# Adaptation de domaine pour l'apprentissage multi-vues

Quentin VACHERON *supervisé par* Riikka HUUSARI

**Domain Adaptation via Transfer Component Analysis**
*Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, Qiang Yang*

**Fast and Provably Effective Multi-view Classification with Landmark-based SVM**
*Valentina Zantedeschi, Rémi Emonet, Marc Sebban*

# Supports

**Transfer Learning via dimensionality reduction**
*Sinno Jialin Pan, James T. Kwok, Qiang Yang*

**Covariate shift by kernel mean matching**
*Arthur Gretton, Alex Smola, Jiayung Huang*

# Plan

1) L’adaptation de domaine

    La méthode Transfer Component Analysis (TCA)

2) L’apprentissage multi-vues

    Unification par Landmark features

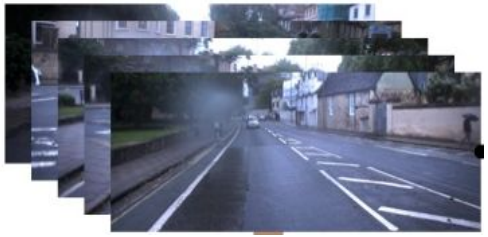
3) Combiner les 2 méthodes ?

4) Experiments

# 1) L'adaptation de domaine

Distribution données entraînement ≠ Distribution données test

*Exemple* : peu de labels sur le domaine d'intérêt



transfer learning

Domaine "ressemblant"
Beaucoup de données d'entraînements
**Source domain S**

Domaine d'intérêt
Peu de données d'entraînements
**Target domain T**

# 1) L'adaptation de domaine

Distance entre distributions :

- Divergence de Kullback-Leibler : $D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)}\, dx$
- Maximum Mean Discrepancy (MMD) : $\|\mu_P - \mu_Q\|_H = \|\mathbb{E}_P[\phi(x)] - \mathbb{E}_Q[\phi(z)]\|_H$

$$\approx \|\frac{1}{n_1}\sum_{i=1}^{n1}\phi(x) - \frac{1}{n_2}\sum_{i=1}^{n2}\phi(z)\|_H$$

$$Dist(\mathrm{X},\mathrm{Y}) = \|\frac{1}{n_1}\sum_{i=1}^{n_1}\phi(x_i) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(y_i)\|_{\mathcal{H}} \quad \text{car} \quad \|\mu_P - \mu_Q\|_H = 0 \quad \text{iff} \quad \mathbb{P} = \mathbb{Q}$$

# 1) L'adaptation de domaine

Plongement de S et T :

$$D_s = \left\{ (x_{S_1}, y_{S_1}), \dots, (x_{S_{n_1}}, y_{S_{n_1}}) \right\} \quad P(X_S)$$
$$D_T = \left\{ x_{T_1}, \dots, x_{T_{n_2}} \right\} \quad Q(X_T)$$

$P \neq Q$ mais on suppose $P(Y_S \mid \phi(X_S)) = P(Y_T \mid \phi(X_T))$

- Kernel Mean Matching (KMM) : $$\| \frac{1}{n_1} \sum_{i=1}^{n1} \beta_i \phi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n2} \phi(z_i) \|$$

- **Transfer Component Analysis (TCA)**

# 1) L'adaptation de domaine

*Note :* $Dist(\mathrm{X},\mathrm{Y}) = \| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(x_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(y_i) \|_{\mathcal{H}} = \mathrm{tr(KL)}$

avec $K = \begin{bmatrix} K_{src,src} & K_{src,tar} \\ K_{tar,src}^T & K_{tar,tar} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$

$$L_{ij} = \begin{cases} \frac{1}{n_1^2} & x_i, x_j \in X_{src} \\ \frac{1}{n_2^2} & x_i, x_j \in X_{tar} \\ -\frac{1}{n_1 n_2} & \text{otherwise.} \end{cases}$$

