domains.

These weights can be:

- manually fixed (using prior knowledge),
- learned (*e.g.* cross validated)
- set from the training data, *e.g.* as the:
  - distance between a source and the target domain
    - Target Density Around Source, Fernando *et al.* ICCV13,
  - NCM classification accuracies
    - computed on the labeled target set given a source.

Outline

1. Domain Specific Class Means Classifier

2. Self-adaptive Metric Learning for Domain Adaptation

3. Experimental results

4. Conclusion

Self-adaptive Metric Learning for DA[6]

The idea is to use the DSCM classifier to select for each class:

- unlabeled target instances to be added:
  - $\boldsymbol{x}_i^t$ for which $p(c^*|\boldsymbol{x}_i^t) - p(c^\dagger|\boldsymbol{x}_i^t)$ is the largest, c^* being the first and $c^\dagger$ the second highest class prediction.
- the most ambiguous source examples to be removed:
  - $\boldsymbol{x}_j^s$ for which $p(c^*|\boldsymbol{x}_j^s) - p(c^\dagger|\boldsymbol{x}_j^s)$ is the smallest.

Then $\boldsymbol{W}$ is iteratively refined with a metric learning (ML) approach and the updated training set.

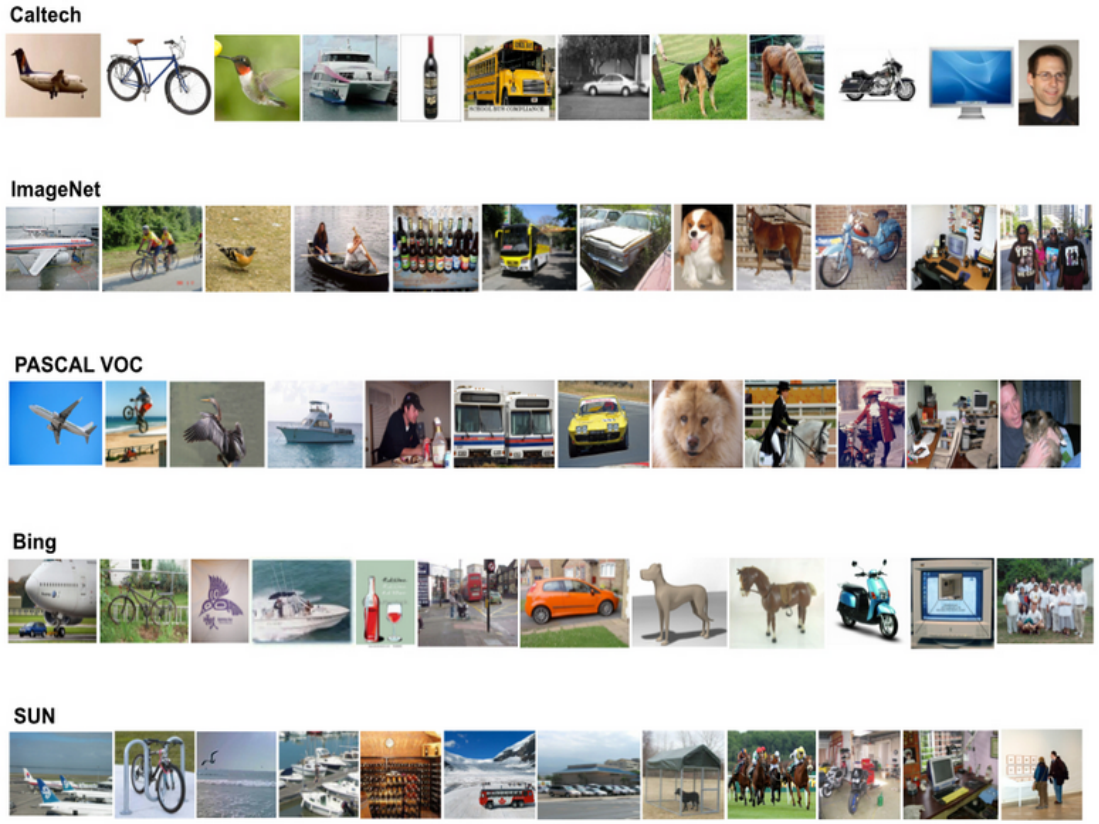
[6] **Inspired by "Frustratingly easy NBNN domain adaptation, T. Tommasi and B. Caputo, ICCV13".**

Outline

1. Domain Specific Class Means Classifier
2. Self-adaptive Metric Learning for Domain Adaptation
3. Experimental results
4. Conclusion

