# Data Science

## Apprentissage par transfert

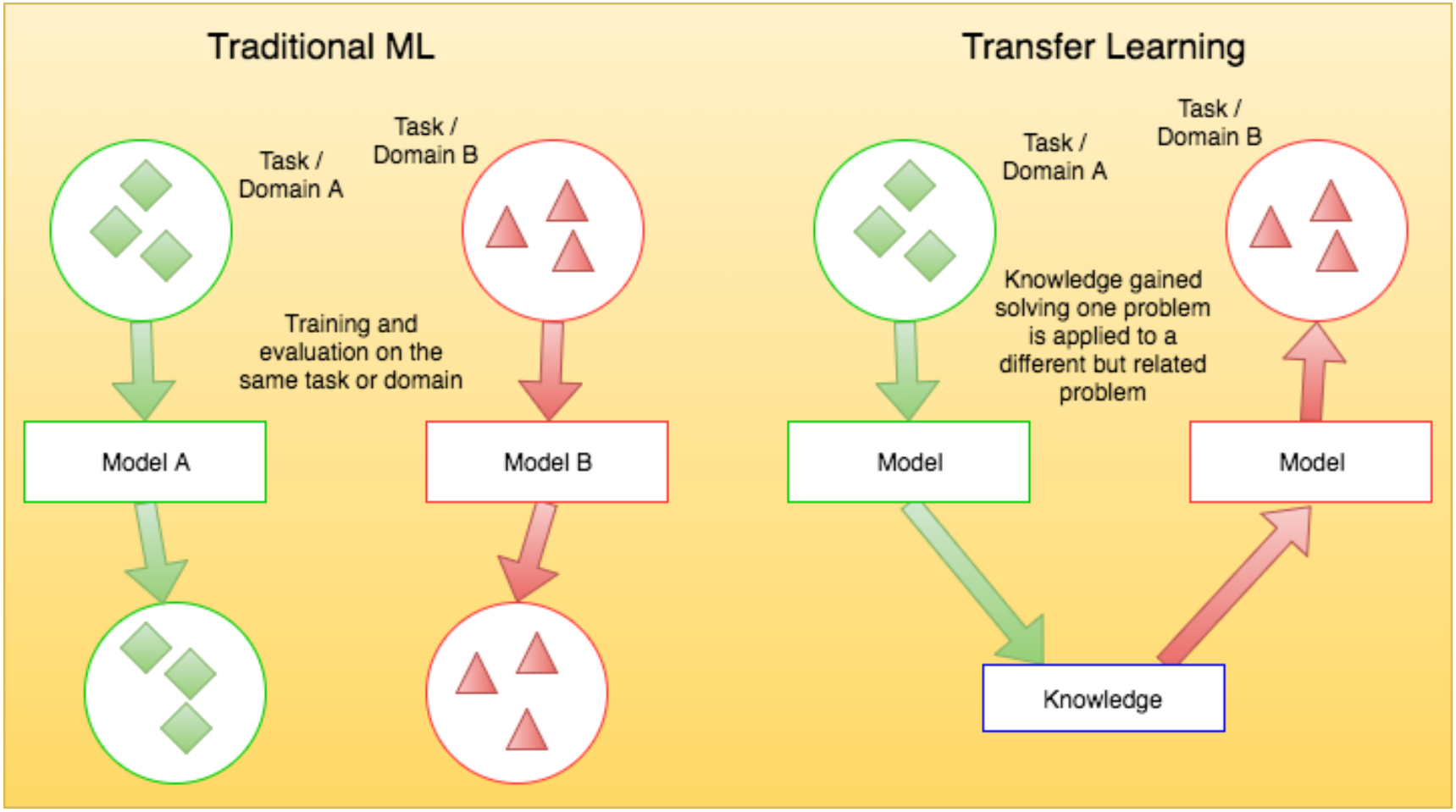
Hachem Kadri

2018-2019

# Apprentissage par transfert

- Dans des contextes réels, le meilleur jeu de données pour la tâche d'apprentissage n'est pas toujours disponible