# Transfer Component Analysis (TCA)

$K = (KK^{-1/2})(K^{-1/2}K)$ kernel empirical map

On introduit $\widetilde{W}$ de taille $(n_1+n_2)\times m$

$$\widetilde{K} = (KK^{-1/2}\widetilde{W})(\widetilde{W}^\top K^{-1/2}K) = KWW^\top K \qquad W = K^{-1/2}\widetilde{W} \in \mathbb{R}^{(n_1+n_2)\times m}$$

$$\text{Dist}(X_S, X_T) = \text{tr}((KWW^\top K)L) = \text{tr}(W^\top KLKW)$$

L'apprentissage du noyau se réduit à :

$$\begin{aligned} \min_W \quad & \text{tr}(W^\top W) + \mu\, \text{tr}(W^\top KLKW) \\ \text{s.t.} \quad & W^\top KHKW = I, \end{aligned}$$

$$\Longleftrightarrow \max_W \; tr((W^\top(I + \mu KLK)W)^{-1}W^\top KHKW)$$

# Transfer Component Analysis (TCA)

$K = (KK^{-1/2})(K^{-1/2}K)$ kernel empirical map

On introduit $\widetilde{W}$ de taille $(n_1+n_2)\times m$

$$\widetilde{K} = (KK^{-1/2}\widetilde{W})(\widetilde{W}^\top K^{-1/2}K) = KWW^\top K \qquad W = K^{-1/2}\widetilde{W} \in \mathbb{R}^{(n_1+n_2)\times m}$$

$$\text{Dist}(X_S, X_T) = \text{tr}((KWW^\top K)L) = \text{tr}(W^\top KLKW)$$

L'apprentissage du noyau se réduit à :

$$\begin{aligned} \min_W \quad & \text{tr}(W^\top W) + \mu\,\text{tr}(W^\top KLKW) \\ \text{s.t.} \quad & W^\top KHKW = I, \end{aligned}$$

$O(m(n_1+n_2)^2)$ $\Longleftrightarrow$ $\max_W tr((W^\top(I+\mu KLK)W)^{-1}W^\top KHKW)$

# 2) L'apprentissage multi-vues

$x = \{$   To boldly go where no man has gone before. $\}$

3 vues
3 espaces
3 distributions

Approche 1-view ? ➔ early/late fusion ➔ insatisfaisant

# Landmark features

**Idée :** unifier l'information dans un espace latent commun à toutes les vues ➔ 1 modèle

L éléments $l$ choisis au hasard dans X (landmarks)

Pour chaque x ∈ X :

L landmarks →

$$\mu(x) = [\ \mu_1(x^1, l_1^1), \mu_1(x^1, l_2^1), \mu_1(x^1, l_3^1), \dots, \mu_1(x^1, l_L^1),$$ vue 1

$$\mu_2(x^2, l_1^2), \mu_2(x^2, l_2^2), \mu_2(x^2, l_3^2), \dots, \mu_2(x^2, l_L^2),$$ vue 2

$$\mu_3(x^3, l_1^3), \mu_3(x^3, l_2^3), \mu_3(x^3, l_3^3), \dots, \mu_3(x^3, l_L^3),$$ ⋮

$$\mu_v(x^v, l_1^v), \mu_v(x^v, l_2^v), \mu_v(x^v, l_3^v), \dots, \mu_v(x^v, l_L^v)\ ]$$ vue v

↓ v views

# Landmark features

**Idée :** unifier l'information dans un espace latent commun à toutes les vues ➔ 1 modèle

L éléments $l$ choisis au hasard dans X (landmarks)

Pour chaque x ∈ X :

L landmarks →

$$\mu(x) = [\ \mu_1(x^1, l_1^1), \mu_1(x^1, l_2^1), \mu_1(x^1, l_3^1), \dots, \mu_1(x^1, l_L^1),$$ vue 1

$$\mu_2(x^2, l_1^2), \mu_2(x^2, l_2^2), \mu_2(x^2, l_3^2), \dots, \mu_2(x^2, l_L^2),$$ vue 2

$$\mu_3(x^3, l_1^3), \mu_3(x^3, l_2^3), \mu_3(x^3, l_3^3), \dots, \mu_3(x^3, l_L^3),$$ ⋮

$$\mu_v(x^v, l_1^v), \mu_v(x^v, l_2^v), \mu_v(x^v, l_3^v), \dots, \mu_v(x^v, l_L^v)\ ]$$ vue v

↓ v views

" 1 vue dans $\mathbb{R}^{LV}$ "

# Landmark features

Provient de l’algorithme MVL-SVM, qui a :

Stabilité uniforme

Erreur en généralisation

Bon temps d’entraînement (% L)

# 3) Combiner les deux méthodes ?

Données multi-vues *étiquetées* (**Source S**)

Données multi-vues *non-étiquetées* (**Target T**)

Unification - Landmark features

Adaptation de domaine - TCA

Classification de **T** avec $\widetilde{K}$

# 4) Experiments

**Les landmark features sont-ils adaptés à l'adaptation de domaine dans le cas de données multi-vues ?**

# Office Caltech Dataset

***3 domaines***

Amazon

Dslr

Webcam

***10 catégories***

....