The ImageCLEF'14 DA Challenge

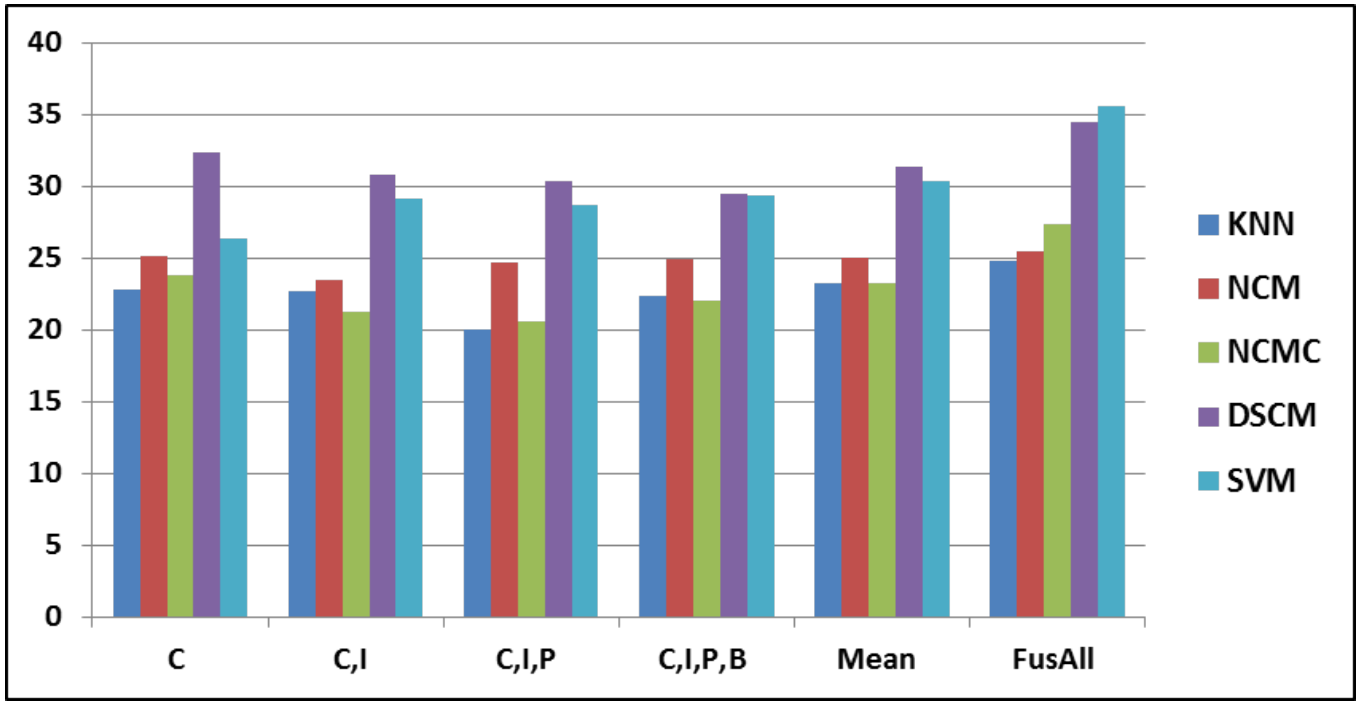
- Sources:
  - Caltech (C)
  - ImageNet (I)
  - Pascal (P)
  - Bing (B)
- Target:
  - SUN (S)



Experimental setup

- 12 common classes:
  - airplane, bike, bird, boat, bus, car, ...
- Only BOV features were provided (no access to images)
  - 600 labeled features from each source
  - 60 labeled and 600 unlabeled features from target
- 11 fold cross validation scheme
  - varying the labeled target set
- Different source combinations:
  - $\{C\}, \{I\}, \{P\}, \{B\}, \{C, I\}, \{C, P\}, \{C, B\}, \{I, P\}, \{I, B\}, \{P, B\}, \{C, I, P\}, \{C, I, B\}, \{C, P, B\}, \{I, P, B\}, \{C, I, P, B\}$
- Report also results for:
  - **Mean** - average over all configurations' results
  - **FuseAll** - late fusion of all configurations' results

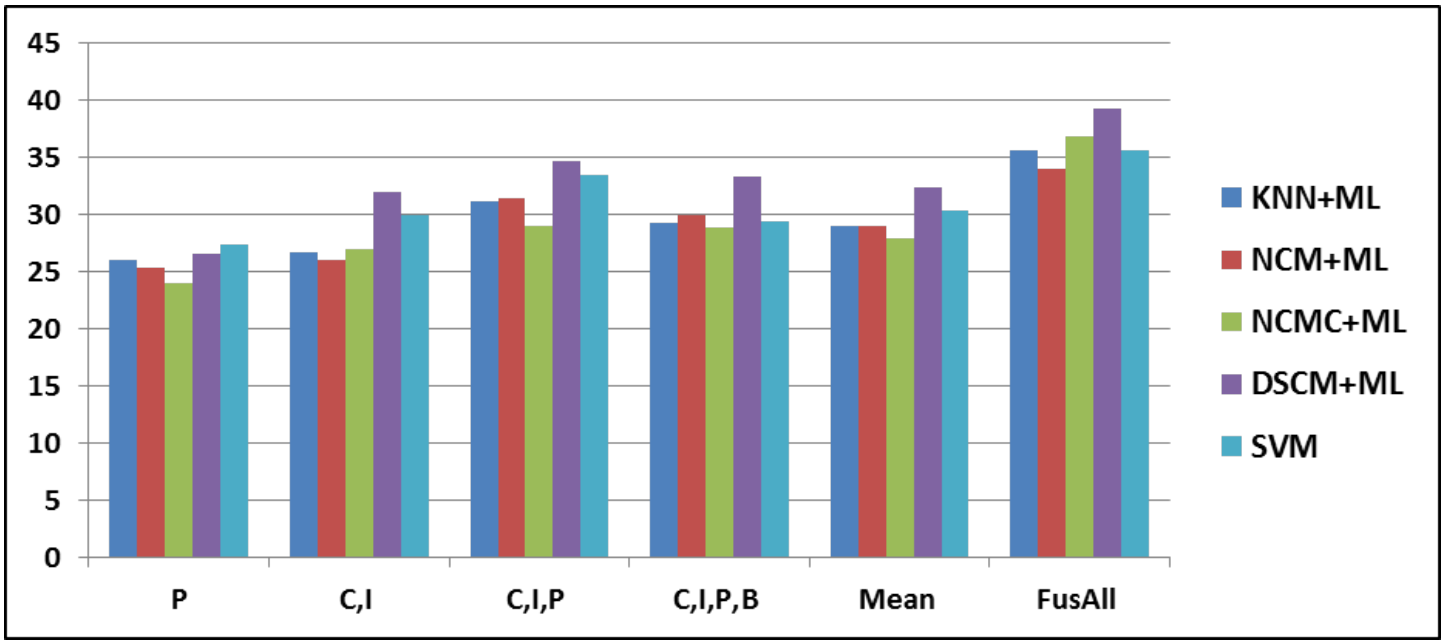
Correct prediction rates with $W = I$



DSCM, even without any learning, is suitable for domain adaptation:

- significantly outperforms KNN, NCM and NCMC
- outperforms SVM on most configurations
- requires only distances to domain-specific class means.