# Apprentissage par transfert




from `http://experiencesutra.com/experiments/deep-learning-in-fashion`

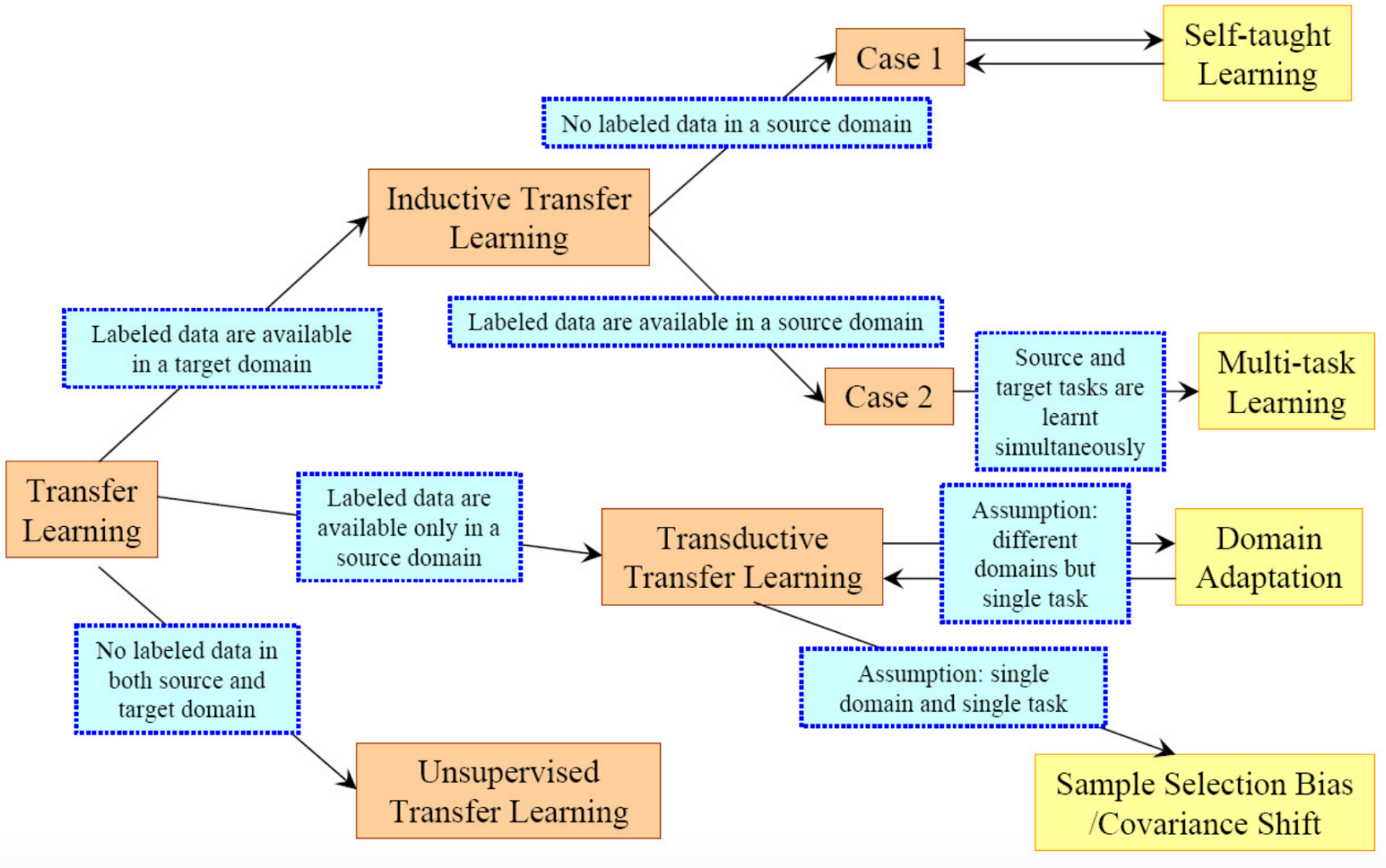
# Apprentissage par transfert

- Un domaine source $\mathcal{D}_S$ et une tâche source $\mathcal{T}_S$
- Un domaine cible $\mathcal{D}_T$ et une tâche source $\mathcal{T}_T$

Apprentissage par transfert a pour objectif de d'améliorer l'apprentissage de la fonction cible utilisant les connaissances de $\mathcal{D}_S$ et $\mathcal{T}_S$ dans les cas où $\mathcal{D}_S \neq \mathcal{D}_T$ et/ou $\mathcal{T}_S \neq \mathcal{T}_T$.

- Généralement, d'autres contraintes sur le domaine cible sont imposées, par exemple peu ou pas d'étiquettes
- Dans le cas où $\mathcal{D}_S = \mathcal{D}_T$ et/ou $\mathcal{T}_S = \mathcal{T}_T$, on retrouve la cas classique d'apprentissage supervisé

