Domain Adaptation for Visual Applications

Gabriela Csurka

Xerox Research Centre Europe

6 Rue de Maupertuis, 38240, Meylan, France

Gabriela.Csurka@xrce.xerox.com

Outline

1. Introduction
   Benchmark Datasets
2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods
3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Domain Adaptation (DA)

Leveraging labeled data in one or more related domains, referred to as source domains, to learn a classifier for unseen data in a target domain.

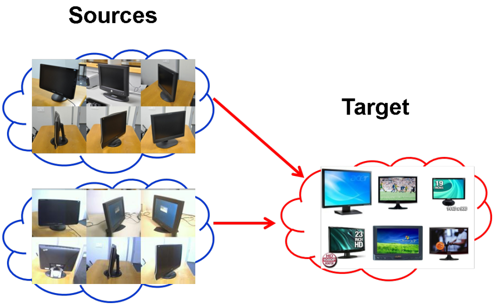


Image: Courtesy to S.J. Pan

- Unsupervised (US) DA when no label is available for the target
- Semi-supervised (SS) DA when we have a few labels in the target

Transfer Learning

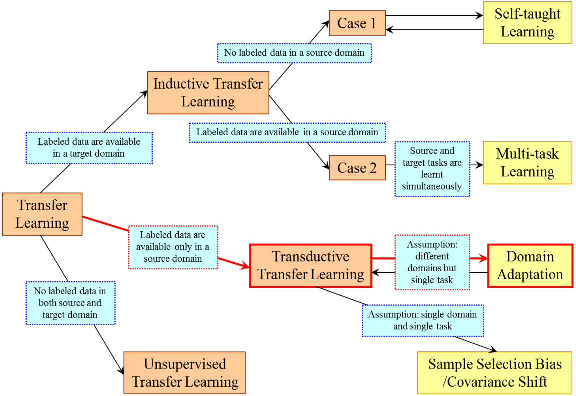


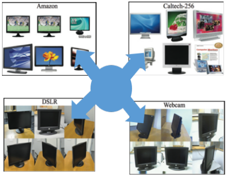
Image: Courtesy to S.J. Pan

Particular case of the transductive transfer learning where

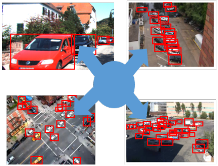
- domains are different but the task is the same (*e.g.* same classes)

Example scenarios

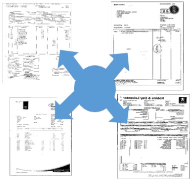
Object recognition



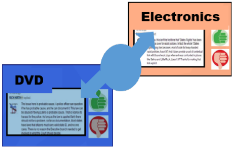
Object detection



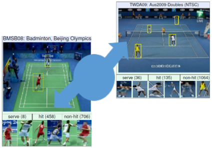
Document image categorization



Sentiment analyses



Action recognition



Speach recognition

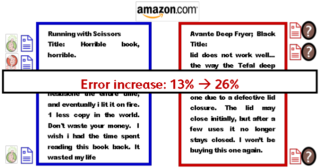
Why not just use source model?

Statistical Machine Translation

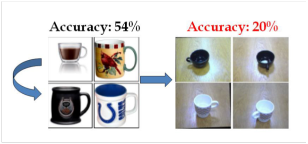
- Source Domain
  - European Parliament Text
- Target Domain:
  - Research Articles in Social Sciences.
- No Adaptation
  - 28% Mean Average Precision (MAP)
- With Adaptation 34%

- **German query:**
  Selbstmord von Jugendlichen
- **Baseline translation:**
  suicide of young (MAP = 0.3210)
- **Adaptation:**
  suicide of adolescents (MAP = 0.5277)

Sentiment analyses



Object recognition



Applying the models learned on the source directly often performs poorly!

Domain shift/distribution mismatch

Underlying causes:

- Image categorization
  - different point of views, acquisition time and conditions,
- Audio recognition
  - different persons, environment, recording quality
- Document image categorization
  - differences in appearance, layout
- Activity recognition
  - different persons, environment, context
- Semantic analyses
  - different topics, vocabularies, ...

Outline

1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Amazon review dataset[1] (AMT)

Products reviews in different domains

- kitchen (K), dvd (D), books (B) and electronics (E)
- 2 classes, about 5,000 document for each class
- TFIDF from processed text

[1] **Blitzer *et al.*, Domain adaptation with coupled subspaces, AIS11**

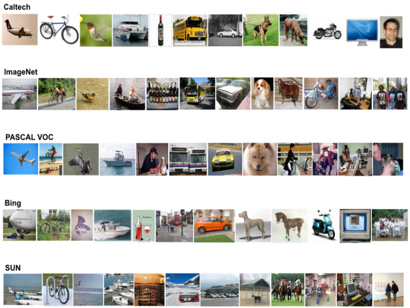
Office31[2] (OFF31) & Office+Caltech[3] (OC10)

Object recognition:

- Amazon (A), Caltech (C) , Dslr (D) , Webcam (W)
- 3 domains and 31 classes in OFF31 and 4 domains and 10 classes in OC10
- SURF BOV and Decaf6 (CNN activation) features
  - using all source examples (*ase*)
  - using several subset of the source data and average(*sse*)

[2] Saenko *et al.*, Adapting visual category models to new domains, ECCV10
[3] Gong *et al.*, Reshaping visual datasets for domain adaptation, NIPS13

The ImageCLEF’14 DA Challenge[4] (ICDA)



Object recognition:

- Caltech (C), ImageNet (I), Pascal (P), Bing (B), SUN (S)
- 12 classes, about 60 document for each class
- SIFT BOV features from images

---

[4] http://www.imageclef.org/2014/adaptation

Outline

1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Different solutions

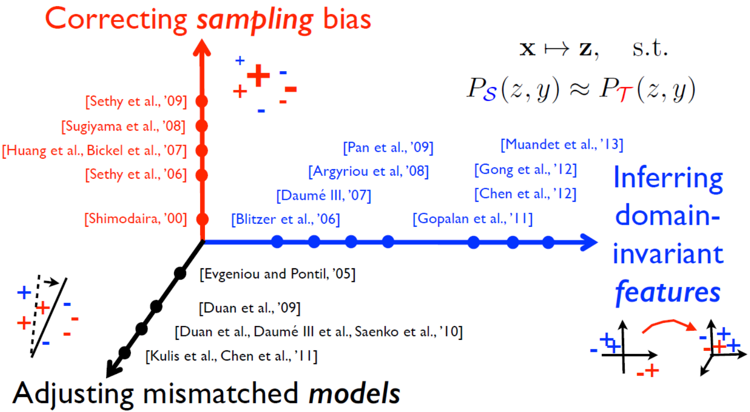


Image: Courtesy to Boqing Gong.

Outline

1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Correcting sampling bias

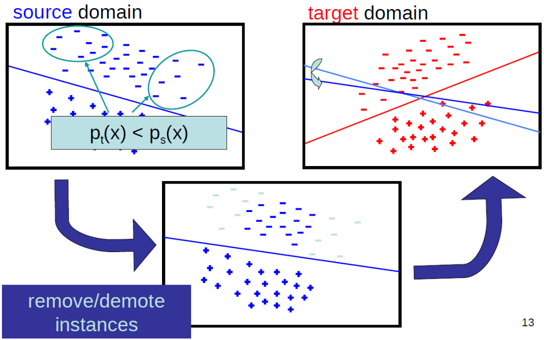


Image: Courtesy to Ming-Wei Chang.

Learn a classifier such that

- more weights are put on the examples that are similar to target instances
- less weights (or removing) on those that are less similar

How to estimate the weights

- Using the classifier that distinguishes between source and target examples (Bickel *et al.* ICML'07)

$$\alpha(\mathbf{x}_i^s) = \frac{1}{p(y_i^s = s|\mathbf{x}_i^s, \theta)}$$

- considering the ratio between the densities estimated for source and target domains (Sugiyama *et al.* NIPS'07, Kanamori *et al.* JMLR'09)

$$\alpha(\mathbf{x}) = \frac{P_T(\mathbf{x})}{P_S(\mathbf{x})} \approx \sum_l \alpha_l \phi_l(\mathbf{x})$$

Maximum Mean Discrepancy

- Minimizing the Maximum Mean Discrepancy (Huang *et al*. NIPS'06)

$$MMD(S,T) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \phi(\mathbf{x}_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} \phi(\mathbf{x}_j^t) \right\|_{\mathcal{H}}$$

where $\mathcal{H}$ is the RKHS (reproducing kernel Hilbert space) associated with the kernel k, and $\phi(\mathbf{x}) =< k(\mathbf{x}), . >$.

- Empirically:

$$MMD(S,T) = \left[\frac{1}{N_s^2} \sum_{i,j=1}^{N_s} k(\mathbf{x}_i^s, \mathbf{x}_j^s) - \frac{2}{N_s N_t} \sum_{i,j=1}^{N_s,N_t} k(\mathbf{x}_i^s, \mathbf{x}_j^t) + \frac{1}{N_t^2} \sum_{j,j=1}^{N_t} k(\mathbf{x}_i^t, \mathbf{x}_j^t) \right]$$

with k being *e.g*. the Gaussian Kernel.

Transfer Adaptive Boosting[5]

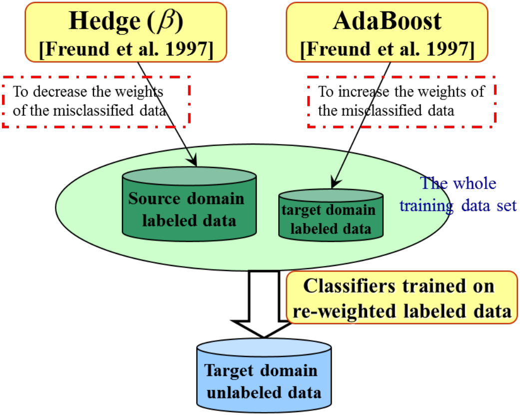


Image: Courtesy to S.J. Pan.

