Assembling Heterogeneous Domain Adaptation Methods for Image Classification

Boris Chidlovskii, Gabriela Csurka and Shalini Gangwar

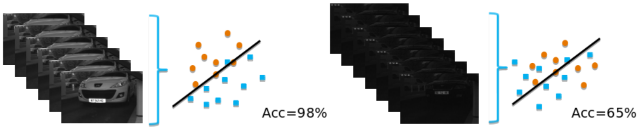
Xerox Research Centre Europe, France

ImageCLEF, 16th September 2014

Domain Adaptation At Xerox

Transportation, image-based solutions

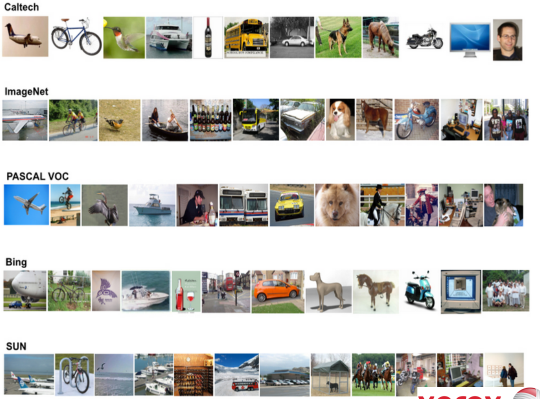
- Adapt learning components under data distribution change, without a costly re-annotation
- Changes caused by scene illumination, view angle, background
  - Daylight to night, from inside to outside
  - From one parking to another, other cameras, etc.



ImageCLEF'14 Domain Adaptation Challenge

Domain adaptation scenario:

- Multiple source domains
- Same labels between the source and the target domains
- Limited number of annotated data in the target domain



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

Challenge setup

- 12 common classes:
  - airplane, bike, bird, boat, bus, car, ...
- No access to images
- BOV features provided only
  - 600 labeled features from each source (C, I, P, B)
  - 60 labeled and 600 unlabeled features from target (S)
- Source and target domains are semantically relevant but different
- Target feature distribution changed between phases 1/2

Build a recognition system for target domain by leveraging the knowledge from source domains

Outline

1. Assembling Heterogeneous Methods
2. Domain Adaptation by Instance Transfer
3. Domain Adaptation by Space Transformation
4. Ensemble Methods
5. Evaluation Results
6. Conclusion

Domain adaptation methods

Instance Transfer

- Instance weighting in source domain (Dai *et al*. 2007, Xu 2012)
- Selecting landmarks in source domain (Gong 2013)

Feature Space Transformation

- Unsupervised transformation of domains
    - 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
    - based on metric learning (Zha *et al*. IJCAI09, Saeko *et al*. ECCV10, Kulis *et al*. CVPR11, Hoffman *et al*. ECCV12)
    - Some methods exploit unlabeled target instances (*e.g*. Duan *et al*. CVPR09, Saha *et al*. ECML11, Tomassi and Caputo ICCV13)

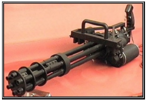
.

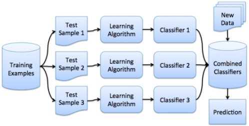
XRCE approach

- Individual methods
  - Brute force: SVM cross validation with all combinations
  - Instance Weighting: Instance transfer from sources to target domain using Boosting trick
  - Space transformation: metric learning-based domain adaptation to push together the same-class instances from different domains

.

- Ensemble techniques to aggregate individual predictions





Brute Force

- $N_{SC} = 2^N_S - 1 = 15$ source combinations SC_i,
- For each source combination SC_j:
  - concatenate the target train set $\mathcal{T}_l$ with sources SC_j
  - train SVM in a cross validation
- Multi-class SVM
  - one kernel and same parameters for all classes
- Binarised one-against-all SVM
  - The best classifier for each class c_j
  - A specific set of parameter values, kernels and source combinations
  - For an unseen sample $\mathbf{x}_i$, take the classifier with the highest confidence

$$\hat{y}_{\mathrm{bsvm}} = \underset{c_j \in Y}{\operatorname{argmax}} f^{c_j}_{\mathrm{bsvm}}(\mathbf{x}_i).$$

Outline

1. Assembling Heterogeneous Methods
2. Domain Adaptation by Instance Transfer
3. Domain Adaptation by Space Transformation
4. Ensemble Methods
5. Evaluation Results
6. Conclusion

Instance Transfer with AdaBoost

- Transfer AdaBoost is an extension of Adaboost to Transfer learning
- boost the accuracy of a weak learner by carefully adjusting the weights of training instances and to learn a classifier
- In TrAdaboost:
  - Target training instances are weighted as in AdaBoost
  - Source training instances are weighted differently
  - Wrongly predicted source instances are the most dissimilar
  - Their weights decrease to weaken their impact

Transfer Adaptive Boosting with one source

Require: Target training set $T_t = (X_t, Y)$; source training set $T_s = (X_s, Y)$; *Learner*; the number of iterations M.