# Apprentissage par transfert



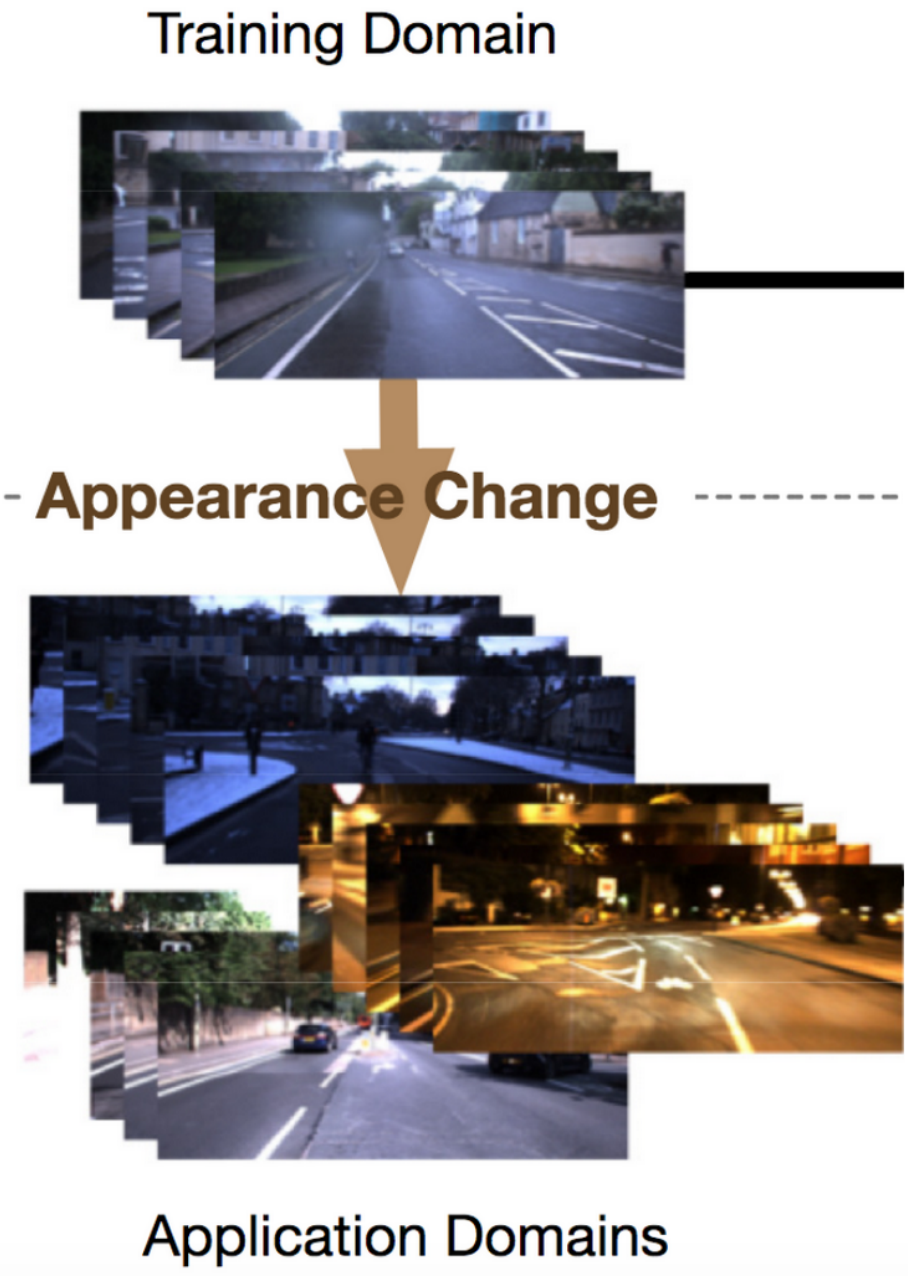

"survey on Transfer Learning" [Pan and Yang, TKDE 2010]

# Apprentissage par transfert

- Même tâche, domaines différents $\Rightarrow$ Adaptation de domaine
- Même domaine, plusieurs tâches $\Rightarrow$ Apprentissage multi-tâche