[5] **Dai *et al.*, Boosting for transfer learning, ICML'07.**

TrAdaboost result

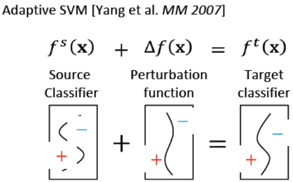
- 20 Newsgroups - text categorization across newsgroups.
- Abalone - seven physical measurements of abalone sea snails.
- Wine - red wine physical and chemical characteristics versus white wine.

Dataset	AdaBoost	TrAdaBoost	Fixed-Cost (1.1,1.2,1.3)	Dynamic
Sci vs Talk	0.552	0.577	0.581	**0.618**
Rec vs Sci	0.546	0.572	0.588	**0.631**
Rec vs Talk	0.585	0.660	0.670	**0.709**
Wine Quality	0.586	0.604	0.605	**0.638**
Abalone Age	0.649	0.689	0.682	**0.740**

- **TrAdaBoost** - Transfer Adaptive Boosting, Dai *et al*., ICML'07.
- **Dynamic** - Dynamic updates for TrAdaBoost, Al-Stouhi and Reddy, PKDD'11.

Outline

Adaptive SVM[6] (A-SVM)



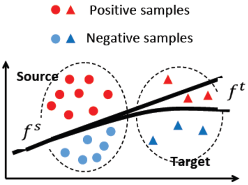


Image: Courtesy to Dong Xu.

The target classifier:

$$f^t(\mathbf{x}) = \sum_{k=1}^{M} \beta_k f_k^a(\mathbf{x}) + \sum_{i=1}^{N} \hat{\alpha}_i y_i K(\mathbf{x}, \mathbf{x}_i)$$

leverages multiple auxiliary classifiers f_k^a.

---

[6] **Yang *et al.*, Cross-domain video concept detection using adaptive SVMs, MM'07**

Domain Adaptation SVM[7] (DASVM)

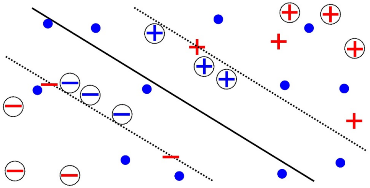


Image: Courtesy to Amaury Habrard.

Adapt iteratively (until stopping criteria reached) the classifier built with the source:

- add to negatives the first k predicted target $h(\mathbf{x}_t) > 0$ with highest margin
- add to positives the first k predicted target $h(\mathbf{x}_t) < 0$ with highest margin
- remove the first k positive and k negative source instances with highest margin

[7] **Bruzzone and Marconcini, Domain Adaptation Problems: A DASVM Classification Technique and a Circular Validation Strategy PAMI'10.**

Adaptive MKL[8]

- Proposed formulation of A-MKL

Structural risk functional

$$\min_{\mathbf{d}} \; G(\mathbf{d}) = \frac{1}{2} \cdot \mathrm{MMD}^2(\mathbf{d}) + \theta \cdot \underline{J(\mathbf{d})}$$

$$\text{where} \quad J(\mathbf{d}) = \min_{\mathbf{w}_m, \boldsymbol{\beta}, b, \xi_i} \; \frac{1}{2}\underline{\left(\sum_{m=1}^{M} d_m \|\mathbf{w}_m\|^2 + \lambda \|\boldsymbol{\beta}\|^2\right)} + C \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad \underline{y_i \cdot f^t(\mathbf{x}) \geq 1 - \xi_i, \xi_i \geq 0}$$

Regularization

Hinge loss

- Dual form of $J(\mathbf{d})$

$$\min_{\boldsymbol{\alpha}} \; -\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{1} + \frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})^{\mathsf{T}} \left(\sum_{m=1}^{M} d_m \tilde{\mathbf{K}}_m\right)(\boldsymbol{\alpha} \circ \mathbf{y}) \quad \text{s.t.} \quad \boldsymbol{\alpha}^{\mathsf{T}}\mathbf{y} = 0, \; \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}$$

where $\tilde{\mathbf{K}}_m(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{\lambda}\sum_{p=1}^{P} f_p(\mathbf{x}_i) f_p(\mathbf{x}_j)$

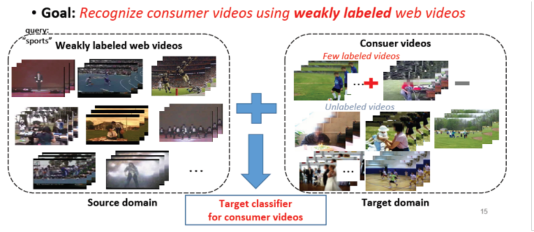
Solution:

Iteratively optimize $\mathbf{d}$ and solve the SVM problem.

Image: Courtesy to D. Xu.

[8] **Duan *et al.* Visual Event Recognition in Videos by Learning from Web Data, CVPR'10.**

Video event recognition



Three settings (a) **SIFT features**; (b) **ST features**; (c) **SIFT and ST features**

Means and standard deviations of mean average precision (%) over six classes

	SVM_T	SVM_AT	FR	A-SVM	MKL	DTSVM	A-MKL
MAP-(a)	42.32 ± 5.50	53.93 ± 5.58	49.98 ± 5.63	38.42 ± 7.93	47.19 ± 2.59	52.36 ± 1.88	57.14 ± 2.34
MAP-(b)	32.56 ± 2.08	24.73 ± 2.22	28.44 ± 2.61	24.95 ± 1.25	35.34 ± 1.55	31.07 ± 2.60	37.24 ± 1.58
MAP-(c)	42.00 ± 4.94	36.23 ± 3.37	44.11 ± 3.57	32.40 ± 4.99	46.92 ± 2.53	53.78 ± 2.99	**58.20 ± 1.87**

Image: Courtesy to D. Xu.

- **A-SVM** - Adaptive SVM, Yang *et al*. MM'07.
- **DTSVM** - Domain Transfer SVM, Duan, CVPR'09.
- **A-MKL** - Adaptive Multiple Kernel Learning, Duan *et al*. CVPR'10.

Outline

Feature space transformation

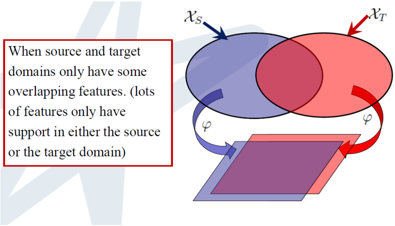


Image: Courtesy to Dong Xu.

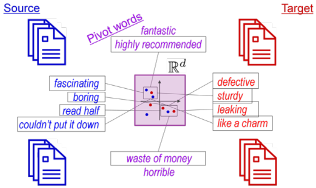
Unsupervised feature transform

- Align pivot features (Biltzer *et al*. EMNLP'06, Pan *et al*. WWW'10)
- Manifold based methods (Pan *et al*. IJCAI'09, Gong *et al*. CVPR'12)
- Unsupevised subspace alignement (Fernando *et al*. ICCV'12)
- Stacked Marginalized Denoising Autoencoders (Chen *et al*. ICML'12)

Semi-supervised feature transform

- Metric learning based approaches (Zha *et al*. IJCAI'09, Saenko *et al*. ECCV'10, Hoffman *et al*. ECCV'12, Csurka *et al*. Task-CV'14)
- Semi-supervised Transfer Component Analysis (Pan *et al*. TNN'11)

Structural Correspondence Learning[9]



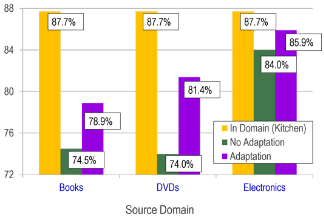


Image: Courtesy to Blitzer.

- Identify pivot features by mutual information between features and domains.
- Build P classifiers to predict the P pivot features from remaining features.
- Project to the shared subspace (using the top K eigenvectors).
- Concatenate with original features and train classifiers.

[9] **Blitzer *et al.*, Domain Adaptation with Structural Correspondence Learning, EMNLP'06**

Spectral Feature Alignment[10]

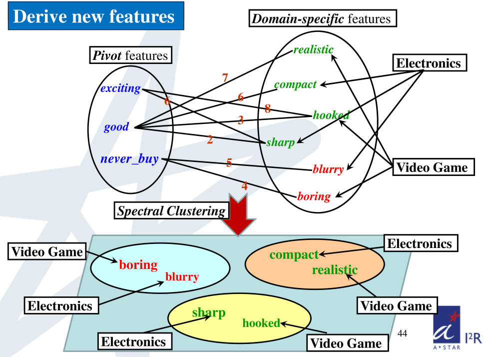


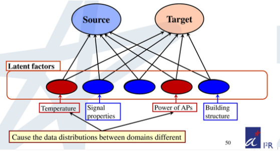
Image: Courtesy to Pan.

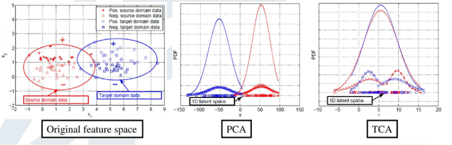
- Bipartite graph to model correlations between pivot features and the others.
- Discover new shared features by spectral clustering on the graph.

