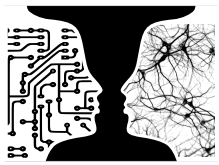
L'adaptation de domaine en apprentissage automatique

Hana Ajakan

Groupe de recherche en apprentissage automatique de l'Université Laval (GRAAL)



22 Avril 2016

Plan

Apprentissage automatique

D'après T. Mitchel,

> *Un programme apprend d'une expérience E à faire une tâche T en regard d'une mesure de performance P si sa performance à accomplir la tâche T s'améliore avec l'expérience E.*

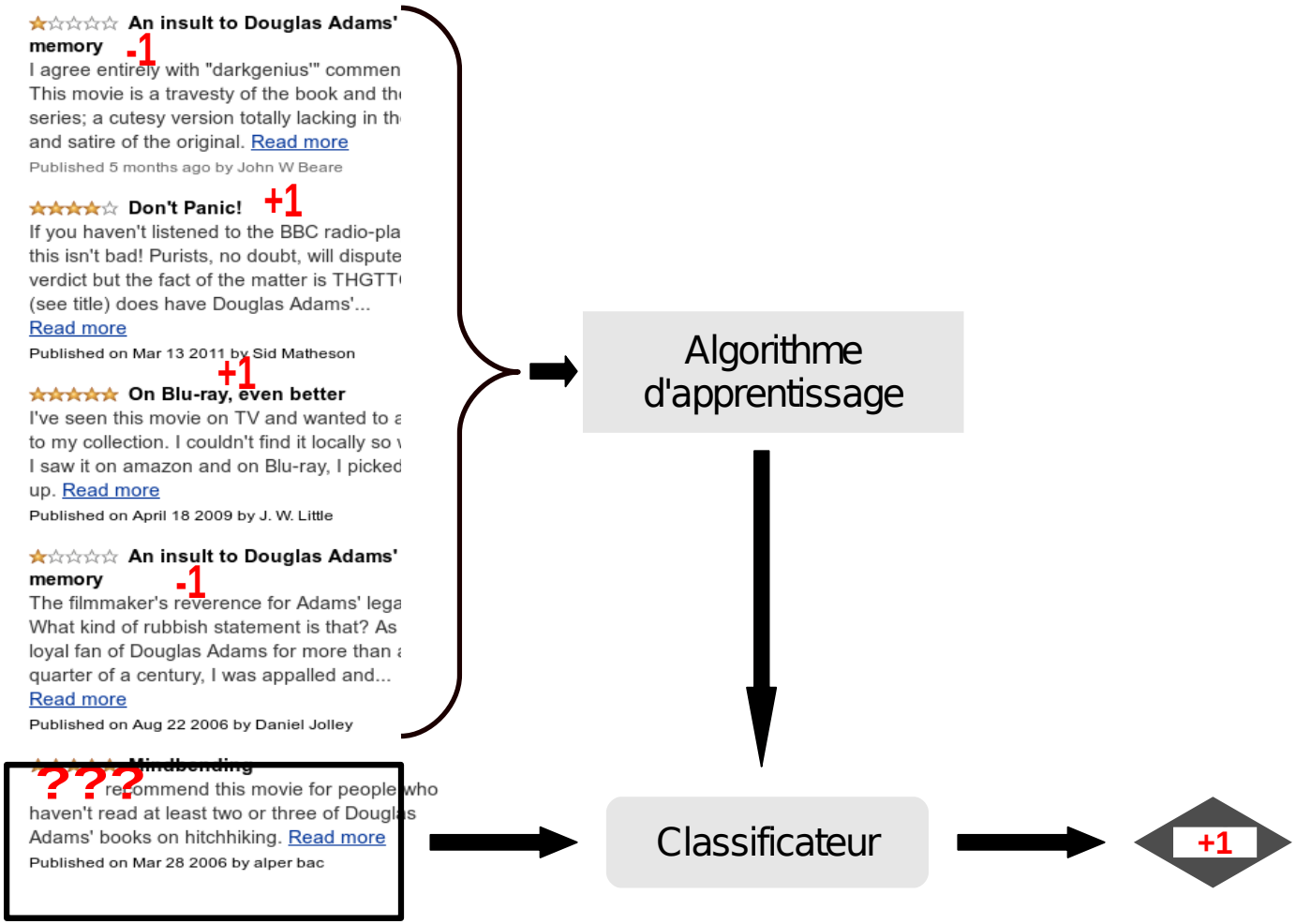
Exemple

Si on veut classifier des avis d'utilisateurs sur des produits trouvés sur le site *Amazon.com*.