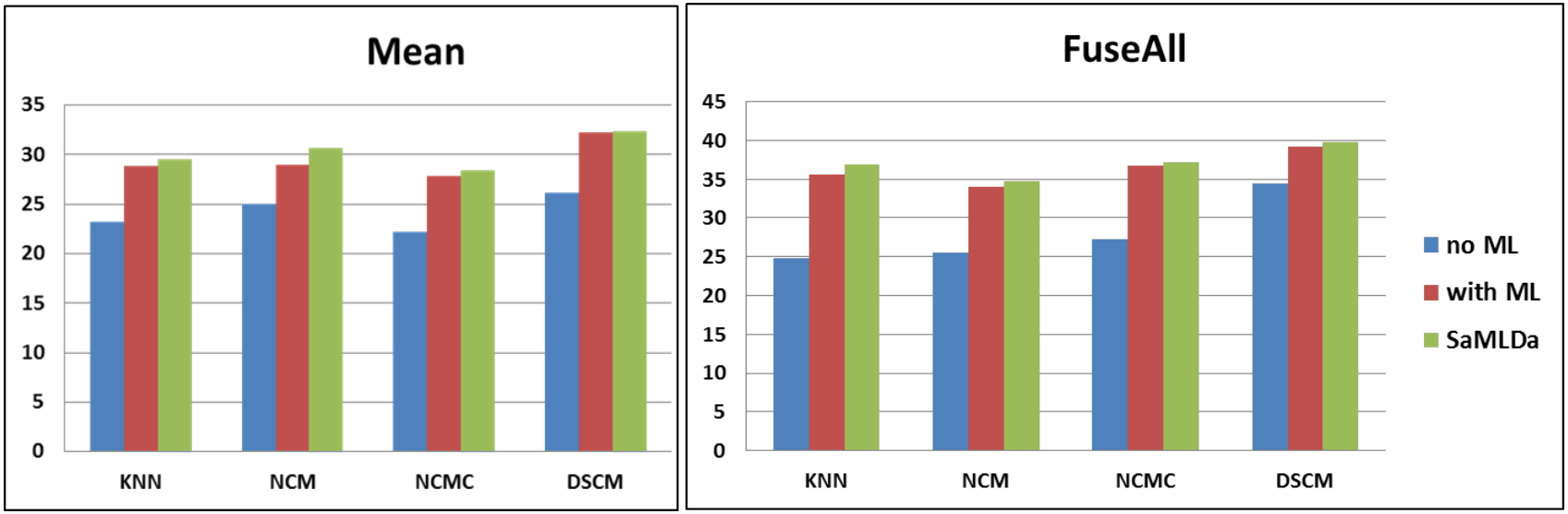
Result with learned W (metric learning)s



Metric learning allows to further improve the results:

- DSCM+ML outperforms KNN+ML, NCM+ML and NCMC+ML
  - ML uses the corresponding objectives;
- DSCM+ML outperforms in general SVM
  - especially if multiple sources are used.

Result with SaMLDa



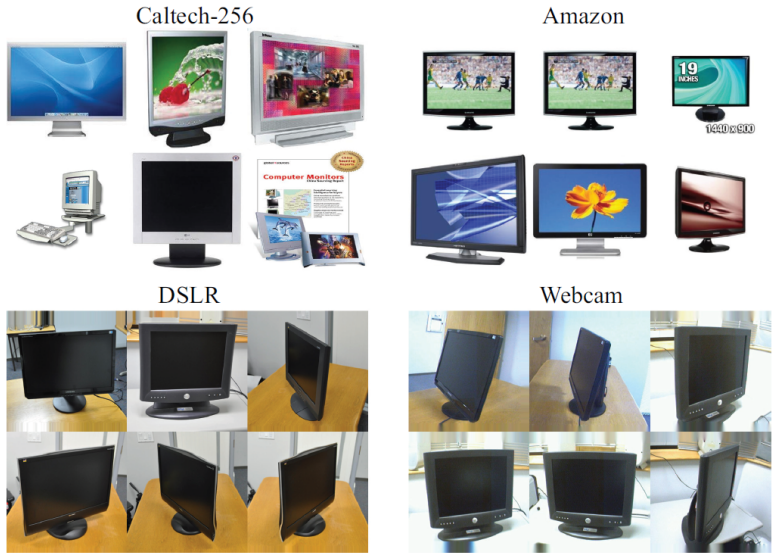
SaMLDa allows improvement for other ML approaches.

- SaMLDa outperforms ML in all cases
- DSCM+SaMLDa performs the best

Office Caltech-10

10 common classes from:

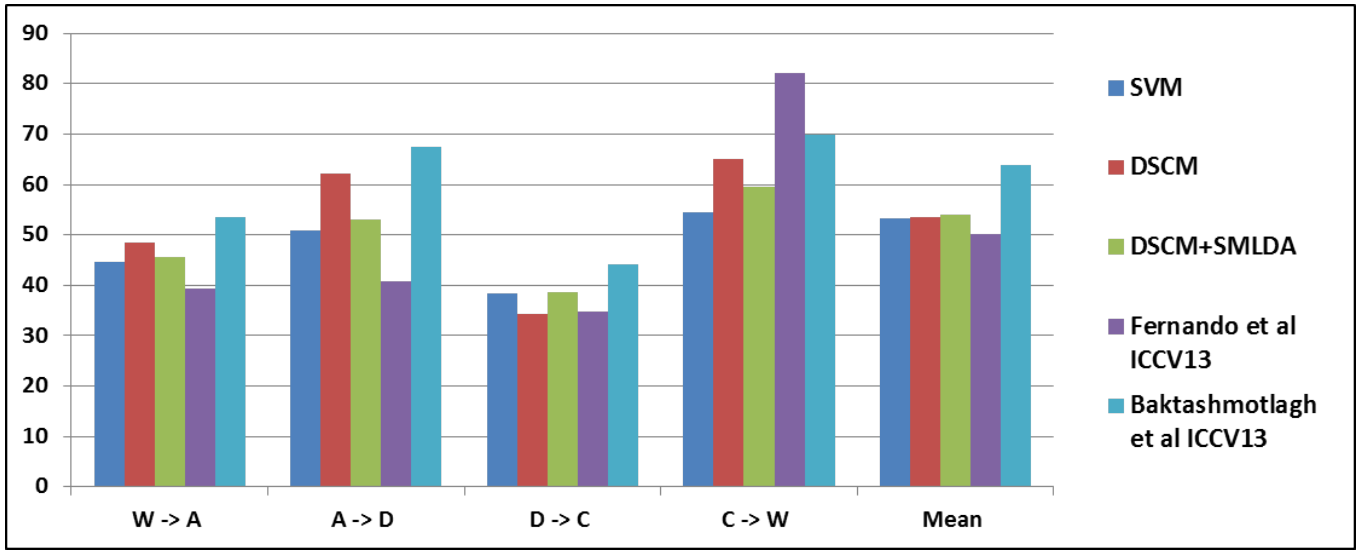
- Amazon (A)
- Caltech (C)
- DSLR (D)
- Webcam (W)



OffCalSS

- the semi-supervised setup as in Gong *et al*. CVPR12
  - 8 or 20 images selected from each class for training
    - 8 for D or W and 20 when A or C
  - adding 3 target images per class
  - repeating the experiment 20 times

Results on OffCalSS



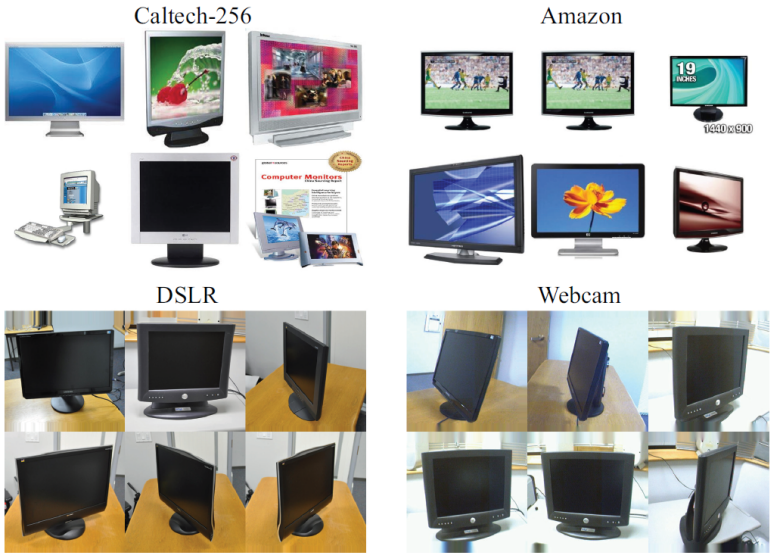
DSCM(+SaMLDa) performs less well if 1 source and few data

- The DIP+CC of Baktashmotlagh et al ICCV13 performs the best
  - it learns ***W*** by minimizing on the Grassman manifold
    - the Maximum Mean Discrepancy between 2 domains,
    - as well as intra-class distances;
  - then non-linear SVMs are trained in the projected space.

Office Caltech-10

10 common classes from:

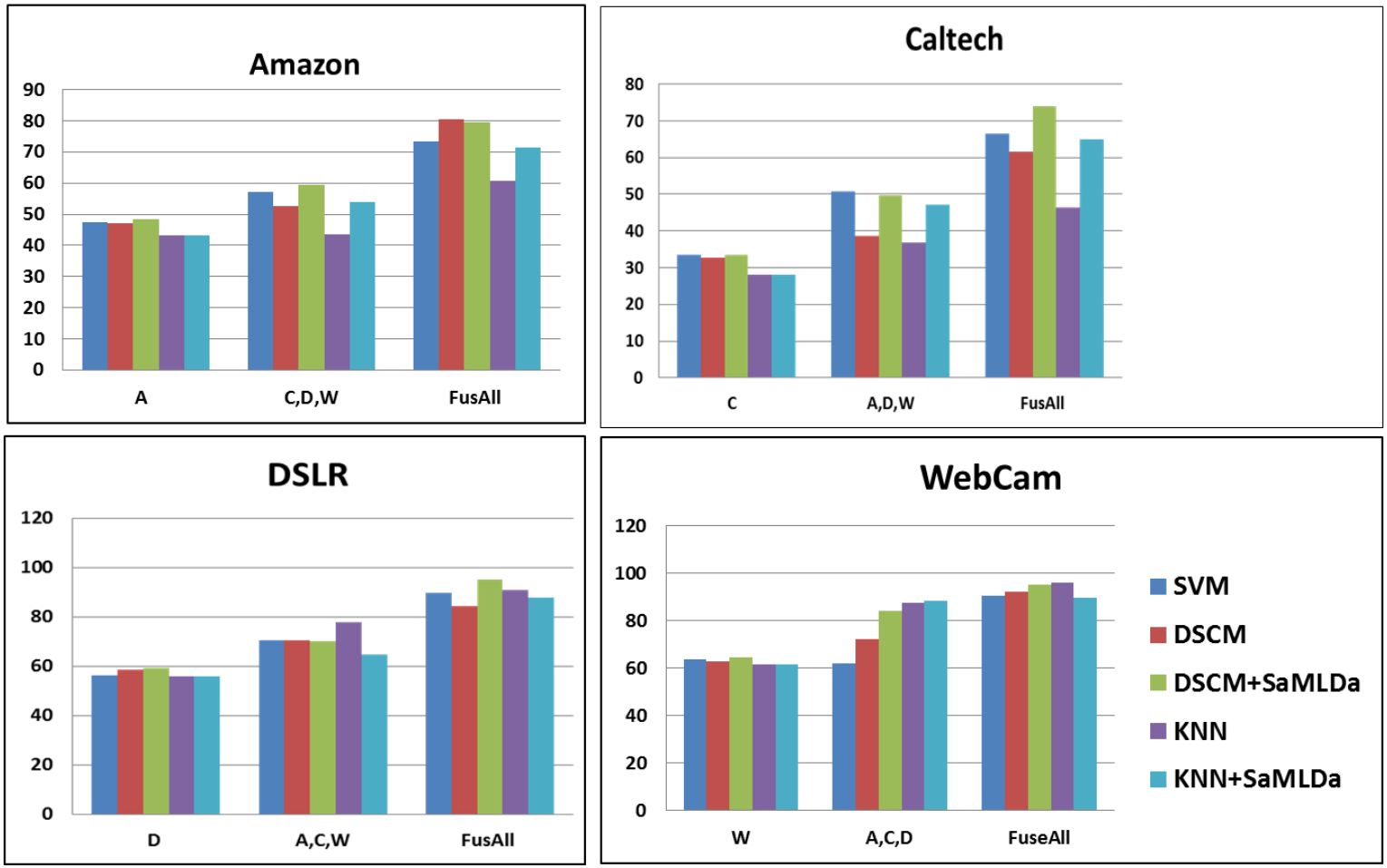
- Amazon (A)
- Caltech(C)
- DSLR (D)
- Webcam (W)