---

10 **Pan *et al.*, Cross-Domain Sentiment Classification via Spectral Feature Alignment, WWW'10**

Transfer Component Analysis[11] (TCA)





To minimize the distance between domains

Regularization on W

$$\min_{W} \ \mathrm{tr}(W^\top KLKW) + \lambda \mathrm{tr}(W^\top W)$$

$$\text{s.t.} \ W^\top KHKW = I.$$

To maximize the data variance

$W^* \Leftrightarrow$ m leading eigenvectors of $(KLK + \lambda I)^{-1} KHK$,

where $m \leq n_S + n_T - 1$.

Image: Courtesy to Pan.

[11] **Pan *et al.*, Domain Adaptation via Transfer Component Analysis, IJCAI'09**

Geodesic Flow Sampling[12] (GFS)

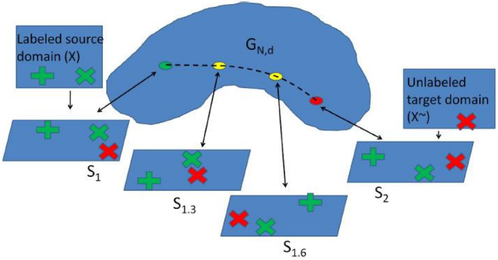


Image: Courtesy to Gopalan.

- Apply PCA on source data ($\mathbf{S}_1$ of rank d) and on the target ($\mathbf{S}_2$ of rank d).
- Geodesic path on the Grassman manifold $\mathbb{G}_{N,d}$ between $\mathbf{S}_1$ and $\mathbf{S}_2$.
- Exponential flow $\psi(t') = \mathbf{Q}\exp(t'\mathbf{B})\mathbf{J}$ such that $\mathbf{Q}^\top\mathbf{S}_1 = \mathbf{J}$ and $\mathbf{J}^\top = [\mathbf{I}_d \mathbf{0}_{N-d,d}]$.
- Compute $\mathbf{B}$ for intermediate subspaces varying $t \in [0, 1]$.

---

[12] **Gopalan *et al*. Domain adaptation for object recognition: An unsupervised approach, ICCV'11.**

Geodesic Flow Kernel[13] (GFK)

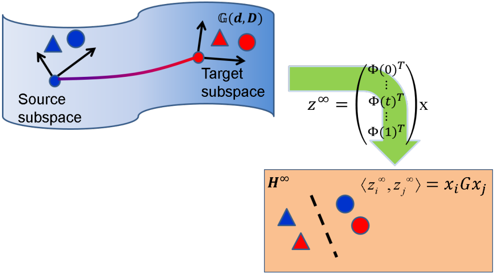


Image: Courtesy to Gong.

- Domain invariant features (infinite number of projections)

$$z^{\infty} = [\Phi(0)^{\top}\mathbf{x}, \ldots \Phi(t)^{\top}\mathbf{x} \ldots \Phi(1)^{\top}\mathbf{x}]$$

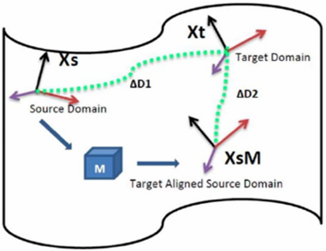
- Use the kernel trick

$$< z_i^{\infty}, z^{\infty} >= \int_0^1 (\Phi(t)^{\top}\mathbf{x})^{\top} (\Phi(t)^{\top}\mathbf{x}) \mathrm{d}t = \mathbf{x}_i^{\top} \mathbf{G} \mathbf{x}_j$$

---

[13] **Gong *et al.*, Geodesic flow kernel for unsupervised domain adaptation, CVPR'12.**

Subspace Alignment[14] (SA)



- $\mathbf{M}^* = \mathbf{S}_1'\mathbf{S}_2$ corresponds to the "subspace alignment matrix": $\mathbf{M}^* = \mathrm{argmin}_{\mathbf{M}} \|\mathbf{S}_1\mathbf{M} - \mathbf{S}_2\|$
- $X_a = \mathbf{S}_1\mathbf{S}_1'\mathbf{S}_2 = \mathbf{S}_1\mathbf{M}^*$ projects the source data to the target subspace
- A natural similarity: $\mathrm{Sim}(\mathbf{x}_s, \mathbf{x}_t) = \mathbf{x}_s\mathbf{S}_1\mathbf{M}^*\mathbf{S}_1'\mathbf{x}_t' = \mathbf{x}_s\mathbf{A}\mathbf{x}_t'$

Image: Courtesy to Fernando.

[14] **Fernando *et al*. Unsupervised Visual Domain Adaptation Using Subspace Alignment, ICCV'13.**

Marginalized Denoising Autoencoders[15] (MDA)

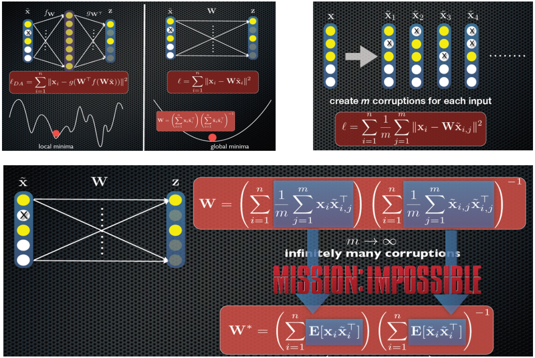


Image: Courtesy to Chen.

- Learns a direct mapping that allow closed form solution for a given corruption.
- Generates many (ideally infinity) corruptions for each input.

[15] **Chen *et al.*, Marginalized Stacked Denoising Autoencoders for Domain Adaptation, ICML'12**

MDA[16]

- Marginalizes out the corruption (convergence to expected values), hence **W** can be expressed in closed form as $W^* = \mathbb{E}[\mathbf{P}]\,\mathbb{E}[\mathbf{Q}]^{-1}$, where:

$$\mathbb{E}[\mathbf{P}]_{ij} = \mathbf{S}_{ij} q_j \;\text{ and }\; \mathbb{E}[\mathbf{Q}]_{ij} = \begin{cases} \mathbf{S}_{ij} q_i q_j, & \text{if } \; i \neq j \\ \mathbf{S}_{ij} q_i, & \text{if } \; i = j \end{cases}$$

with:

- $q = [1-p, \ldots, 1-p, 1] \in R^{n+1}$, p being the noise level and n the feature dimension
- $\mathbf{S} = \mathbf{X}\mathbf{X}^\top$ the covariance matrix of the uncorrupted data $\mathbf{X}$
- $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_m}]$ where $\mathbf{x}_i = [\boldsymbol{x}_i, 1]$ for the inputs $\boldsymbol{x}_i$, where the constant is never corrupted

[16] **Chen *et al.*, Marginalized Stacked Denoising Autoencoders for Domain Adaptation, ICML'12**

Stacked MDA[17] (SMDA)

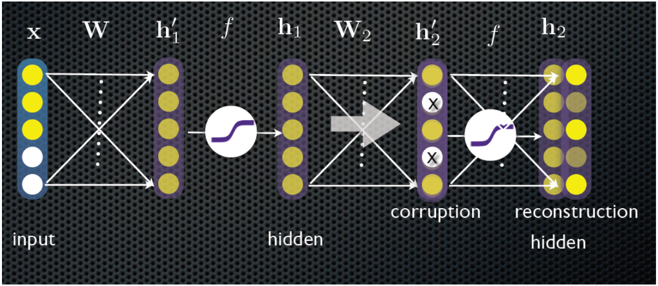


Image: Courtesy to M. Chen.

- Several MDA layers are stacked together
- Applying nonlinearities between layers helps
  - tangent-hyperbolic nonlinearities: $\mathbf{h}_t = \mathtt{tanh}(\mathbf{W}^t \mathbf{h}_{t-1})$

[17] **Chen *et al.*, Marginalized Stacked Denoising Autoencoders for Domain Adaptation, ICML'12**

Results on the OC10 (US *sse*) dataset

	C ->A	D ->A	W ->A	A ->C	D ->C	W ->C	A ->D	C ->D	W->D	A ->W	C ->W	D ->W	Average
TCA	46.7	39.6	**40.2**	40	34	33.7	**39.1**	41.4	77.5	**40.1**	36.2	80.4	45.74
GFS	36.8	32	27.5	35.3	29.4	21.7	30.7	32.6	54.3	31	30.6	66	35.66
GFK (PCA)	36.9	32.6	31.3	35.6	29.8	27.3	35.2	35.2	70.6	34.4	33.7	74.9	39.79
GFK (PLS)	40.4	36.1	35.5	37.9	32.7	29.3	35.1	41.1	71.2	35.7	35.8	79.1	42.49
SA (SVM)	46.1	**42**	39.3	39.9	35	31.8	38.8	39.4	**77.9**	39.6	38.9	**82.3**	**45.92**
SMDA	**49.85**	37.05	37.26	**41.99**	**36.65**	**33.93**	37.17	**45.59**	73.31	37.28	**43.7**	80.3	**46.17**

- **TCA** - Transfer Component Analysis, Pan *et al*. IJCAI'09.
- **GFS** - Geodesic Flow Sampling, Gopalan *et al*. ICCV'11.
- **GFK** - Geodesic Flow Kernel, B. Gong *et al*. CVPR'12.
- **SA** - Subspace Alignment, Fernando *et al*. ICCV'13.

- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen *et al*. ICML'12.