- L'expérience E est que chaque utilisateur fournit son avis et exprime son sentiment (J'aime, Je n'aime pas) par rapport à un produit.
- La tâche T est que l'algorithme d'apprentissage va essayer de prédire le sentiment de l'utilisateur à partir de son avis.
- La mesure de performance P peut être le nombre de fois que l'algorithme a réussi à prédir le bon sentiment sur le nombre total de prédictions effectuées.

On dit que le programme a réussi si le sentiment prédit correspond au sentiment de l'utilisateur.

Avis sur les films



D'autres exemples...

- Détection de Spams.
- Classification de texte.
- Reconnaissance d'objets.
- Détection des applications malicieuses.
- Prédiction de la météo.
- Analyse du sentiment.

Description du problème et notations

Tâche de classification binaire

- Espace d'entrée: $\mathcal{X} \in \mathbb{R}^d$, Espace de sortie : $\mathcal{Y} = \{0, 1\}$.
- Une distribution $\mathcal{D}$ sur $\mathcal{X} \times \mathcal{Y}$, Ensemble de classificateurs: $\mathcal{H}$

Deux ensembles de données

- Un ensemble d'apprentissage $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m} \sim (\mathcal{D})^m$ tiré i.i.d de la distribution $\mathcal{D}$.
- Un ensemble de test $S' = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m'} \sim (\mathcal{D})^{m'}$ tiré i.i.d de la même distribution $\mathcal{D}$.

Tâche d'apprentissage

Minimiser le risque réel du classificateur h sur $\mathcal{D}$.

$$R_{\mathcal{D}}(h) \overset{\text{def}}{=} \Pr_{(\mathbf{x}_i, y_i))\sim\mathcal{D}}[h(\mathbf{x}_i) \neq y_i] .$$

$\Rightarrow$ IL existe plusieurs méthodes peut estimer ce risque.

Principale problématique

- Les données d'apprentissage et celles du test proviennent de deux distributions (domaines) différentes.
- Exemple : livres $\rightarrow$ films.
- Objectif: On doit adapter le modèle d'apprentissage du premier domaine (source) au deuxième domaine (cible)
  $\Rightarrow$ Adaptation de domaine.

Plan

Un sous domaine de l'apprentissage par transfert

Définition de l'apprentissage par transfert

L'apprentissage par transfert peut être vu comme la capacité d'un système à reconnaître et appliquer des connaissances et des compétences, apprises à partir de tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes.

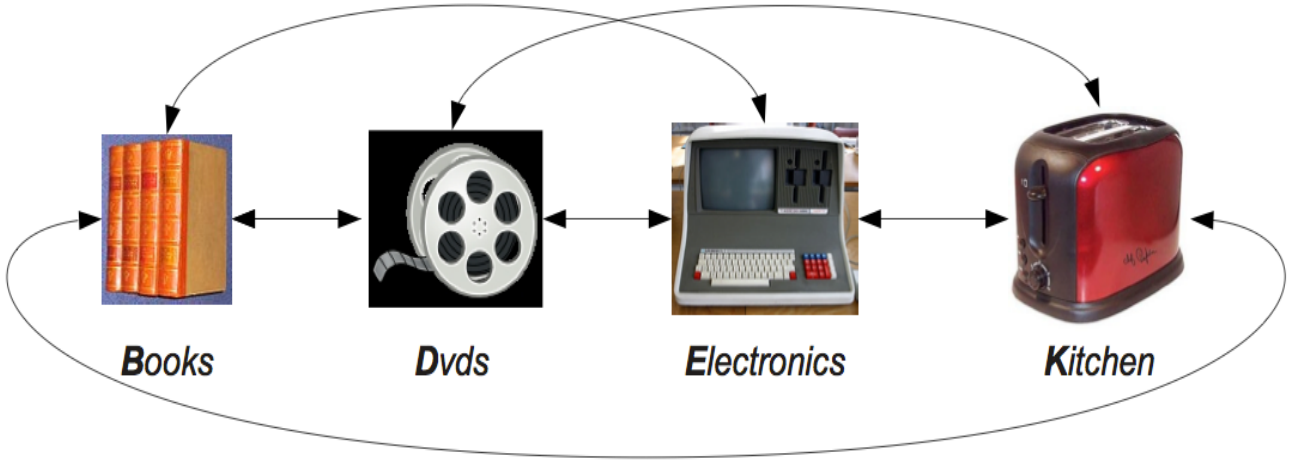
Quand est ce qu'on parle de l'adaptation de domaine?