OffCalMS

- considering multiple sources, similar to ICDA1 setup
  - one domain as target, the others as source
  - using all training data from the sources
  - adding randomly 3 target images per class
  - repeating the experiment 10 times

Results on OffCaIMS



Best results obtained when we do adaptation for each source configuration and merge all predictions.

Outline

1. Domain Specific Class Means Classifier
2. Self-adaptive Metric Learning for Domain Adaptation
3. Experimental results
4. Conclusion

Conclusion

We proposed for domain adaptation:

- DSCM, a simple and efficient method
  - with corresponding metric learning (ML) to improve the DSCM accuracy in the projected space.
- A Self-adaptive Metric Learning for Domain Adaptation (SaMLDa) framework that:
  - exploits unlabeled target samples to refine the metric;
  - can be applied beyond DSCM to other ML.

Back-up slides

Domain Specific Class Means (DSCM) Classifier

An image x_i is assigned to $c^* = \text{argmax}_c\, p(c|\boldsymbol{x}_i)$ where:

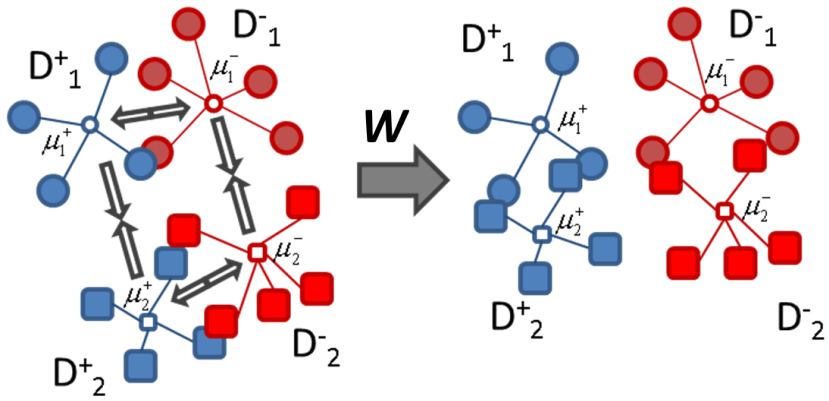
$$p(c|\boldsymbol{x}_i) = w_c \frac{\sum_d w_d \exp\left(-\frac{1}{2} d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_d^c)\right)}{\sum_{c'} w_{c'} \sum_d w_d \exp\left(-\frac{1}{2} d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_d^{c'})\right)}$$

- $d_{\boldsymbol{W}}(\boldsymbol{x}_i, \boldsymbol{\mu}_d^c) = \|\boldsymbol{W}(\boldsymbol{x}_i - \boldsymbol{\mu}_d^c)\|^2$, is the squared Euclidean distance in the space projected by $\boldsymbol{W}$,
- $\boldsymbol{\mu}_d^c$ are the domain specific class means:

$$\boldsymbol{\mu}_d^c = \frac{1}{|\mathcal{D}_d^c|} \sum_{x_j \in \mathcal{D}_d^c} \boldsymbol{x}_j, \quad \text{where } \mathcal{D}_d^c = \{x_i | y_i = c, x_i \in \mathcal{D}_d\}$$

- w_d are the domain specific weights.

Metric Learning for DSCM



Find $\boldsymbol{W}$, that maximizes $\ln p(c = y_i|\boldsymbol{x}_i)$ over the training set:

$$\mathcal{L} = \sum_{x_i \in \{\cup \mathcal{D}_d\}} \ln p(c = y_i|\boldsymbol{x}_i) \sum_{x_i \in \{\cup \mathcal{D}_d\}} \left[\ln \sum_d w_d p(\boldsymbol{x}_i|y_i, d) - \ln Z_i \right]$$

using SGD to update $\boldsymbol{W}$ with a fixed rate and the gradient at $\boldsymbol{x}_i$:

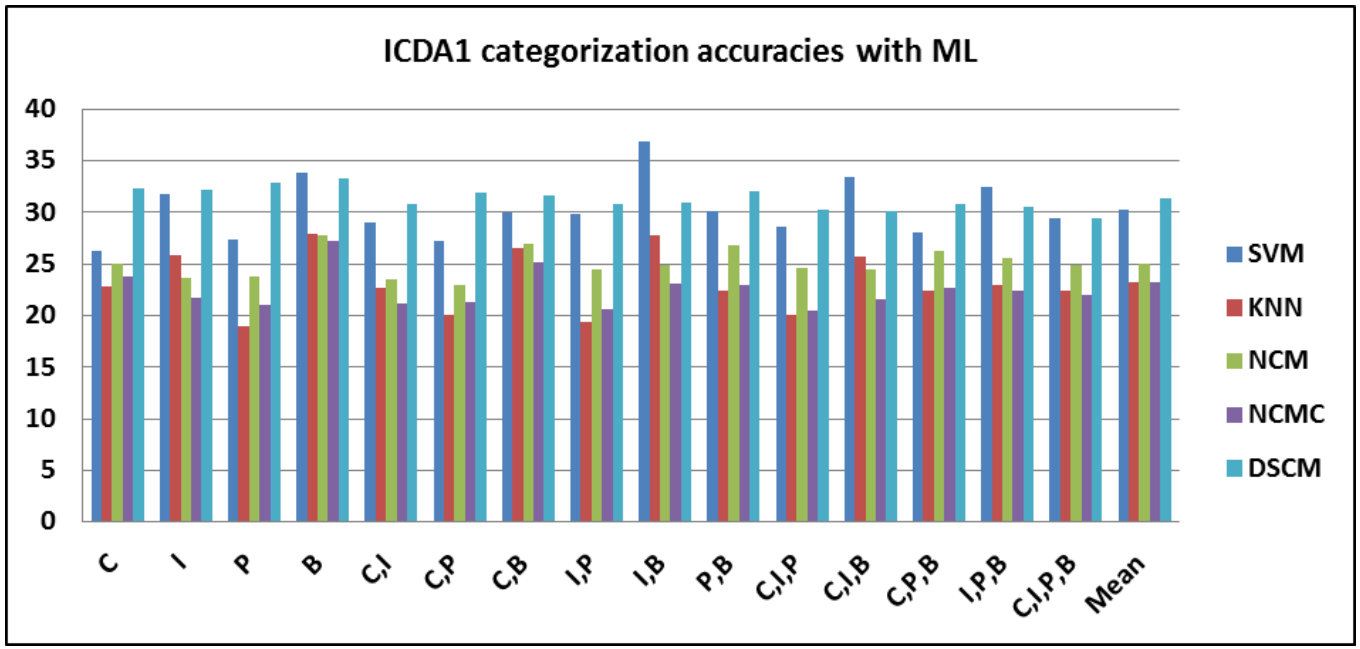
$$\nabla_W \mathcal{L}(\boldsymbol{x}_i) = \sum_{c'} \sum_d \left(\frac{w_d p(\boldsymbol{x}_i|c', d)}{Z_i} - [\![c' = y_i]\!] \frac{w_d p(\boldsymbol{x}_i|c', d)}{p(\boldsymbol{x}_i|c')} \right) \boldsymbol{W}(\boldsymbol{\mu}_d^{c'} - \boldsymbol{x}_i)(\boldsymbol{\mu}_d^{c'} - \boldsymbol{x}_i)^\top$$

The SaMLDa algorithm

Given an initial multi-domain training set X_0, and a ML component $f_{\boldsymbol{W}}$, compute $\boldsymbol{W}_1 = f_{\boldsymbol{W}}(X_1, \boldsymbol{W}_0, \boldsymbol{w}_d^0)$, where $\boldsymbol{W}_0$ is initialized with PCA.

- **For** $r = 1, \ldots, N_R$, **do**
  1. Set $X_r = X_{r-1}$ and $w_d^r = w_d^{r-1}$.
  2. Compute domain-specific class means μ_d^c.
  3. Optionally, update $\boldsymbol{w}_d^r$ using *TDAS* or NCM with $\boldsymbol{W}_r$.
  4. For each $\boldsymbol{x}_i$ and c compute $p(c_j|\boldsymbol{x}_i)$ with DSCM.
     - For each class c_j, add unlabeled target $\boldsymbol{x}_i^t$ for which $p(c^*|\boldsymbol{x}_i^t) - p(c^\dagger|\boldsymbol{x}_i^t)$ is the largest.
     - For each class c_j, remove source $\boldsymbol{x}_j^s$ for which $p(c^*|\boldsymbol{x}_j^s) - p(c^\dagger|\boldsymbol{x}_j^s)$ is the smallest.
  5. Refine $\boldsymbol{W}_{r+1} = f_{\boldsymbol{W}}(X_r, \boldsymbol{W}_r, \boldsymbol{w}_d^r)$.
  6. Stop if stopping criteria is met, otherwise continue.