Results on AMT

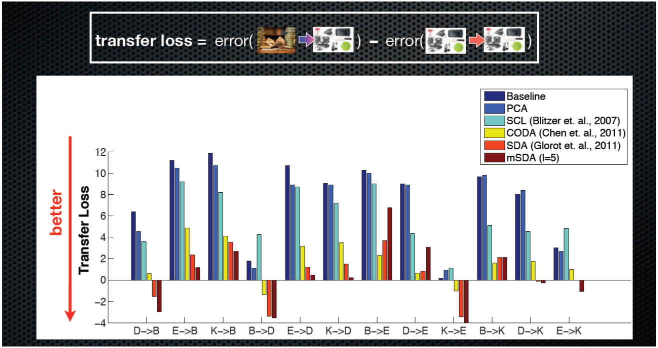
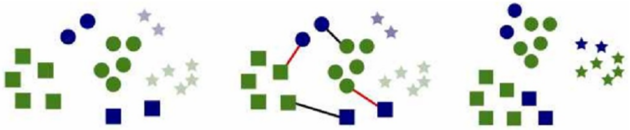


Image: Courtesy to M. Chen.

- **SCL** - Structural Correspondence Learning, Blitzer *et al.*, EMNLP'06.
- **SDA** - Deep learning approach - Glorot *et al*. ICML'11.
- **CODA** - Co-training for Domain Adaptation, Chen *et al*. NIPS'11.

- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen *et al*. ICML'12.

Information Theoretic Metric Learning[18] (ITML)



(a) Domain shift problem (b) Pairwise constraints (c) Invariant space

[Saenko et al.,ECCV'10]

Formulation (based on ITML [Davis et al.,ICML'07])

$$\min_{\mathbf{W}\succeq 0} \quad Tr(\mathbf{W}) - \log\det\mathbf{W}$$

$$\text{s.t.} \quad d^2_{\mathbf{W}}(\mathbf{x}^s_i, \mathbf{x}^t_j) \leq u, \forall(\mathbf{x}^s_i, \mathbf{x}^t_j) \in \mathcal{S}imilarSet$$

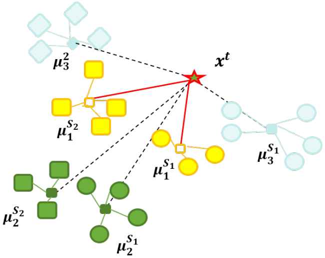
$$d^2_{\mathbf{W}}(\mathbf{x}^s_i, \mathbf{x}^t_j) \geq l, \forall(\mathbf{x}^s_i, \mathbf{x}^t_j) \in \mathcal{D}issimilarSet$$

$\Rightarrow$ Can be kernelized

Image: Courtesy to Habrard.

[18] **Saenko *et al.*, Adapting visual category models to new domains, ECCV 10**

Domain Specific Class Means[19] (DSCM)



The class label of a target is predicted based on

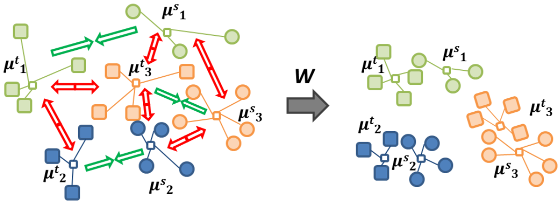
$$p(c|\boldsymbol{x}_i) = \frac{\sum_{d \in D} w_d e^{(-\frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{\mu}_d^c\|)}}{Z_i = \sum_{c'} \sum_d w_d e^{(-\frac{1}{2}\|\boldsymbol{x}_i - \boldsymbol{\mu}_d^{c'}\|}}$$

where

- μ_d^c is the mean of the class $c \in C$ in the domain $d \in D$
- w_d are domain specific weights (we used $w_{s_i} = 1$ and $w_t = 2$)

[19] **Csurka *et al.*, Domain adaptation with a domain specific class means classifier. TASK-CV14**

Metric Learning for DSCMs[20] (MLDSCM)



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,
- domain-specific weights w_d.

---

[20] **G. Csurka *et al*. Domain Adaptation with a Domain Specific Class Means Classifier, Task-CV'14.**

Semi-Supervised TCA[21]

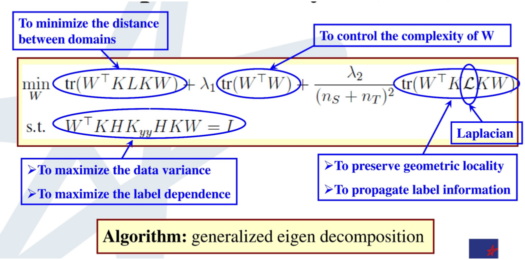


Image: Courtesy to Pan.

- **K** is the (RBF) kernel matrix between the data points (both source and target)
- $\mathcal{L}$ is the Laplacian of the affinity matrix $M_{i,j} = exp(-d^2_{i,j}/2\sigma^2)$.
- $\mathbf{K}_{yy}$ is a kernel matrix corresponding to source labels
- **L** integrates the normalizations $1/N_S^2$, $1/N_T^2$ and $1/N_S N_T$ and **H** is a centering matrix.

[21] **S.J. Pan *et al.*, Domain Adaptation via Transfer Component Analysis, TNN'11.**

Results on the OC10 (SS *sse*) dataset

	C ->A	D ->A	W ->A	A ->C	D ->C	W ->C	A ->D	C ->D	W->D	A ->W	C ->W	D ->W	Average
ITML	33.7	30.3	32.3	27.3	22.5	21.7	33.7	35	51.3	36	34.7	55.6	34.51
MLDSCM	**50.64**	**48.76**	**48.43**	**34.89**	**34.24**	33.42	**62.05**	**61.57**	64.65	**66.08**	**65.06**	71.47	**53.44**
SSTCA	47.1	40.1	41.5	40.4	34.2	**33.5**	39	41.7	**77.8**	41.1	36.2	**80.5**	46.09

- **ITML** - Information Theoretic Metric Learning, Saenko *et al.*, ECCV'10
- **MLDSCM** - ML for Domain Specific Class Means, Csurka *et al.* Task-CV WS'14.
- **SSTCA** - Semi-Supervised TCA, Pan *et al.*, TNN'11.

Outline

1. Introduction
    - Benchmark Datasets
2. Main domain adaptation methods
    - Instance reweighing methods
    - Parameter based methods
    - Feature transformation-based methods
3. Combined methods
    - Joint feature transform and parameter adaptation
    - Joint feature/instance selection and feature transform
    - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Methods combining several of the above ideas

- Joint feature and parameter adaptation
  - Max-Margin Domain Transforms (MMDT) of Hoffman *et al*. ICLR13
  - Naive Bayes NN based DA (NBNN-DA) of Tommasi and Caputo, ICCV13
  - Domain Invariant Projection (DIP-CC) of M. Baktashmotlagh *et al*. ICCV13
  - Joint Distribution Adaptation (JDA) of Long *et al*. ICCV 14
  - Transfer Joint Matching (TJM) of Long *et al*. ICCV 14
  - Optimal Transport for Domain Adaptation (OTDA) of Courty *et al*. PAMI 15
- Joint feature or instance selection and parameter adaptation
  - Feature selection and subspace learning (FSSL) of Gu *et al*. IJCAI 11
  - Landmark Selection (LM) of Gong *et al*. ICML13
  - Landmarks Selection-based SA (LSSA) of Aljundi CVPR15
- Joint instance selection, feature transform and parameter adaptation
  - Adaptive Transductive Transfer Machines (ATTM) of Farajidavar *et al*. BMVC 14
  - Statistically Invariant Embedding (SIE-CC) of M. Baktashmotlagh *et al*. CVPR 14

Outline

1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Max-Margin Domain Transforms[22] (MMDT)

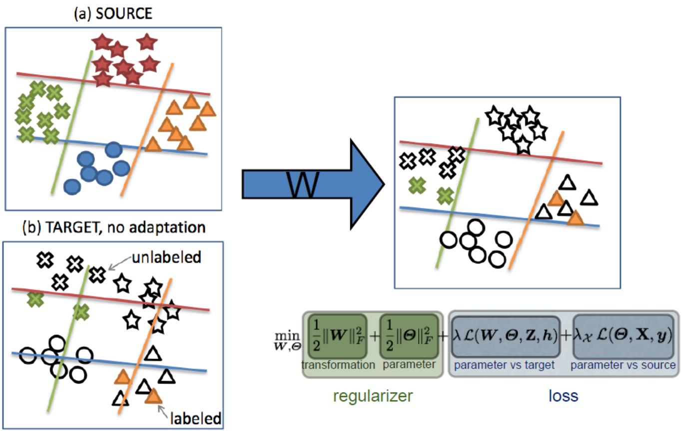


Image: Courtesy to Hoffman.

[22] **Hoffman *et al.*, Max-margin transforms for visual domain adaptation, ICLR'13.**

Naive Bayes NN based DA [23] (NBNN-DA)

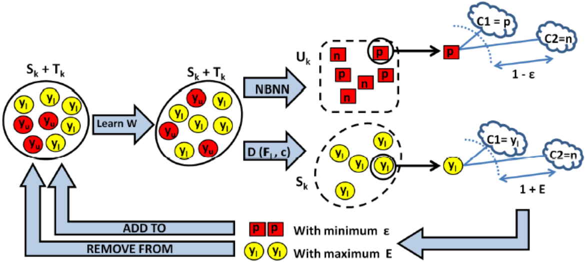


Image: Courtesy to Tommasi.