# Adaptation de domaine

# Adaptation de domaine



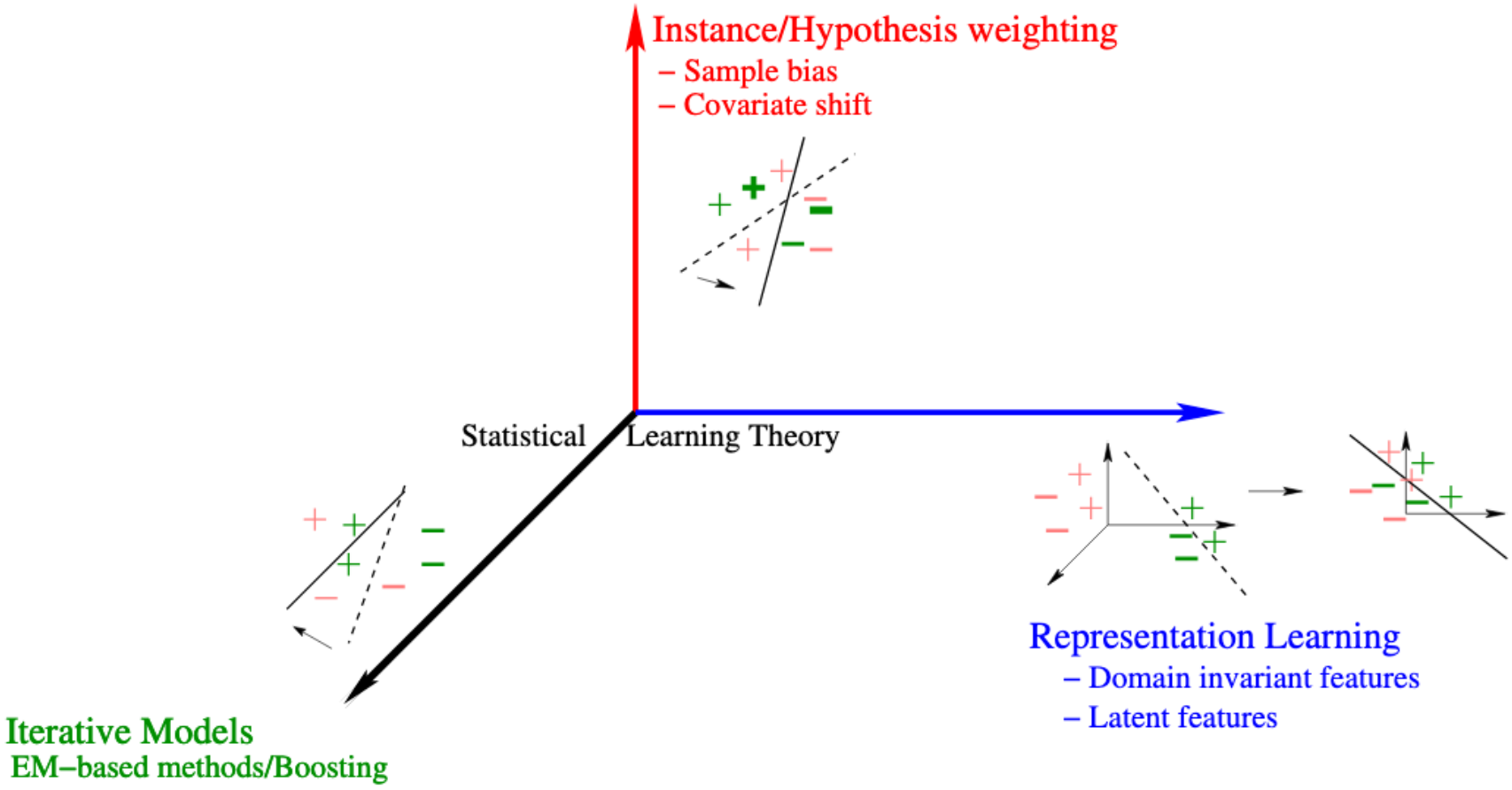
# Adaptation de domaine

- un espace de features $\mathcal{X}$, un ensemble de labels $\mathcal{Y} = \{-1, 1\}$
- une distribution source $P_s$, une distribution cible $P_t$
- une fonction inconnue $f : \mathcal{X} \to \mathcal{Y}$ selaon $P_t(y|x)$
- un ensemble de données d'apprentissage $S_s = \{(x_i, y_i)\}$généré i.i.d selon $P_s$, un ensemble de données test $S_t = \{x_i\}$ généré selon $D_t$, la loi marginale de $P_t$ sur $\mathcal{X}$
- apprendre un classifieur $h$ le plus proche possible de la fonction inconnue $f$
- risque réel (source) : $R_s(h) = E_{(x,y)\sim P_s} \mathrm{I}[h(x) \neq y)]$
  risque empirique sur $S_s$ : $\hat{R}_s(h) = \sum_{(x,y)\in S_s} \mathrm{I}[h(x) \neq y)]$
- risque réel (cible) : $R_t(h) = E_{(x,y)\sim P_t} \mathrm{I}[h(x) \neq y)]$

Borne de généralisation : $R_s(h) \leq \hat{R}_s(h) + \sqrt{\frac{complexite(h \in \mathcal{H})}{|S_s|}}$

$\Rightarrow$ guaranties théoriques pour le domaine cible ?

# Adaptation de domaine - Stratégies




from https://limos.fr/media/uploads/seminaire/hebrard_slides_Nov18.pdf

# Adaptation de domaine - par repondération

$$\epsilon_T(h) = \mathop{\mathbf{E}}_{(\mathbf{x}^t,y^t)\sim P_T} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

$$= \mathop{\mathbf{E}}_{(\mathbf{x}^t,y^t)\sim P_T} \frac{P_S(\mathbf{x}^t,y^t)}{P_S(\mathbf{x}^t,y^t)} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

$$= \sum_{(\mathbf{x}^t,y^t)} P_T(\mathbf{x}^t,y^t) \frac{P_S(\mathbf{x}^t,y^t)}{P_S(\mathbf{x}^t,y^t)} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

$$= \mathop{\mathbf{E}}_{(\mathbf{x}^t,y^t)\sim P_S} \frac{P_T(\mathbf{x}^t,y^t)}{P_S(\mathbf{x}^t,y^t)} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