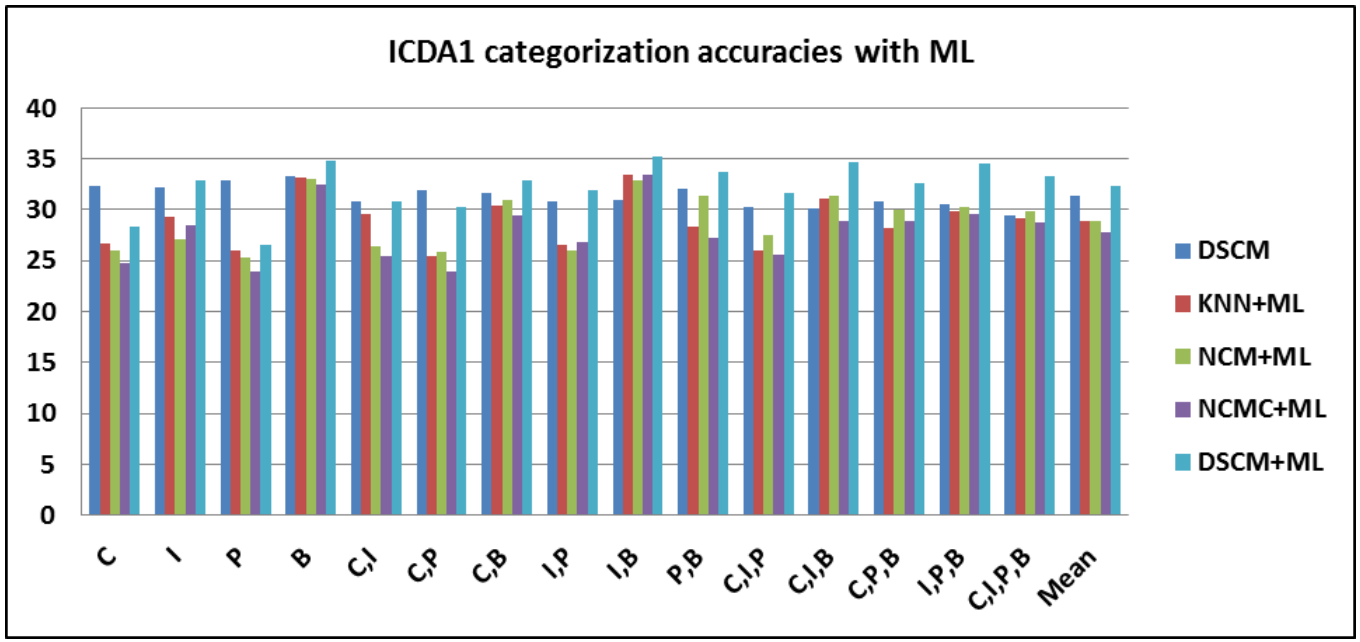
Result with $W = I$



- DSCM outperforms significantly KNN, NCM and NCMC
- It outperforms for several configuration (and in average) SVM.

DSCM even without any learning, is suitable for domain adaptation.

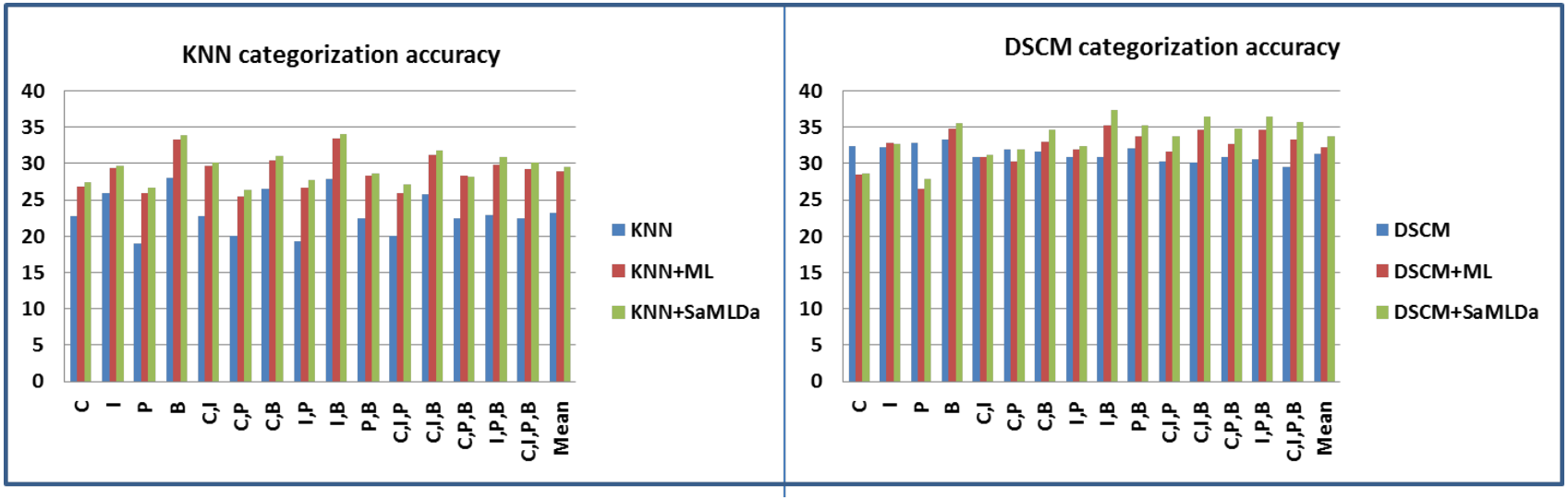
Result with ***W*** learned by metric learning



- DSCM outperforms in KNN+ML, NCM+ML and NCMC+ML
  - ***W*** is learned using SGD to optimize the corresponding (LMNN, NCM or NCMC) objectives.
- DSCM+ML outperforms in general and on average DSCM.

Metric learning allows to further improve the results.