Iteratively combine metric learning and NBNN-based sample selection to:

- adjust the image-to-class distances by tuning the per class metrics
- iteratively making the metric progressively more suitable for the target

[23] **Tommasi and Caputo, Frustratingly Easy NBNN Domain Adaptation, ICCV'13.**

xerox

Transfer Joint Matching[24] (TJM)

```
Algorithm 1: TJM: Transfer Joint Matching
Input: Data X; #subspace bases k, regularization parameter λ.
Output: Adaptation matrix A, embedding Z, adaptive classifier f.
begin
    Compute MMD matrix M by Equation (5), and kernel matrix
    K by K_ij ← K(x_i, x_j) where K(·,·) is a predefined kernel.
    Set M ← M/||M||_F, G ← I.
    repeat
        Solve the generalized eigendecomposition problem in
        Equation (9) and select the k smallest eigenvectors to
        construct the adaptation matrix A, and Z ← A^T K.
        Update the sub-gradient matrix G by Equation (10).
    until Convergence
    Return an adaptive classifier f trained on {A^T k_i, y_i}_{i=1}^{n_s}.
```

[24] **Long *et al*. Transfer Joint Matching for Unsupervised Domain Adaptation, CVPR'14.**

Joint Distribution Adaptation[25] (JDA)

Algorithm 1: JDA: Joint Distribution Adaptation

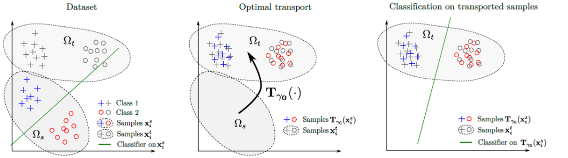
Input: Data $\mathbf{X}$, $\mathbf{y}_s$; #subspace bases k, regularization parameter λ.
Output: Adaptation matrix $\mathbf{A}$, embedding $\mathbf{Z}$, adaptive classifier f.

```
begin
    Construct MMD matrix M_0 by Eq. (4), set {M_c := 0}_{c=1}^{C}.
    repeat
        Solve the generalized eigendecomposition problem in
        Equation (10) and select the k smallest eigenvectors to
        construct the adaptation matrix A, and Z := A^T X.
        Train a standard classifier f on {(A^T x_i, y_i)}_{i=1}^{n_s} to
        update pseudo target labels {ŷ_j := f(A^T x_j)}_{j=n_s+1}^{n_s+n_t}.
        Construct MMD matrices {M_c}_{c=1}^{C} by Equation (6).
    until Convergence
    Return an adaptive classifier f trained on {Ax_i, y_i}_{i=1}^{n_s}.
```

---

25 **M. Long *et al.* Transfer Feature Learning with Joint Distribution Adaptation, ICCV'14**

Optimal Transport for Domain Adaptation[26] (OTDA)



Algorithm 1 Conditional gradient splitting (CGS)

1: Initialize $k = 0$ and $\gamma^0 \in \mathcal{P}$
2: **repeat**
3: With $\mathbf{G} \in \nabla f(\gamma^k)$, solve

$$\gamma^\star = \underset{\gamma \in \mathcal{B}}{\operatorname{argmin}} \quad \langle \gamma, \mathbf{G} \rangle_F + g(\gamma)$$

4: Find the optimal step with $\Delta\gamma = \gamma^\star - \gamma^k$

$$\alpha^k = \underset{0 \le \alpha \le 1}{\operatorname{argmin}} f(\gamma^k + \alpha\Delta\gamma) + g(\gamma^k + \alpha\Delta\gamma)$$

5: $\gamma^{k+1} \leftarrow \gamma^k + \alpha^k \Delta\gamma$, set $k \leftarrow k + 1$
6: **until** Convergence

Image: Courtesy to Courty.

[26] **N. Courty *et al.*, Optimal Transport for Domain Adaptation, CoRR'15.**

Domain Invariant Projection[27] (DIP-CC)

- Optimizing the MMD on the Grassman manifold $\mathcal{G}(d, D)$

$$D_{\mathcal{H}}(\mathbf{W}^\top \mathbf{X}_S, \mathbf{W}^\top \mathbf{X}_T) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \phi(\mathbf{W}^\top \mathbf{x}_i^s) - \frac{1}{N_t} \sum_{j=1}^{N_t} \phi(\mathbf{W}^\top \mathbf{x}_j^t) \right\|_{\mathcal{H}}$$

where $\mathbf{W}$ is a point on $\mathcal{G}$ with the constraint $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$.

- Adding term to encourage Class Clustering (CC) :

$$\sum_{c=1}^{C} \sum_{i=1}^{n_c} \|\mathbf{W}^\top (\mathbf{x}_{i,c}^s - \mu_\mathbf{c})\|^2$$

- The whole yielding to the optimization problem:

$$W^* = \underset{W}{\operatorname{argmin}} \; Tr(K_W L) + \lambda \sum_{c=1}^{C} \sum_{i=1}^{n_c} \left\| W^T (x_s^{i,c} - \mu_c) \right\|^2$$
$$\text{s.t.} \quad W^T W = I \,,$$

$$K_W = \begin{bmatrix} K_{s,s} & K_{s,t} \\ K_{t,s} & K_{t,t} \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$$
$$L_{ij} = \begin{cases} 1/n^2 & i,j \in \mathcal{S} \\ 1/m^2 & i,j \in \mathcal{T} \\ -1/(nm) & \text{otherwise} \end{cases},$$

[27] M. Baktashmotlagh *et al.*, Unsupervised Domain Adaptation by Domain Invariant Projection, ICCV'13.

Outline

1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Feature selection and subspace learning[28] (FSSL)

(d) Selected features

file attachments

Algorithm 2 Joint Feature Selection and Subspace Learning (Situation 2)

Initialize: $\mathbf{G}_0 = \mathbf{I}, t = 0$ and μ;
Compute $\mathbf{Y}$ based on $\mathbf{WY} = \mathbf{\Lambda DY}$;
repeat
  Compute $\mathbf{A}_{t+1} = \mathbf{G}_t^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{G}_t^{-1}\mathbf{X} + \frac{1}{2\mu}\mathbf{I})^{-1}\mathbf{Y}$;
  Compute $\mathbf{G}_{t+1}$ based on $\mathbf{A}_{t+1}$;
  $t = t + 1$;
until convergence

Image: Courtesy to Gu.

[28] **Gu, *et al.* Joint feature selection and subspace learning, IJCAI'11.**

Landmark Selection[29]

Landmarks are labeled source instances distributed similarly to the target domain.

Identifying landmarks:

$$P_{\mathcal{L}}(\text{landmarks}) \approx P_{\mathcal{T}}(\text{target})$$

$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

[Gong *et al.*, ICML'13]

Convex relaxation

$$\min_{\{\alpha_m\}} \quad \left\| \frac{1}{\sum_i \alpha_i} \sum_{m=1}^{M} \alpha_m \phi(x_m) - \frac{1}{N} \sum_{n=1}^{N} \phi(x_n) \right\|_{\mathcal{H}}^2$$

$$\beta_m = \frac{\alpha_m}{\sum_i \alpha_i}$$

$$\min_{\beta} \quad \beta^T K^s \beta - \frac{2}{N} \beta^T K^{st} \mathbf{1}$$

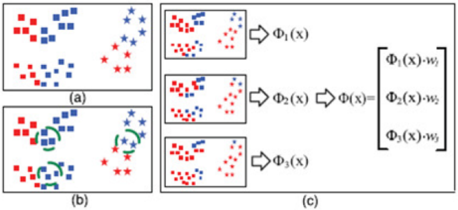


Image: Courtesy to Gong.

[29] **Gong *et al.*, Connecting the dots with landmarks: Discriminatively learning domain invariant features for unsupervised domain adaptation, ICML'13.**

Landmarks Selection-based SA[30] (LSSA)

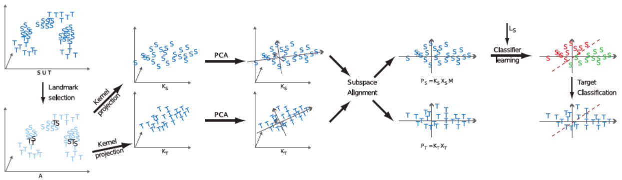


Image: Courtesy to Aljundi.

- Landmark selection using a Gaussian Kernels and overlap between the probability densities:

$$overlap(\mu_S, \sigma_S; \mu_T, \sigma_T) = \frac{\mathcal{N}(mu_S, -\mu_T | 0, (\sigma_S + \sigma_T)^2)}{\mathcal{N}(0|0, (\sigma_S + \sigma_T))}$$

- Subspace Alignment using the selected landmarks and linear SVM classifiers.

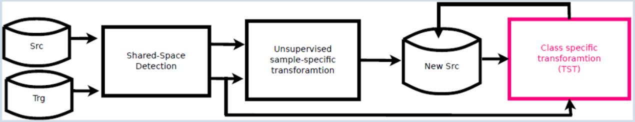
30 **Aljundi *et al.*, Landmarks-based Kernelized Subspace Alignment for Unsupervised Domain Adaptation, CVPR'15.**

Outline

1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Transductive Transfer Machines[31] (ATTM)

TTM Diagram



ATTM Algorithm

Input: $\mathbf{X}^{src}, \mathbf{Y}^{src}, \mathbf{X}^{trg}$
Output: $\mathbf{Y}^{trg}$
1. Search for the shared subspace between the two domains
2. Adjust the marginal distribution mismatch between the two domains
3. Select the appropriate classifier, if it is kernel-based, tune σ using

while $T < 10$ and $|D^{\text{global}}(G^t(\mathbf{X}^{src}), \mathbf{X}^{trg})| > \mathit{threshold}$ **do**
  4. Find the feature-wise TST transformation
  5. Transform the source domain clusters
  6. Retrain the classifier using the transformed source

end while

Image: Courtesy to deCampos.

[31] **Farajidavar *et al.*, Adaptive Transductive Transfer Machines, BMVC'14.**

Statistically Invariant Embedding [32] (SIE-CC)

Image: Courtesy to Baktashmotlagh.