Réponse: Lorsque la distribution d'apprentissage (domaine source) et la distribution de test (domaine cible) sont différentes mais similaires.

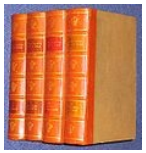
Motivation: Exemples

Domaine de photos étiquetées

Domaine de vidéos non étiquetées



Exemple



Avis sur les livres

??? **The end of the series.**
This book was written to provoke those who wanted Adams to continue the trilogy but I loved it. Aurthor setteled down on a bob fearing planet where he has aquired the prestigous...
Read more
Published on Mar 18 2002 by dan

??? **Mostly Harmless is Underrated**
I think most of the reviews for this book downplay it seriously. While the ending is kind of disappointing, the book overall is wonderful.
Read more
Published on Jan 22 2002 by A Big Adams Fan

??? **Please pretend this book was never written.**
I have long been a fan of the Hitchhikers series as they are comic genius. The book Mostly Harmless, however, should never have come about. It is frustration at its peak. Read more
Published on Jan 14 2002 by Paul Norrod

??? **Kinda like horror movies...**
...in that the last one usually isn't all that appealing. I liked it fine, with some of Adams's wit, but it was a bit disappointing. Read more
Published on Nov 4 2001 by Kristopher Vincent

??? **A Terrible End to A Great Series**
The ending for this books was so bad that I vowed never to read another Douglas Adams book. Adams was obviously sick and tired of the series and used this book to kill it off with...
Read more
Published on Oct 17 2001 by David A. Lessnau

Avis sur les films

-1 **An insult to Douglas Adams' memory**
I agree entirely with "darkgenius'" comments. This movie is a travesty of the book and the TV series; a cutesy version totally lacking in the wit and satire of the original. Read more
Published 5 months ago by John W Beare

+1 **Don't Panic!**
If you haven't listened to the BBC radio-play, this isn't bad! Purists, no doubt, will dispute my verdict but the fact of the matter is THGTTG (see title) does have Douglas Adams'...
Read more
Published on Mar 13 2011 by Sid Matheson

+1 **On Blu-ray, even better**
I've seen this movie on TV and wanted to add it to my collection. I couldn't find it locally so when I saw it on amazon and on Blu-ray, I picked it up. Read more
Published on April 18 2009 by J. W. Little

-1 **An insult to Douglas Adams' memory**
The filmmaker's reverence for Adams' legacy? What kind of rubbish statement is that? As a loyal fan of Douglas Adams for more than a quarter of a century, I was appalled and...
Read more
Published on Aug 22 2006 by Daniel Jolley

Algorithme d'apprentissage

Classificateur

-1

Description formelle du problème

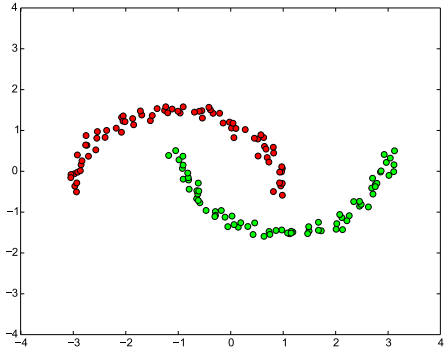
Tâche de classification binaire

- Espace d'entrée: $\mathcal{X} \in \mathbb{R}^d$
- Espace de sortie : $\mathcal{Y} = \{0, 1\}$
- ensemble de classificateurs: $\mathcal{H}$

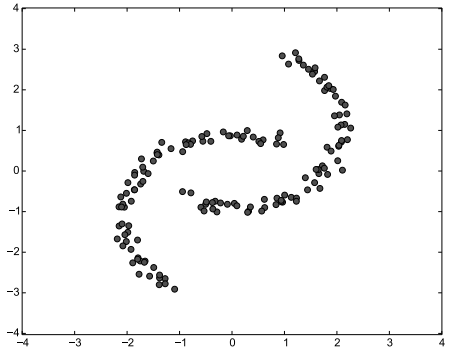
Deux différentes distributions

- Domaine source: $\mathcal{D}_S$
- Domaine cible: $\mathcal{D}_T$

l'ensemble **source étiqueté**
$S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^m \sim (\mathcal{D}_S)^m$,



l'ensemble **cible non-étiqueté**
$T = \{\mathbf{x}_i^t\}_{i=1}^m \sim (\mathcal{D}_T)^m$.