Result with SaMLDa

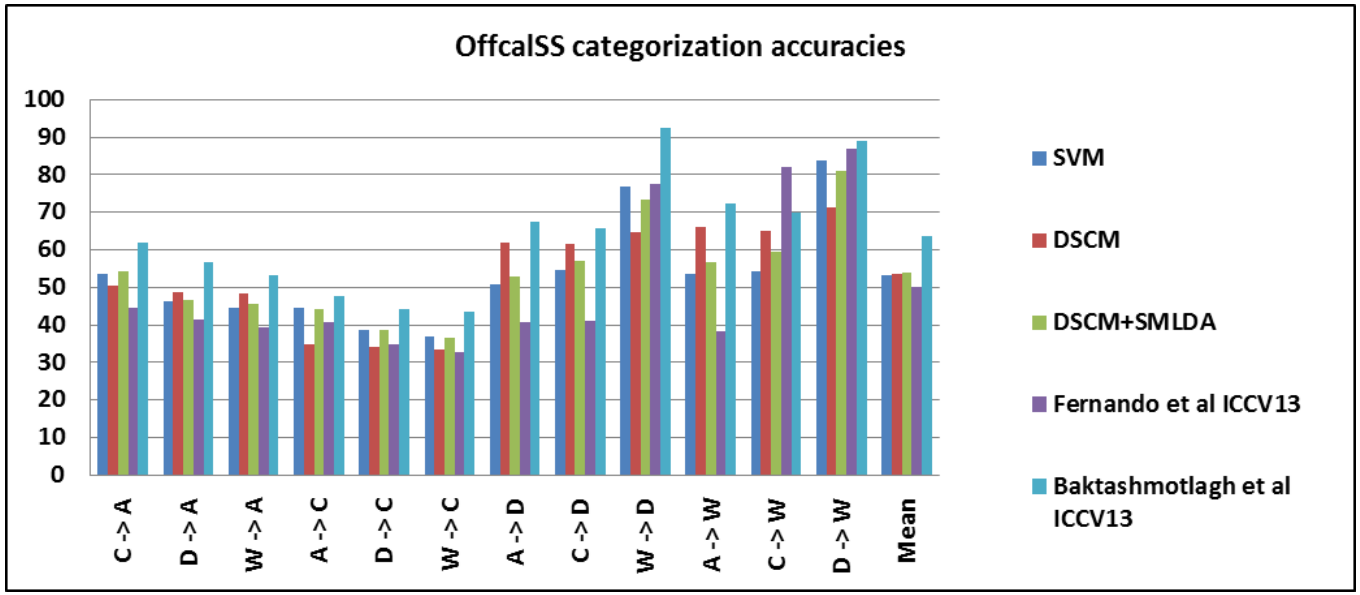


- KNN+SaMLDa outperforms KNN+ML
- NCM(C)+SaMLDa outperforms NCM(C)+ML (see paper)
- DSCM+SaMLDa outperforms DSCM+ML

SaMLDa allows improvement for other ML approaches.

DSCM+SaMLDa yields the best results

Results on OffCalSS

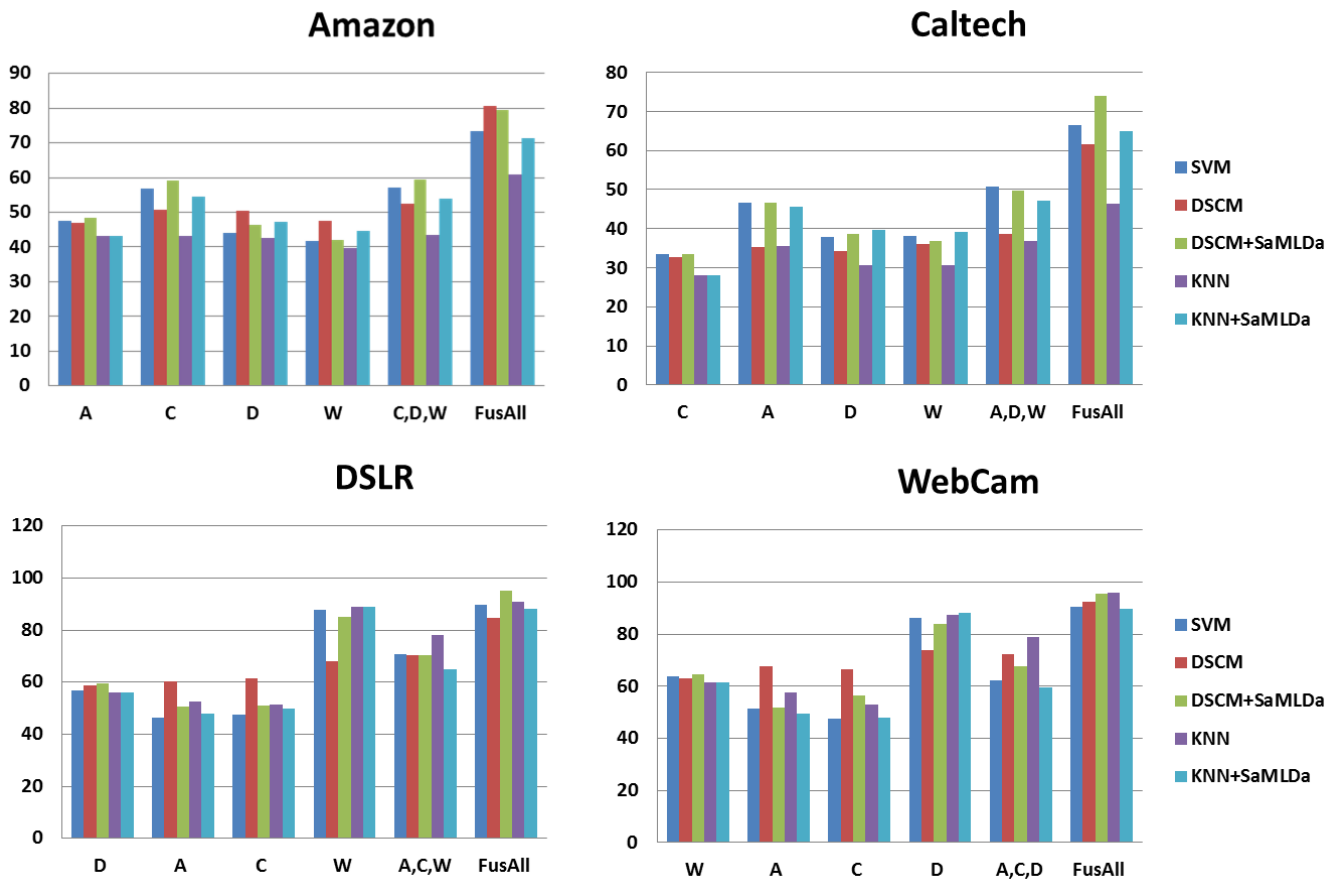


The DIP+CC of Baktashmotlagh et al ICCV13 performs the best

- they learn ***W*** by optimizing on the Grassman manifold the
  - Maximum Mean Discrepancy (MMD) between 2 domains
  - to which they add a term to encourage class clustering
- they train non-linear SVM in the projected space.

DSCM(+SaMLDa) performs less well if 1 source and few data

Results on OffCaIMS



Best results obtained when we do adaptation for each source combination and merge all predictions.