***2 vues***

“decaf6” *(dim 800)*

“surf” *(dim 800)*

# Protocole

Source S
labels

2 vues

“ decaf6 “
“ surf ”

μ

Landmark features $\mu(x)$
“1 view”

TCA $\widetilde{K}$

svm.fit( $\widetilde{K}_{src,src}$ , y)

svm.predict( $\widetilde{K}_{tar,src}$ )

Target T
no labels

2 vues

“ decaf6 “
“ surf ”

μ

Landmark features $\mu(x)$
“1 view”

# Résultats provisoires

- Score ( landmarks sans TCA ) = 13,00%
- Score ( landmarks avec TCA ) = 11,89%
- Score ( landmarks sans TCA) > Score ( landmarks avec TCA)  ...

# Résultats

## Alors ?

**Les landmark features sont-ils adaptés à l'adaptation de domaine dans le cas de données multi-vues ?**

# Résultats

## Alors ?

**Les landmark features sont-ils adaptés à l'adaptation de domaine dans le cas de données multi-vues ?**

*Trop tôt pour conclure :*

- Bugs
- Dataset
- “surf”, “decaf6”
- Noyaux utilisés

# Résultats

**Alors ?**

**Les landmark features sont-ils adaptés à l'adaptation de domaine dans le cas de données multi-vues ?**

**Merci !**

$$
\begin{aligned}
dist(X_S, X_T) &= \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S) - \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T) \right\|^2 \\
&= \left\langle \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S) - \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T), \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S) - \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T) \right\rangle \\
&= \left\langle \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S), \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S) \right\rangle + \left\langle \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T), \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T) \right\rangle \\
&\quad - 2 \left\langle \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S), \frac{1}{n_T} \sum_{j=1}^{n_T} \phi(x_j^T) \right\rangle \\
&= \frac{1}{n_S^2} \sum_{i,j=1}^{n_S} k(x_i^S, x_j^S) + \frac{1}{n_T^2} \sum_{i,j=1}^{n_T} k(x_i^T, x_j^T) - \frac{2}{n_S n_T} \sum_{i=1}^{n_S} \sum_{j=1}^{n_T} k(x_i^S, x_j^T)
\end{aligned}
$$

$$
\begin{aligned}
dist(X_S, X_T) &= tr(KL) = tr \left( \begin{bmatrix} K_{SS} & K_{ST} \\ K_{TS} & K_{TT} \end{bmatrix} \begin{bmatrix} [\frac{1}{n_S^2}] & [\frac{-1}{n_S n_T}] \\ [\frac{-1}{n_T n_S}] & [\frac{1}{n_T^2}] \end{bmatrix} \right) \\
&= tr \left( \begin{bmatrix} [\frac{1}{n_S^2}]K_{SS} + [\frac{-1}{n_T n_S}]K_{ST} & [\frac{-1}{n_S n_T}]K_{SS} + [\frac{1}{n_T^2}]K_{ST} \\ [\frac{1}{n_S^2}]K_{TS} + [\frac{-1}{n_T n_S}]K_{TT} & [\frac{-1}{n_S n_T}]K_{TS} + [\frac{1}{n_T^2}]K_{TT} \end{bmatrix} \right) \\
&= tr \left( [\frac{1}{n_S^2}]K_{SS} + [\frac{-1}{n_T n_S}]K_{ST} \right) + tr \left( [\frac{-1}{n_S n_T}]K_{TS} + [\frac{1}{n_T^2}]K_{TT} \right) \\
&= tr \left( [\frac{1}{n_S^2}]K_{SS} \right) + tr \left( [\frac{1}{n_T^2}]K_{TT} \right) - 2tr \left( [\frac{1}{n_T n_S}]K_{ST} \right) \\
&= \sum_{i,j=1}^{n_S} \frac{1}{n_S^2} k(x_i^S, x_j^S) + \sum_{i,j=1}^{n_T} \frac{1}{n_T^2} k(x_i^T, x_j^T) - 2 \sum_{i=1}^{n_S} \sum_{j=1}^{n_T} \frac{1}{n_S n_T} k(x_i^S, x_j^T).
\end{aligned}
$$