L'objectif est de construire un classificateur $\eta \in \mathcal{H}$ minimisant le risque cible:

$$R_{\mathcal{D}_T}(\eta) \stackrel{\text{def}}{=} \Pr_{(\mathbf{x}^t, y^t) \sim \mathcal{D}_T}[\eta(\mathbf{x}^t) \neq y^t] .$$

Problème difficile à résoudre!!!

Quelques hypothèse particulières:

- *Covariate shift*: Suppose que les deux domaines partagent la même fonction d'étiquetage. Autrement dit les distributions conditionnelles de $\mathcal{Y}$ sachant $\mathcal{X}$ sont les mêmes dans toutes les distributions:

$$\forall \mathbf{x} \in \mathcal{X},\ P_S(y|\mathbf{x}) = P_T(y|\mathbf{x}) \ \text{ mais: } \ D_S^{\mathcal{X}}(\mathbf{x}) \neq \mathcal{D}_T^{\mathcal{X}}(\mathbf{x})$$

- λ-shift: Suppose que les points proches ont la même étiquette. Il restreint le changement de la probabilité cible d'une étiquette, ce changement est d'au plus une proportion $1-\lambda$ de la probabilité source de l'étiquette. On dit que $\mathcal{D}_S$ et $\mathcal{D}_T$ sont liés par l'hypothèse du λ-shift si.

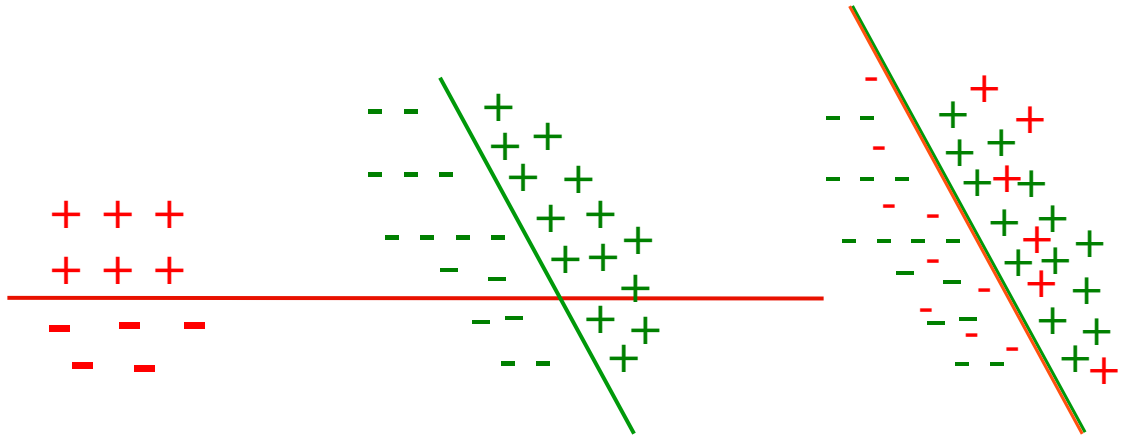
$$\forall y \in \mathcal{Y},\ P_S(y|\mathbf{x})(1-\lambda) \leq P_T(y|\mathbf{x}) \leq P_S(y|\mathbf{x})(1-\lambda) + \lambda$$

Quand est ce que c'est possible pour un classificateur de s'adapter d'un domaine à un autre?

Question

Si le classificateur η est appris sur le domaine source, quelle sera sa performance sur le domaine cible?

Réponse $\Rightarrow$ Si les deux domaines sont "proches", alors un classificateur ayant un risque source faible pourra avoir aussi un risque cible faible.