**Assume similar tasks - covariate shift**, $P_S(y|\mathbf{x}) = P_T(y|\mathbf{x})$, then:

$$= \mathop{\mathbf{E}}_{(\mathbf{x}^t,y^t)\sim P_S} \frac{D_T(\mathbf{x}^t)P_T(y^t|\mathbf{x}^t)}{D_S(\mathbf{x}^t)P_S(y^t|\mathbf{x}^t)} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

$$= \mathop{\mathbf{E}}_{(\mathbf{x}^t,y^t)\sim P_S} \frac{D_T(\mathbf{x}^t)}{D_S(\mathbf{x}^t)} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

from https://limos.fr/media/uploads/seminaire/hebrard_slides_Nov18.pdf

# Adaptation de domaine - par repondération

- covariate shift

$$R_t(h) = E_{(x^t,y^t)\sim P_s} \frac{D_t(x^t)}{D_s(x^t)} \mathrm{I}[h(x^t) \neq y^t]$$

- erreur de pondération dans le domaine source : $\omega(x^t) = \frac{D_t(x^t)}{D_s(x^t)}$
- repondérer les données sources étiquetées :

$$\sum_{(x,y)\in S_s} \hat{\omega}(x_i^s)\mathrm{I}[h(x^s) \neq y^s],$$

avec $\hat{\omega}$ une estimation de $\omega$

# Adaptation de domaine - par repondération

## Estimateurs de densité

Construire des estimateurs de densité pour les domaines source et cible et estimer le ratio

- $\hat{\omega}(x) = \sum_l \alpha_l \psi_l(x)$
- apprendre $\alpha$ : $\arg\min_\alpha KL(\hat{\omega} D_s, D_t)$

## Kernel mean matching

Matcher les distributions avec des noyaux

- $MMD(P_s, P_t) = \|1/n_s \sum_{i=1}^{n_s} \phi(x_i^s) - 1/n_t \sum_{i=1}^{n_t} \phi(x_i^t)\|_{\mathcal{H}}$
- $\min_\beta MMD(\beta(x)P_s(x), P_t)$ $s.t.$ $\beta(x) > 0$ et $E_{P_s}[\beta(x)] = 1$

# Apprentissage multi-tâche

# Apprentissage multi-tâche

- Apprendre plusieurs tâches simultanément
- . . . au lieu d'apprendre les tâches indépendamment
- Améliorer l'apprentissage et la prédiction en tenant en compte les dépendences entre les tâches

# Apprentissage supervisé

- Classification $f : X \to Y \subseteq \{-1, 1\}$
- Régression $f : X \to Y = \mathbb{R}$

Autres tâches ?

- Multi-classe $f : X \to Y = \{1, 2, \ldots \mathrm{T}\}$
- Régression à valeur vectorielle $f : X \to Y \subseteq \mathbb{R}^{\mathrm{T}}$
- ...

# Apprentissage supervisé multi-tâche

Données d'apprentissage

$$S_1 = (x_i^1, y_i^1)_{i=1}^{n_1}, \ldots, S_{\mathrm{T}} = (x_i^{\mathrm{T}}, y_i^{\mathrm{T}})_{i=1}^{n_{\mathrm{T}}}$$

Apprendre

$$f_1 : X_1 \to Y_1, \ldots, f_{\mathrm{T}} : X_{\mathrm{T}} \to Y_{\mathrm{T}}$$

- Régression à valeur vectorielle

  $$S_n = (x_i, y_i)_{i=1}^{n}, \quad x_i \in X, \quad y_i \in \mathbb{R}^{\mathrm{T}}$$