- Statistically Invariant Sample Selection

$$\min_{\beta} \sum_{i=1}^{n_s} \beta_i \left(\sqrt{T(x_i^s)} - \sqrt{1 - T(x_i^s)} \right)^2 + \frac{1}{n_t} \sum_{i=1}^{n_t} \left(\sqrt{T(x_i^t)} - \sqrt{1 - T(x_i^t)} \right)^2$$

$$\text{s.t.} \quad \beta_i \in [0,1]; \quad \sum_{i=1}^{n_s} \beta_i = 1; \quad \sum_{i=1}^{n_s} \beta_i y_{i,c} = \frac{1}{n_s} \sum_{i=1}^{n_s} y_{i,c} \quad \forall c$$

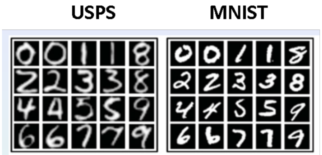
- Statistically Invariant Embedding

$$\min_{\mathbf{W}} D_{\mathcal{H}}(\mathbf{W}^\top \mathbf{X}_S, \mathbf{W}^\top \mathbf{X}_T) + \lambda \sum_{c=1}^{C} \sum_{i=1}^{n_c} \| \mathbf{W}^\top (\mathbf{x}_{i,c}^s - \mu_c) \|^2$$

$$\text{s.t.} \quad \mathbf{W}^\top \mathbf{W} = \mathbf{I}$$

[32] **Baktashmotlagh *et al.*, Domain Adaptation on the Statistical Manifold, CVPR'14.**

Digit recognition results



	GFK	TCA	FSSL	TJM	JDA	OTDA	ATTM
U ->M	46.45	51.05	51.45	52.25	59.65	58.3	**61.15**
M ->U	67.22	56.28	57.44	63.28	67.28	69.39	**77.9**

- **GFK** - Geodesic Flow Kernel, B. Gong *et al*. CVPR'12.
- **TCA** - Transfer Component Analysis, Pan *et al*. IJCAI'09.
- **FFSL** - Feature selection and subspace learning, Gu *et al*. IJCAI'11.
- **TJM** - Transfer Joint Matching, Long *et al*. CVPR'14.
- **JDA** - Joint Distribution Adaptation, Long *et al*. ICCV'14.
- **OTDA** - Optimal Transport for Domain Adaptation, Courty *et al*. CoRR'15.
- **ATTM** - Adaptive Transductive Transfer Machines, Farajidavar *et al*., BMVC'14.

Object recognition (OC10 US *ase*)

	C ->A	W ->A	A ->C	W ->C	A ->D	C ->D	W->D	A ->W	C ->W	avg
SMDA	52.8	35.3	41.9	32.3	37.8	47.2	77.9	37	49.8	45.79
SA	52.7	39.4	41.6	34.7	46.4	49	78.9	40.7	42.7	47.34
LSSA	58.4	39.4	44.8	34.7	42.4	54.1	87.2	42.4	48.1	50.17
LS	56.7	40.2	45.5	35.4	47.1	57.3	75.2	46.1	49.5	50.3
TJM	58.6	40.8	45.7	34.8	42	49	83.4	42	48.8	49.45
JDA	44.8	32.8	39.4	31.2	39.5	45.2	89.2	38	41.7	44.62
OTDA	48	40	38.6	**37.2**	45	45.2	93	41.4	44.7	48.08
ATTM	**60.8**	39.7	42.9	34	31.8	50.3	89.2	**50.5**	38	48.59
DIP-CC	58.7	40.9	47.2	**37.2**	**49**	**61.2**	91.7	47.8	**58**	**54.63**
SIE-CC	57.6	**42.4**	**47.6**	36.2	**49**	**61.2**	**93**	47.8	57.3	**54.68**

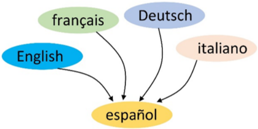
- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen *et al*. ICML'12.
- **SA** - Subspace Alignment, Fernando *et al*. ICCV'13.
- **LSSA** - Landmarks Selection-based SA, Aljundi *et al*. CVPR'15.
- **LS** - Landmark Selection, Gong *et al*. ICML'13.
- **TJM** - Transfer Joint Matching, Long *et al*. CVPR'14.
- **JDA** - Joint Distribution Adaptation, long *et al*. ICCV'14.
- **OTDA** - Optimal Transport for Domain Adaptation, Courty *et al*. CoRR'15.
- **ATTM** - Adaptive Transductive Transfer Machines, Farajidavar *et al*., BMVC'14.
- **DIP-CC** Domain Invariant Projection, Baktashmotlagh *et al*. ICCV'13.

- **SIE-CC** - Statistically Invariant Embedding, Baktashmotlagh *et al*. CVPR'14.

Outline

Example: cross-lingual text categorization

- Experiments on the Reuters multilingual dataset



- Dataset information.

	Domain	# dim after PCA	# total docs	# training docs per class
Source	English	1,131	18,758	100
	French	1,230	26,648	100
	German	1,417	29,953	100
	Italian	1,041	24,039	100
Target	Spanish	807	11,547	5/7/10/15/20

- Means and standard deviations of classification accuracies (%) of all methods on the Reuters multilingual dataset by using 10 labeled training samples per class from the target domain Spanish. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

Source Domain	SVM_T	KCCA	HeMap	DAMA	ARC-t	HFA
English	72.6 ± 2.3	71.4 ± 3.2	65.7 ± 3.1	72.4 ± 2.4	72.9 ± 2.0	$\mathbf{75.3 \pm 1.7}$
French		72.8 ± 2.8	64.2 ± 4.2	72.8 ± 2.0	73.5 ± 1.8	$\mathbf{75.7 \pm 1.6}$
German		73.8 ± 2.2	64.6 ± 3.6	72.9 ± 2.3	74.7 ± 1.6	$\mathbf{76.1 \pm 1.5}$
Italian		73.8 ± 2.1	65.8 ± 2.3	73.3 ± 2.1	74.0 ± 2.0	$\mathbf{75.8 \pm 1.8}$

Image: Courtesy to Dong Xu.

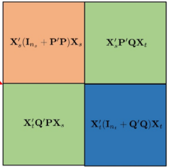
Heterogeneous Feature Augmentation[33] (HFA)

- Objective
  - The dual form is similar to SVM with a different kernel.

$$\min_{\mathbf{P},\mathbf{Q}} \max_{\boldsymbol{\alpha}} \ \mathbf{1}'_{n_s+n_t}\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\boxed{\mathbf{K}_{\mathbf{P},\mathbf{Q}}}(\boldsymbol{\alpha}\circ\mathbf{y})$$

$$s.t.\ \mathbf{y}'\boldsymbol{\alpha} = 0,\ \mathbf{0}_{n_s+n_t} \leq \boldsymbol{\alpha} \leq C\mathbf{1}_{n_s+n_t},$$

$$\|\mathbf{P}\|_F^2 \leq \lambda_p, \|\mathbf{Q}\|_F^2 \leq \lambda_q.$$



  - Global optimum can be solved using the method similarly as in MKL (see our T-PAMI 2014 work for more details)

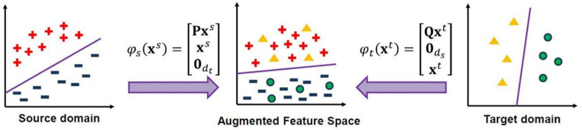


Image: Courtesy to Dong Xu.

- Generalizes the feature replication method of Daume III, ACL 07.

[33] **Duan *et al.* Learning with Augmented Features for Heterogeneous Domain Adaptation, CVPR12.**

Dictionary-based Approaches

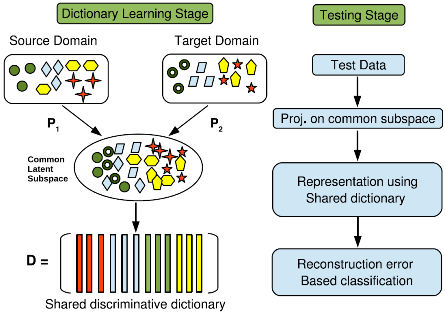


Image: Courtesy to S. Shekhar.

- Each domain has its own projection matrix.
- Can be used when features in source and target are different.
- Generalizes well to multiple sources.