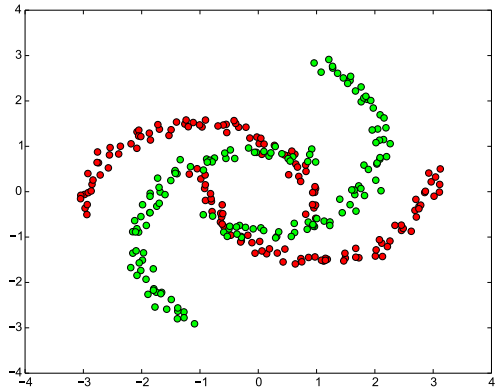
La Divergence entre les domaines source et cible

Definition (Ben David et al., 2006)

Étant donné deux distributions de probabilité $\mathcal{D}_S^X$ et $\mathcal{D}_T^X$ sur $\mathcal{X}$, et une classe d'hypothèses $\mathcal{H}$, la $\mathcal{H}$-*divergence* entre $\mathcal{D}_S^X$ et $\mathcal{D}_T^X$ est donnée par:

$$
\begin{aligned}
d_{\mathcal{H}}(\mathcal{D}_S^X, \mathcal{D}_T^X) &\stackrel{\text{def}}{=} 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_{\mathbf{x}^s \sim \mathcal{D}_S^X} \left[\eta(\mathbf{x}^s) = 1\right] - \Pr_{\mathbf{x}^t \sim \mathcal{D}_T^X} \left[\eta(\mathbf{x}^t) = 1\right] \right|. \\
&= 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_{\mathbf{x}^s \sim \mathcal{D}_S^X} \left[\eta(\mathbf{x}^s) = 1\right] + \Pr_{\mathbf{x}^t \sim \mathcal{D}_T^X} \left[\eta(\mathbf{x}^t) = 0\right] - 1 \right|.
\end{aligned}
$$

La $\mathcal{H}$-**divergence** mesure la capacité d'une classe d'hypothèse $\mathcal{H}$ à **discriminer** entre les distributions source $\mathcal{D}_S^X$ et cible $\mathcal{D}_T^X$.



Borne sur le risque cible

Theorem (Ben David et al., 2006)

Soit $\mathcal{H}$ une classe d'hypothèse de VC-dim égale à d. Soit les échantillons $S \sim (\mathcal{D}_S)^m$ et $T \sim (\mathcal{D}_T^X)^m$. Avec probabilité $1-\delta$, pour tout $\eta \in \mathcal{H}$ on a:

$$R_{\mathcal{D}_T}(\eta) \leq R_S(\eta) + \frac{4}{m}\sqrt{d \log \frac{2em}{d} + \log \frac{4}{\delta}} + \hat{d}_{\mathcal{H}}(S,T) + \frac{4}{m^2}\sqrt{d \log \frac{2m}{d} + \log \frac{4}{\delta}} + \beta$$

avec $\beta \geq \inf_{\eta^* \in \mathcal{H}} \left[R_{\mathcal{D}_S}(\eta^*) + R_{\mathcal{D}_T}(\eta^*)\right]$, *et e est la base du logarithme népérien.*

Avec, $R_S(\eta)$ est le risque empirique sur l'ensemble source:

$$R_S(\eta) \overset{\text{def}}{=} \frac{1}{m}\sum_{i=1}^{m} I\left[\eta(\mathbf{x}_i^s) \neq y_i^s\right].$$

Et la $\mathcal{H}$-divergence empirique:

$$\hat{d}_{\mathcal{H}}(S,T) \overset{\text{def}}{=} 2 \max_{\eta \in \mathcal{H}} \left[\frac{1}{m}\sum_{i=1}^{m} I[\eta(\mathbf{x}_i^s) = 1] + \frac{1}{m}\sum_{i=1}^{m} I[\eta(\mathbf{x}_i^t) = 0] - 1\right].$$

Borne sur le risque cible

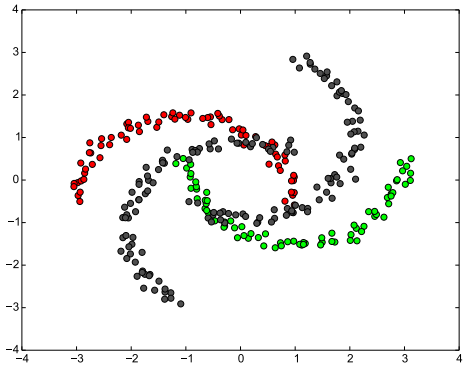
Theorem (Ben David et al., 2006)

Soit $\mathcal{H}$ une classe d'hypothèse de VC-dim égale à d. Soit les échantillons $S \sim (\mathcal{D}_S)^m$ et $T \sim (\mathcal{D}_T^X)^m$. Avec probabilité $1-\delta$, pour tout $\eta \in \mathcal{H}$ on a:

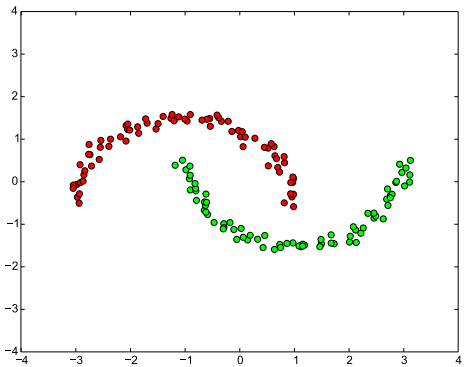
$$R_{\mathcal{D}_T}(\eta) \leq R_S(\eta) + \frac{4}{m}\sqrt{d \log \frac{2em}{d} + \log \frac{4}{\delta}} + \hat{d}_{\mathcal{H}}(S,T) + \frac{4}{m^2}\sqrt{d \log \frac{2m}{d} + \log \frac{4}{\delta}} + \beta$$

avec $\beta \geq \inf_{\eta^* \in \mathcal{H}} [R_{\mathcal{D}_S}(\eta^*) + R_{\mathcal{D}_T}(\eta^*)]$, *et e est la base du logarithme népérien.*

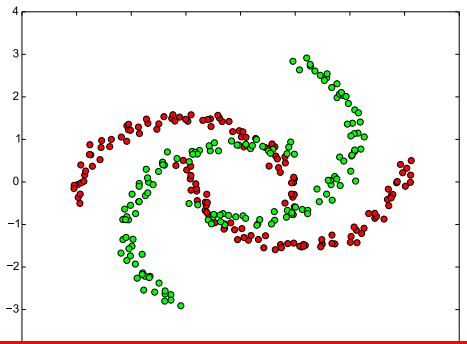
$R_{\mathcal{D}_T}(\eta)$ **est petit**
si, étant donné S et T,



$R_S(\eta)$ **est petit**,
i.e., $\eta \in \mathcal{H}$ est bon pour



et $\hat{d}_{\mathcal{H}}(S,T)$ **est petit**,
i.e., tout $\eta' \in \mathcal{H}$ sont
mauvais sur la tâche



Quelques méthodes de l'adaptation de domaine

- Repondération: [HGB+06, MMR09, BHS12]
- Auto-étiquetage: [BM10a]
- Recherche d'un espace de représentation commun: [CWB11, BMP06, BM10b, GHLM13, BHLS13]

Plan

Approche par réseau de neurones: Motivation

Depuis 2006, les réseaux de neurones ont fait un retour en force!

- Initialisation judicieuse des paramètres,
- Meilleure régularisation (Dropout, Maxout, etc.),
- Innovation au niveau des fonctions d'activation,
- Meilleurs algorithmes d'optimisation stochastique.
- Aprentissage de la representation de données.

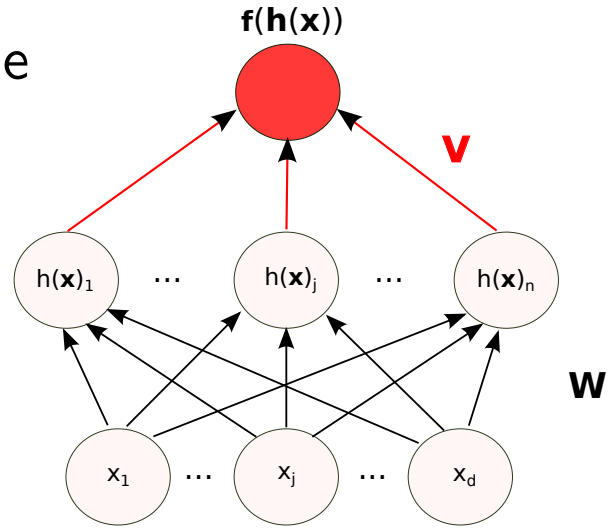
Considérons une architecture de réseau de neurones à une seule couche cachée:

- La représentation cachée, $\mathbf{h}(\mathbf{x}) = \mathrm{sigm}(\mathbf{b} + \mathbf{W}\mathbf{x})$
- La transformation, $\mathbf{f}(\mathbf{h}(\mathbf{x})) = \mathrm{softmax}(\mathbf{c} + \mathbf{V}\mathbf{h}(\mathbf{x}))$