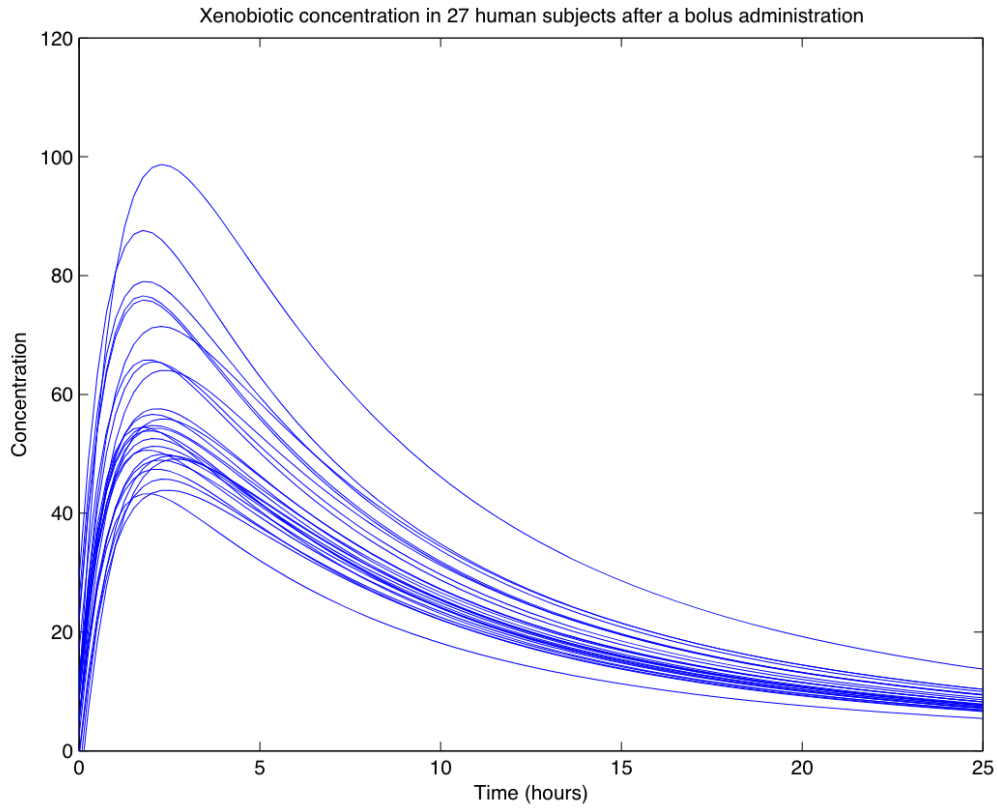
  Multi-tâche avec les mêmes données d'entrée ! Chaque composante du vecteur de sortie est une "tâche"

- Multi-classe

  $$S_n = (x_i, y_i)_{i=1}^{n}, \quad x_i \in X, \quad y_i \in \{1, \ldots, \mathrm{T}\}$$

# Pourquoi apprendre plusieurs tâches simultanément ?

- Cas de la régression multiple

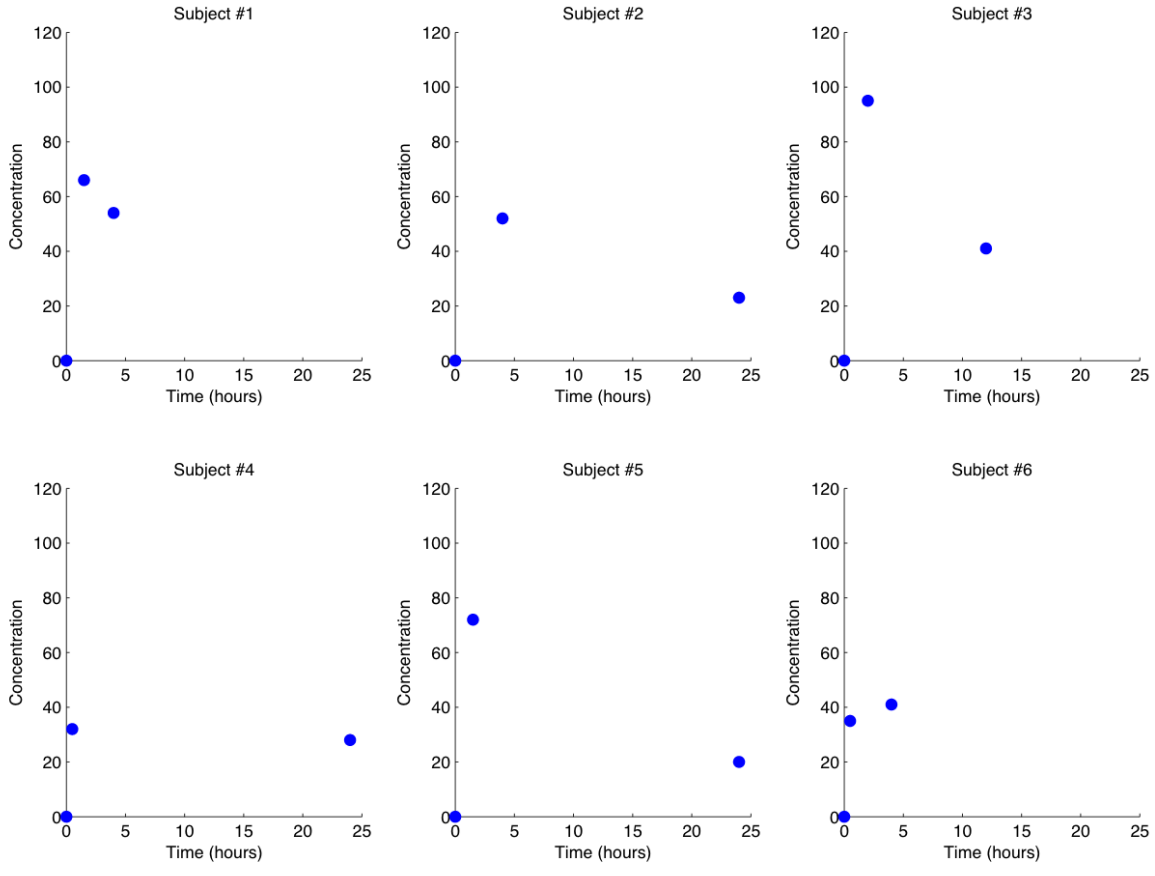



from [Dinuzzo et al., 2013]