Shared Domain-adapted Dictionary Learning[34] (SDDL)

- Main idea:

$$\{\mathbf{D}^*, \tilde{\mathbf{W}}^*, \tilde{\mathbf{X}}^*\} = \underset{\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}}{\operatorname{argmin}} \, \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{W}})$$

$$\text{s.t. } \mathbf{W_i W_i^T} = \mathbf{I}, \; i = 1, 2 \text{ and } \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j$$

$$\tilde{\mathbf{W}} = [\mathbf{W_1} \;\; \mathbf{W_2}], \;\; \tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_l \end{pmatrix}, \text{ and } \tilde{\mathbf{X}} = [\mathbf{X_1} \;\; \mathbf{X_2}].$$

$$\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{W}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2,$$

$$\mathcal{C}_2(\tilde{\mathbf{W}}) = -\text{trace}((\tilde{\mathbf{W}}\tilde{\mathbf{Y}})(\tilde{\mathbf{W}}\tilde{\mathbf{Y}})^T)$$

- Its kernelized extension

$$\mathcal{C}_1(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}\|_F^2 +$$
$$\mu\|\tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K}(\mathbf{I} - \tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{in}}\|_F^2 + \nu\|\tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K}\tilde{\mathbf{B}}\tilde{\mathbf{X}}_{\mathbf{out}}\|_F^2,$$
$$\mathcal{C}_2(\tilde{\mathbf{A}}) = -\text{trace}((\tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K})(\tilde{\mathbf{A}}^{\mathbf{T}}\mathcal{K})^T)$$

[34] **Shekhar *et al.*, Generalized Domain-Adaptive Dictionaries, CVPR 13.**

Results on the OC10 (SS *sse*)

	C ->A	D ->A	W ->A	A ->C	W ->C	C ->D	A ->W	D ->W	Average
FDDL	39.3	36.7	41.1	24.3	22.9	55	50.4	65.9	38.5
SDDL	49.5	48.9	49.4	27.4	29.7	**76.7**	72	72.6	53.3
MLDSCM	50.64	48.76	48.43	34.89	33.42	61.57	66.08	71.47	51.9
DIP-CC	**61.8**	**56.9**	**53.4**	**47.8**	**43.6**	65.8	**72.5**	**89.1**	**61.36**

- **FDDL** Fisher Discrimination Dictionary Learning, Yang *et al*. ICCV'11.
- **SDDL** Shared Domain-adapted Dictionary Learning, Shekhar *et al*. CVPR 13.
- **MLDSCM** - ML for Domain Specific Class Means, Csurka *et al*. Task-CV WS'14.
- **DIP-CC** Domain Invariant Projection, Baktashmotlagh *et al*. ICCV'13.

Outline

1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. **Multiple sources**
6. Deep Learning
7. Conclusion and Perspectives

Multi-sources results on the OFF31 (SS *sse*)

(a) *amazon* (b) *dslr* (c)*webcam*

(SS)

	A ,D->W	A,W ->D	D,W->A	Avg
DSCM	58.21	57.65	21.67	45.84
SDDL	57.8	56.7	24.1	46.2
BGFS	64.5	51.3	**38.4**	51.4
HL2L	**66.1**	**67.9**	25.8	**53.27**

(US)

	A ,D->W	A,W ->D	D,W->A	Avg
DSCM	45.04	35.51	12.12	30.89
GFS	37.5	33.9	31.5	34.3
SSF	**47.8**	**49.6**	**40.2**	**45.87**

- **DSCM** - Domain Specific Class Means, Csurka *et al*. Task-CV'14.
- **SDDL** Shared Domain-adapted Dictionary Learning, Shekhar *et al*. CVPR 13.
- **GFS** - Geodesic Flow Sampling, Gopalan *et al*. ICCV'11.
- **BGFS** - Boosted Geodesic Flow Sampling, Gopalan *et al*. CoRR'15.
- **HL2L** - High-level Learning to Learn, Patricia and Caputo, CVPR'14.
- **SSF** - Spline Flow Sampling, Caseiro *et al*. CVPR'15.

Boosted Geodesic Flow Sampling[35] (BGFS)

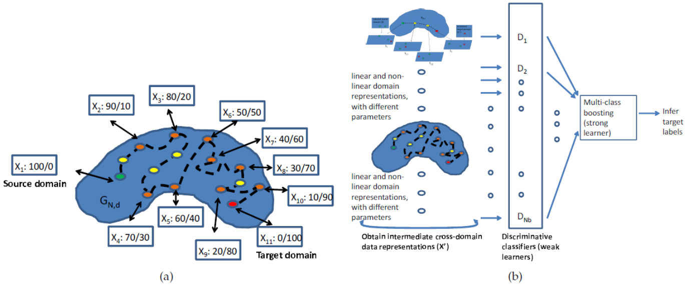


Image: Courtesy to Gopalan.

- Intermediate ”domains” between source and target on the Grassman Manifold
- Multi-domain adaptation using Karcher mean of domains
- Jointly learn domain shift features and classifiers with AdaBoost

[35] **Gopalan *et al.* Unsupervised Adaptation Across Domain Shifts By Generating Intermediate Data Representations, CoRR'15.**

Spline Flow Sampling [36] (SSF)

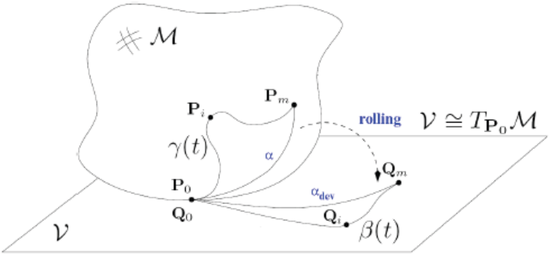


Image: Courtesy to Caseiro.

- $\alpha(t) : [0, T] \mapsto \mathcal{M}$ the rolling geodesic curve on the manifold $\mathcal{M}$
- $\beta(t) : [0, T] \mapsto \mathcal{V}$ the spline curve on the tangent space $\mathcal{V}$
- $\gamma(t) : [0, T] \mapsto \mathcal{M}$ the spline curve on the manifold $\mathcal{M}$

---

[36] **Caseiro *et al.* Beyond the shortest path : Unsupervised Domain Adaptation by Sampling Subspaces along the Spline Flow, CVPR 15.**

High-level Learning to Learn [37] (H-L2L)

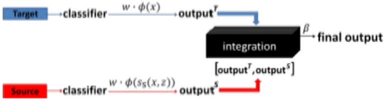


Image: Courtesy to Patricia.

Boosting approach to learn β_y^c where the weak learners are:

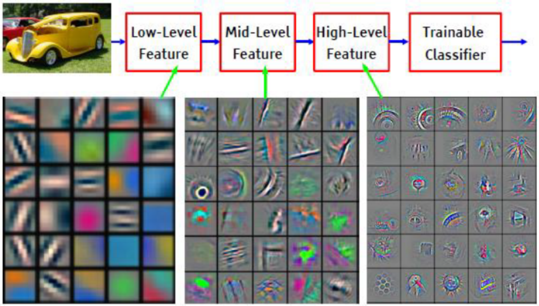
$$s(\mathbf{x}, y) = \beta^0 \mathbf{w}_0 \phi^0(\mathbf{x}, y) + \sum_{c=1}^{N_C} \beta_y^c \mathbf{w}_y^c \phi_y^c(s_S(\mathbf{x}, c), y)$$

- $s_S(\mathbf{x}, c)$ the score of $\mathbf{x}$ with the source classifier c
- $\phi_y^c \mapsto Y \times Y$ is the y^{th} score mapping corresponding to the c^{th} source model
- $\mathbf{w}_y^c$ the $y - th$ source model in predicting that $\mathbf{x}$ belongs to class c
- $\phi^0(\mathbf{x}, y)$ is the feature mapping for the original input features

[37] **Patricia and Caputo, Learning to Learn, from Transfer Learning to Domain Adaptation: A Unifying Perspective, CVPR 14.**

Outline

What about deep features?



Deep Learning: More than one stage of non-linear feature transformation

Image: Courtesy to Y LeCun H.

- High-non linearity makes these features more invariant across domains!

Deep convolutional activation features[38] (DeCAF)

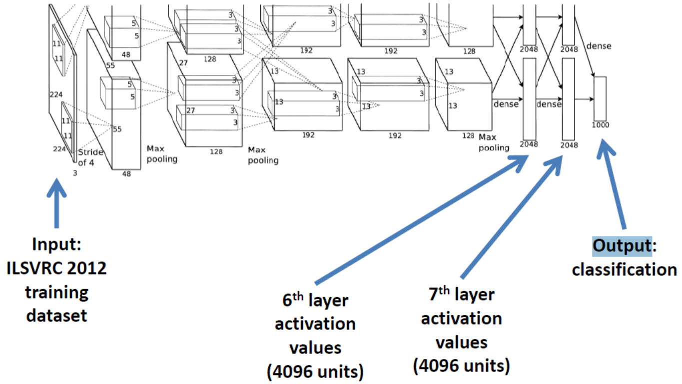
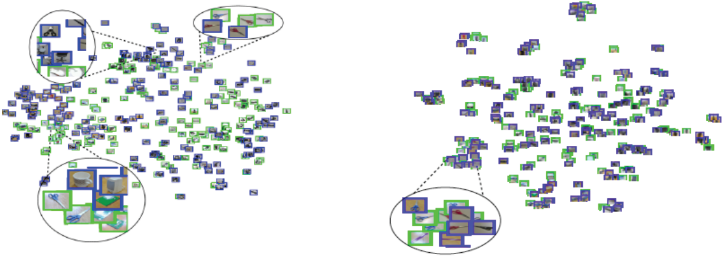


Image: Courtesy to A. Krizhevsky.

- CNN model pre-trained on big data (ImageNet)
- use convolutional activation layers as features.

[38] **Donuahe *et al.* DeCAF: A deep convolutional activation feature for generic visual recognition, ICML 14**

Compared to bag of visual-words[39] (BOV)



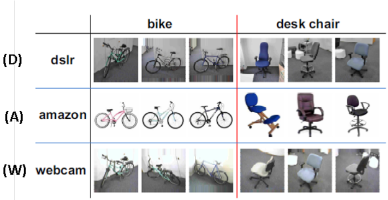
(a) SURF features (b) $DeCAF_6$

Image: Courtesy to Donuahe.

- DeCAF provides better category level clustering than SURF BOV

[39] **G. Csurka *et al*. Visual categorization with bags of keypoints, SLCV 2004.**

Results on OFF31 (US *sse*)



	A ->W	D ->W	W->D	Avg
NN	42.8	86.4	88.6	72.6
SVM	63.05	95.04	96.74	84.94
NCM	64.76	92.17	94.37	83.77
SMDA	68.56	92.92	95.01	85.5
GFK	46.8	87.2	88.1	74.03
TCA	44.6	89	87.9	73.83
SA	47.2	91.8	92.4	77.13

- **NN** - Nearest Neighbor classifier (no adaptation).
- **SVM** - Linear SVM classifier (no adaptation).
- **NCM** - Nearest Class Means classifier (DSCM with single domain, no adaptation).
- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen *et al*. ICML'12.
- **GFK** - Geodesic Flow Kernel, B. Gong *et al*. CVPR'12.
- **TCA** - Transfer Component Analysis, Pan *et al*. IJCAI'09.
- **SA** - Subspace Alignment, Fernando *et al*. ICCV'13.