Ensure: Target learner $f : X_t \to Y$.

Initial weights: $\mathbf{w}_T^1 = (w_{t_1}^1, \ldots, w_{t_{N_t}}^1)$, $\mathbf{w}_S^1 = (w_{s_1}^1, \ldots, w_{s_{N_s}}^1)$,

Set $\mathbf{w} = (\mathbf{w}_T, \mathbf{w}_S)$, $\beta = 1/(1 + 2\sqrt{\ln N_t/M})$ and $T = (T_t, T_s)$.

for $r = 1, \ldots, M$ **do**

    Normalize $\mathbf{w}^r = \mathbf{w}^r/|\mathbf{w}^r|$.

    Call *Learner* on the training set T with $\mathbf{w}^r$ to find $f_r : X \to Y$

    Calculate error of h_r on T_t :

$$\epsilon_r = min\left(\frac{1}{2}, \frac{1}{\sum_{i=1}^{N_t} w_{t_i}^r} \sum_{i=1}^{n} w_{t_i}^r \cdot [\![f_r(\mathbf{x}_i^t) \neq y_i]\!]\right).$$

    Set $\beta^r = 1/2 \log((1 - \epsilon_r)/\epsilon_r)$; $\Gamma^r = 2(1 - \epsilon_r)$.

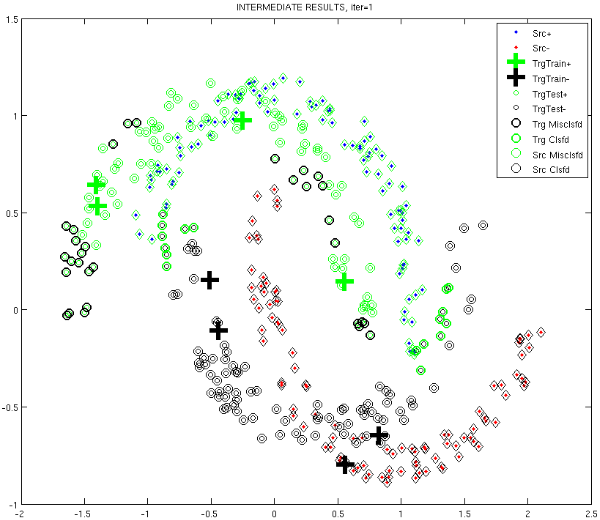
    Update the weight vectors:

$$\begin{aligned} w_{s_j}^{r+1} &= \Gamma^r w_{s_j}^r \exp(-\beta\, [\![f_r(\mathbf{x}_j^s) \neq y_j]\!]), \quad && \mathbf{x}_j^s \in X_s, \\ w_{t_i}^{r+1} &= w_{t_i}^r \exp(2\beta^r\, [\![f_r(\mathbf{x}_i^t) \neq y_i]\!]), && \mathbf{x}_i^t \in X_t. \end{aligned}$$

end for

Output the aggregated estimate $f_{\text{tra}}(\mathbf{x}) = \left(\sum_{r=1}^{M} \beta^r f_r(\mathbf{x})\right)$.

Transfer Adaptive Boosting: Two moons

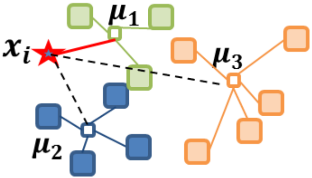


B. Chidlovskii et al, Assembling Heterogeneous DA

xerox

Outline

The Nearest Class Mean (NCM) classifier[1]



The NCM assigns an image to the closest class mean:

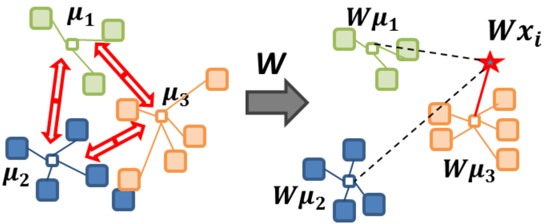
$$\mu_c = \frac{1}{|\{\mathbf{x}_i | y_i = c\}|} \sum_{\mathbf{x}_i \in \{\mathbf{x}_i | y_i = c\}} \mathbf{x}_i$$

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

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

[1] **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[2]



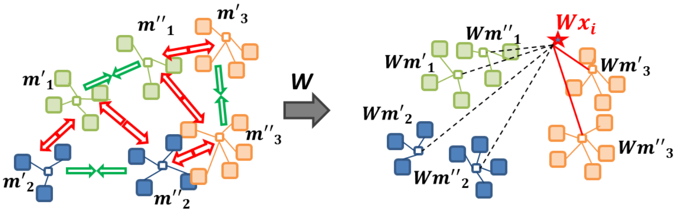
Learning a projection $\boldsymbol{W}$ that maximizes the NCM accuracy:

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

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

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

The Nearest Class Multiple Centroids (NCMC)[3]



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

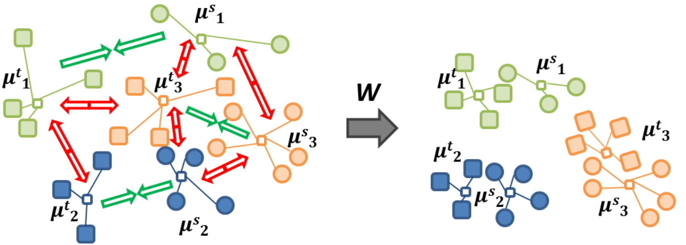
- The model becomes a mixture of GMMs:

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

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

[3] **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 $\frac{1}{N_d}$.

xerox

Outline

Heterogeneous set of classifiers

Combine outputs of multiple classifiers of 3 different types

- Pool F of classifiers $F = \{f_1, \ldots, f_{N_f}\}$, with class scores/ probabilities
- Unweighted majority voting (UMV)

$$c^* = \operatorname{argmax}_{c \in Y} \sum_{f_k \in F} [\![g_k(f_k, \mathbf{x}_i^t) = c]\!]$$

  - In probabilistic setting, the class with the highest probability:

$$c^* = \operatorname{argmax}_{c \in Y} \sum_{f_k \in F} P(y_i = c | f_k(\mathbf{x}_i^t))$$

- Weighting majority voting (WMV), weights proportional to classifier's accuracy

$$P(y_i = c | \mathbf{x}_i^t) = \frac{\prod_{c'' \in Y} P\big(y_i = c | g(f_k, \mathbf{x}_i^t) = c''\big)}{\sum_{c' \in Y} \prod_{c'' \in Y} P\big(y_i = c' | g(f_k, \mathbf{x}_i^t) = c''\big)}$$

Outline

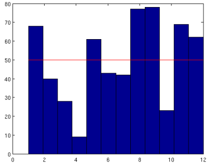
1. Assembling Heterogeneous Methods
2. Domain Adaptation by Instance Transfer
3. Domain Adaptation by Space Transformation
4. Ensemble Methods
5. Evaluation Results
6. Conclusion

Evaluation setup

- Test individual and ensemble methods on phase 1
- Select best strategies to apply on phase 2
- Divergence measure:
  - Deviation of prediction vector from equi-weighted class vector

$$div = \sum_{c \in Y} \left| \mathrm{Card}(\{i | g(f, \mathbf{x}_i^t) = c\}) - \frac{N}{N_c} \right|$$

  - N is number of test images, N_c is the number of classes
  - $\{i | g(f, \mathbf{x}_i^t) = c\}$ is target instances with predicted c



- Under equal class assumption, minimize the divergence.

xerox

Challenge Results

Place	Score	Acc	Run Name	Divgrc	Comment
1	228	**38.0**	combin6_Np20	108	UMV
2	228	**38.0**	combin3_Np18	108	UMV
3	226	37.67	combinAll6_Np19	164	UMV
4	217	36.17	combin6A_Np19	78	UMV + min div
5	214	35.67	MLNCM_MLDA_128	174	ML
6	212	35.33	combinAll7A_Np19	134	WMV
7	208	34.67	combin8A_Random_Np25	78	WMV + min div
8	185	30.83	MLNCMC_ML_128	168	ML
9	182	30.33	combin2_Np10	134	TrA+UMV
10	158	26.33	svmBoost_Mul_Power_f60	186	TrA

Table: Ten runs submitted by XRCE team.

Submission analysis

Individual DA methods

- Brute force performed poorly as expected, but but participated in various ensembles
- TrAdaboost and Metric Learning performed reasonably well

Ensembles of heterogeneous classifiers

- Is a right strategy
- Unweighted majority vote (UMV) on a small selection of classifiers performed the best
- Weighted majority vote (WMV) works well on large sets of classifiers but underperforms against the UMV
- Divergence minimization did not play any important role

Outline

1. Assembling Heterogeneous Methods
2. Domain Adaptation by Instance Transfer
3. Domain Adaptation by Space Transformation
4. Ensemble Methods
5. Evaluation Results
6. Conclusion

Conclusion

Using heterogeneous methods for domain adaptation is a right strategy

- Image classification in target domain can benefit a knowledge transfer from source domains
- Ensembles of heterogeneous classifiers with different majority votings yield the high accuracy
- Won the ImageCLEF Domain Adaptation competition
- New directions
  - Semi-supervised Learning in target domain
  - Both Instance Reuse and Metric Learning