- Les courbes de réponses ont des allures similaires
- Il y a une variabilité inter-individuelle

# Pourquoi apprendre plusieurs tâches simultanément ?

- Cas de la régression multiple

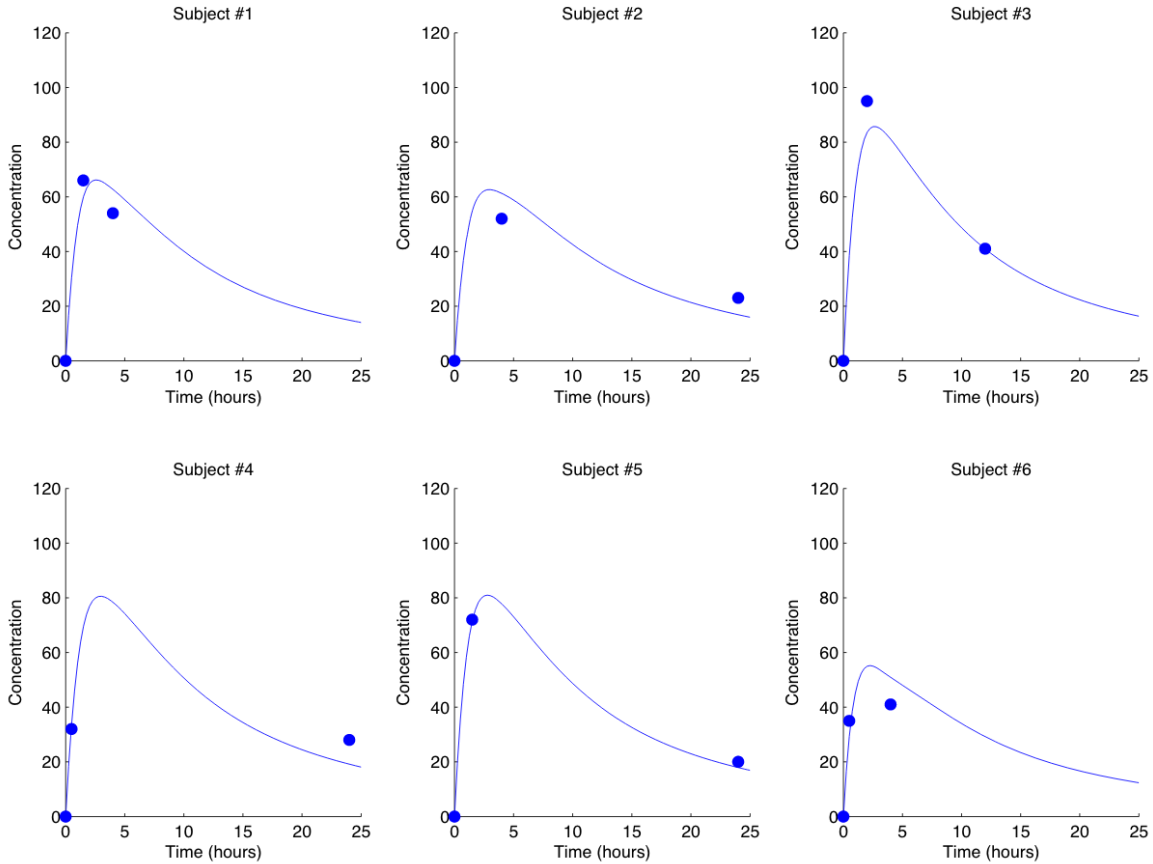



from [Dinuzzo et al., 2013]

- Peu de données par sujet et différents échantiollonages

# Pourquoi apprendre plusieurs tâches simultanément ?

- Cas de la régression multiple




from [Dinuzzo et al., 2013]

- Combiner les données pour mieux estimer toutes les courbes

# Apprentissage multi-tâche régularisé

$$\mathrm{err}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T}) + \mathrm{pen}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T})$$

On considère des modèles linéaires

$$f_1(x) = \mathrm{w}_1^\top x, \ldots, f_\mathrm{T}(x) = \mathrm{w}_\mathrm{T}^\top x$$

# Erreur empirique

$$\mathrm{err}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T}) = \sum_{t=1}^{\mathrm{T}} \frac{1}{n_t} \sum_{i=1}^{n_t} (y_i^t - \mathrm{w}_t^\top x_i^t)^2$$

- Il est possible de consider d'autres fonctions coûts

# Erreur empirique

$$\mathrm{err}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T}) = \sum_{t=1}^{\mathrm{T}} \frac{1}{n_t} \sum_{i=1}^{n_t} (y_i^t - \mathrm{w}_t^\top x_i^t)^2$$