Results on OC10 (US *sse*)

	C ->A	C ->W	C ->D	A ->C	A ->W	A ->D	W->C	W->A	W->D	D ->C	D ->A	D ->W	Avg
NN	85.7	66.1	74.52	70.35	64.97	57.29	60.37	62.53	98.73	52.09	62.73	89.15	70.33
SVM	**94.61**	85.28	87.4	88.47	86.79	88.98	86	87.28	100	84.81	87.28	98.4	89.61
NCM	94.07	87.55	87.4	88.2	87.17	85.04	87.01	90.3	99.21	85.82	90.52	98.49	90.6
SMDA	93.86	**90.19**	88.98	**89.48**	**87.92**	88.98	**89.48**	**93.64**	99.21	**89.02**	**93.32**	**99.25**	**91.94**
JDA	89.77	83.73	86.62	82.28	78.64	80.25	83.53	90.19	**100**	85.13	91.44	98.98	87.55
ATTM	92.17	90.84	**92.99**	86.55	89.15	**90.45**	83.44	92.27	**100**	82.28	91.65	98.98	90.9

- **NN** - Nearest Neighbor classifier (no adaptation).
- **SVM** - Linear SVM classifier (no adaptation).
- **NCM** - Nearest Class Means classifier (DSCM with single domain, no adaptation).
- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen *et al*. ICML'12.
- **JDA** - Joint Distribution Adaptation, long *et al*. ICCV'14.
- **ATTM** - Adaptive Transductive Transfer Machines, Farajidavar *et al*., BMVC'14.

Interpolating between Domains[40] (DLID)

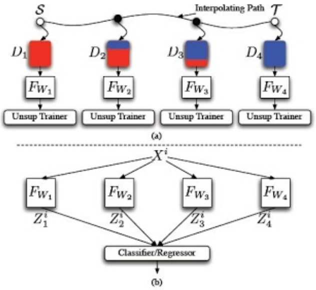


Image: Courtesy to S. Chopra

- generate intermediate datasets D_p by mixing target and source
- unsupervised deep nonlinear extractor $\mathbf{F}_{\mathbf{W}_p}$ learned on
- train classifiers on the concatenated features $\mathbf{Z}_p^i = \mathbf{F}_{\mathbf{W}_i}(\mathbf{X}^i)$.

[40] **S. Chopra *et al*. Deep Learning for domain adaptation by Interpolating between Domains, RL-WS ICML13.**

Deep Domain Confusion [41] (DDC)

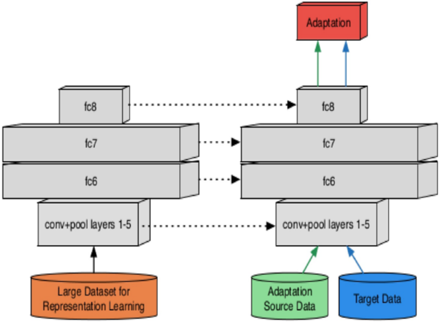


Image: Courtesy to E. Tzeng

- Minimizing the loss

$$\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda\,\mathrm{MMD}^2(X_S, X_T)$$

[41] **E. Tzeng *et al.*, Deep Domain Confusion: Maximizing for Domain Invariance, CoRR 14.**

Deep Adaptation Networks[42] (DAN)

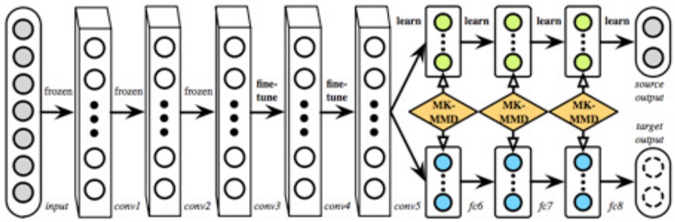


Image: Courtesy to M. Long

- first 3 convolutional layers are kept frozen
- next 2 convolutional layers refined with the current dataset
- deeply adapt fc6-fc8 using Multiple Kernel MMD

[42] **M. Long *et al.*, Learning Transferable Features with Deep Adaptation Networks, CoRR'15.**

Domain Adaptation by Backpropagation[43] (DAB)

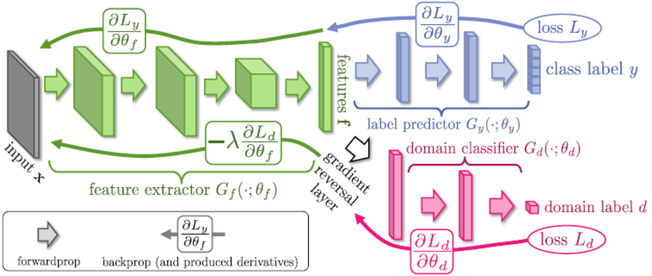


Image: Courtesy to Y. Ganin.

- L_y is the loss for label prediction (e.g. multinomial),
- L_d is the loss for the domain classification (e.g. logistic),

[43] **Y. Ganin and V. Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML 15.**

Results on OFF31 (US *sse*)

	A ->W	D ->W	W->D	Avg
SMDA	69.8	94.44	97.28	87.17
DLID	51.9	78.2	89.9	73.33
DDC	60.5	94.8	98.5	84.6
DAN	64.5	95.2	98.6	86.1
DAB	**73**	**96.4**	**99.2**	**89.53**

- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen *et al*. ICML'12.
- **DLID** - Interpolating between Domains Chopra *et al*. RL-WS ICML13.
- **DDC** - Deep Domain Confusion, Tzeng *et al*. CoRR'14.
- **DAN** - Deep Adaptation Networks, Long *et al*., CoRR'15.
- **DAB** - Domain Adaptation by Backpropagation, Ganin and Lempitsky, ICML'15.

Outline

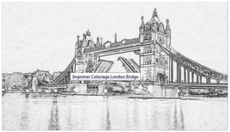
1. Introduction
   - Benchmark Datasets
2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods
3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation
4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives

Conclusion

- Many method exploits the Maximum Mean Discrepancy (MMD)
- Most popular methods are based on feature transform
  - Manifold based methods (GFS,GFK, BGFK)
  - Unsupervised subspace alignment (SA, LSSA)
  - Stacked marginalized denoising autoencoders (SMDA)
- Best performing methods exploit jointly instance selection, feature transform and parameter adaptation
  - Adaptive Transductive Transfer Machines (ATTM)
  - Statistically Invariant Embedding (SIE-CC)
- DeCAF features yield to significantly better results
  - Adding adaptation methods can further improve the results.
  - Using Deep Learning to perform adaptation performs the best.

More challenging transfer problems

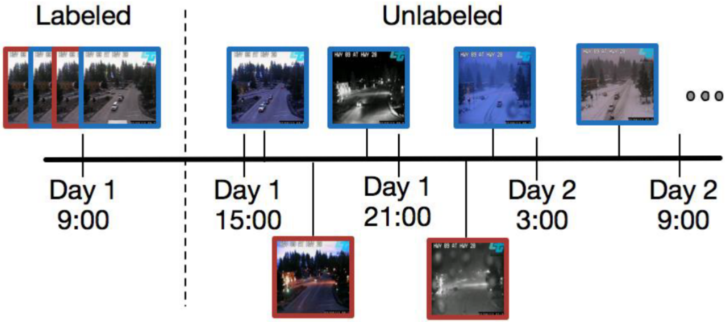




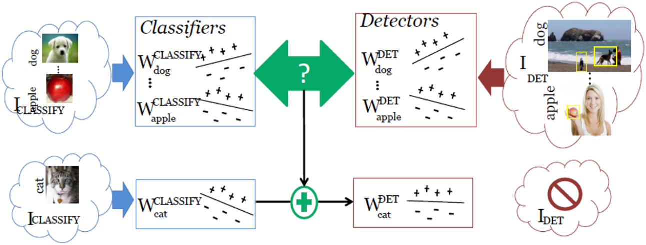




Continuous adaptation



Adapting object detection[44]



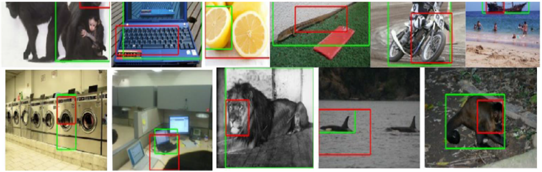
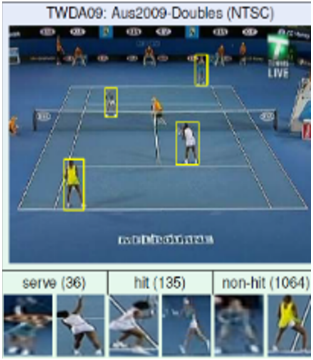


Image: Courtesy to T. Hoffman.

[44] **Hoffman *et al*. LSDA: Large Scale Detection Through Adaptation, NIPS'14.**

Adapting video action recognition[45]



- adapting between single and double, between tennis and badmington

[45] N. FarajiDavar *et al.*, Domain Adaptation in the Context of Sport Video Action Recognition, NIPSW DA , 2011. `http://cvssp.org/acasva/Downloads.html`

Thank you !