- Cas de la régression à valeur vectorielle

$$S_n = (x_i, y_i)_{i=1}^n, \quad x_i \in X, \quad y_i \in \mathbb{R}^\mathrm{T}$$

$$\mathrm{err}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T}) = \frac{1}{n} \sum_{t=1}^{\mathrm{T}} \sum_{i=1}^{n} (y_i^t - \mathrm{w}_t^\top x_i)^2 = \frac{1}{n} \|Y - XW\|^2$$

  - $Y = (y_i)_{i=1}^{\mathrm{n}} \in \mathbb{R}^{n \times \mathrm{T}}, \quad X = (x_i)_{i=1}^{\mathrm{n}} \in \mathbb{R}^{n \times d}, \quad W \in \mathbb{R}^{d \times \mathrm{T}}$

# Régularisation multi-tâche

$$\mathrm{pen}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T})$$

- Coupler les solutions obtenues pour chaque tâche par régularisation
- Exploiter la structure entre les tâches

# Régularisation multi-tâche

$$\mathrm{pen}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T}) = \sum_{t=1}^{\mathrm{T}} \|\mathrm{w}_t\|^2$$

- Indépendence entre tâches !

$$\begin{aligned}
\min_{w_1,\ldots,w_\mathrm{T}} \frac{1}{n} \sum_{t=1}^{\mathrm{T}} \sum_{i=1}^{n} (y_i^t - \mathrm{w}_t^\top x_i)^2 + \lambda \sum_{t=1}^{\mathrm{T}} \|\mathrm{w}_t\|^2 \\
= \sum_{t=1}^{\mathrm{T}} \Big( \min_{w_t} \sum_{i=1}^{n} (y_i^t - \mathrm{w}_t^\top x_i)^2 + \lambda \|\mathrm{w}_t\|^2 \Big)
\end{aligned}$$

# Régularisation multi-tâche

$$\mathrm{pen}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T}) = \mathit{rank}(W)$$

- Régression de faible rang

$$\min_{W \in \mathbb{R}^{d \times \mathrm{T}}} \frac{1}{n} \| Y - XW \|^2$$
$$s.t. \quad \mathit{rank}(W) \leq r$$

# Régression de faible rang

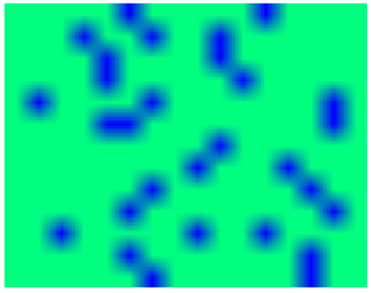
$$\hat{W}^* = \min_{W \in \mathbb{R}^{d \times \mathrm{T}}} \frac{1}{n} \| Y - XW \|^2$$
$$s.t. \quad rank(W) \leq r$$

1. Calculer la solution de la régression par moindres carrés (sans contraintes)
$$W^* = (X^\top X)^{-1} X^\top Y$$
2. Calculer l'estimation par moindres carrés
$$Y^* = XW^*$$
3. Calculer la SVD de $Y^*$ et la matrice de projection $P_r$
$$Y^* = UDV, \quad P_r = \sum_{i=1}^{r} v_i v_i^\top$$
4. Obtenir la solution de la régression de faible range
$$\hat{W}^* = W^* P_r$$

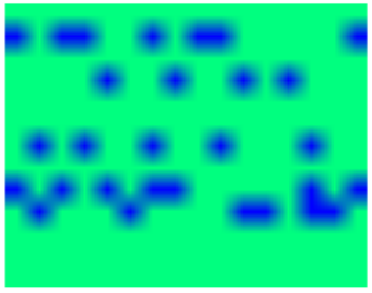
# Régularisation multi-tâche

$$\mathrm{pen}(\mathrm{w}_1, \ldots, \mathrm{w}_\mathrm{T})$$

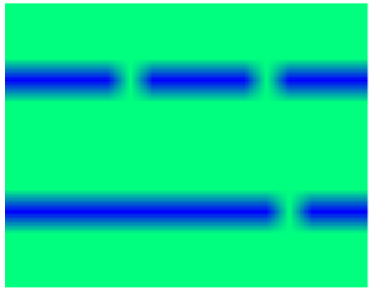
- Régularisation matricielle
  - rang
  - norme nucléaire : $\|W\|_* = \mathrm{trace}(\sqrt{W^\top W})$
  - norme $L_{2,1}$ : $\|W\|_{2,1} = \sum_{j=1}^{d} \sqrt{\sum_{t=1}^{\mathrm{T}} W_{jt}^2}$
  - ...



$\#\text{rows} = 13$
$\|\cdot\|_{2,1} = 19$
$\ell_1\text{-norm} = 29$



5
12
29



2
8
29