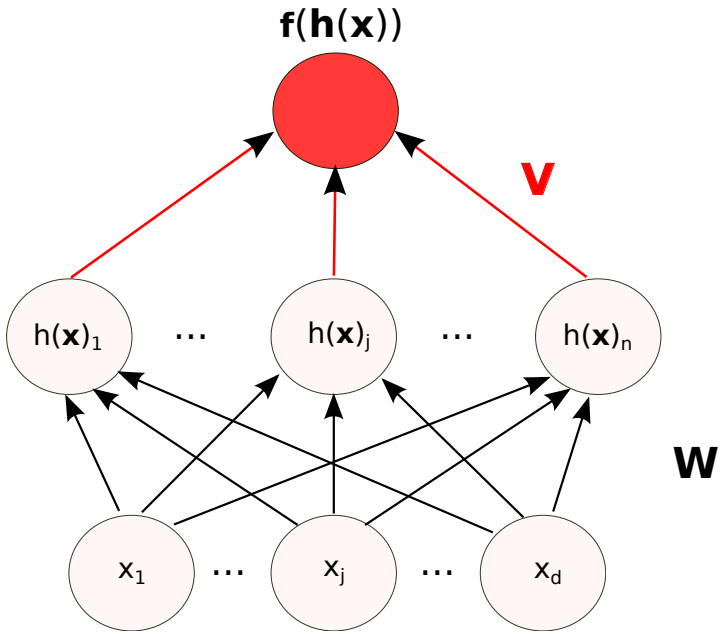
Étant donné un ensemble source $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^m \sim (\mathcal{D}_S)^m$,

L'objectif est de minimiser l'erreur empirique régularisée:

$$\min_{\mathbf{W},\mathbf{V},\mathbf{b},\mathbf{c}} \underbrace{\left[\frac{1}{m}\sum_{i=1}^{m} -\log(f_{y_i^s}(\mathbf{x}_i^s)) + \lambda\Omega(\mathbf{W},\mathbf{V})\right]}_{\text{Risque source + Un eventuel régularisateur}} .$$



Avec, $f_{y_i^s}(\mathbf{x}_i^s) = softmax\left(\mathbf{v}_{y_i^s}^\top \mathbf{h}(\mathbf{x}_i^s) + c_{y_i^s}\right)$,
et $c_{y_i^s}$ et $\mathbf{v}_{y_i^s}$ sont les i^{eme} composantes de $\mathbf{c}$ et V respectivement.



1. Prendre un exemple $(\mathbf{x}^s, y^s) \in S$
2. Mettre à jour $\mathbf{V}$ pour avoir $f_{y^s}(\mathbf{x}^s)$ proche de 1.
3. Mettre à jour $\mathbf{W}$ pour avoir $f_{y^s}(\mathbf{x}^s)$ proche de 1.

La couche cachée apprend une **représentation** interne $\mathbf{h}(\cdot)$ à partir de laquelle l'hypothèse linéaire $\mathbf{f}(\cdot)$ peut **classifier les exemples sources correctement.**

Notre approche: Domain-Adversarial Neural Network (DANN)

Domain-Adversarial Neural Network (DANN)

La $\mathcal{H}$-divergence empirique

$$\hat{d}_{\mathcal{H}}(S, T) \stackrel{\text{def}}{=} 2 \max_{\eta \in \mathcal{H}} \left[\frac{1}{m} \sum_{i=1}^{m} I[\eta(\mathbf{x}_i^s) = 1] + \frac{1}{m} \sum_{i=1}^{m} I[\eta(\mathbf{x}_i^t) = 0] - 1 \right].$$

On estime la $\mathcal{H}$-divergence par une régression logistique qui modélise la probabilité qu'une entrée donnée (soit $\mathbf{x}^s$ ou $\mathbf{x}^t$) est issue du domaine source:

$$o(\mathbf{h}(\mathbf{x})) \stackrel{\text{def}}{=} \text{sigm}(d + \mathbf{w}^\top \mathbf{h}(\mathbf{x})).$$

Étant donnée une représentation $\mathbf{h}(\cdot)$:

$$\hat{d}_{\mathcal{H}}\Big(\mathbf{h}(S), \mathbf{h}(T)\Big) \approx 2 \max_{\mathbf{w}, d} \left[\frac{1}{m} \sum_{i=1}^{m} \log\big(o(\mathbf{h}(\mathbf{x}_i^s))\big) + \frac{1}{m} \sum_{i=1}^{m} \log\big(1 - o(\mathbf{h}(\mathbf{x}_i^t))\big) - 1 \right].$$

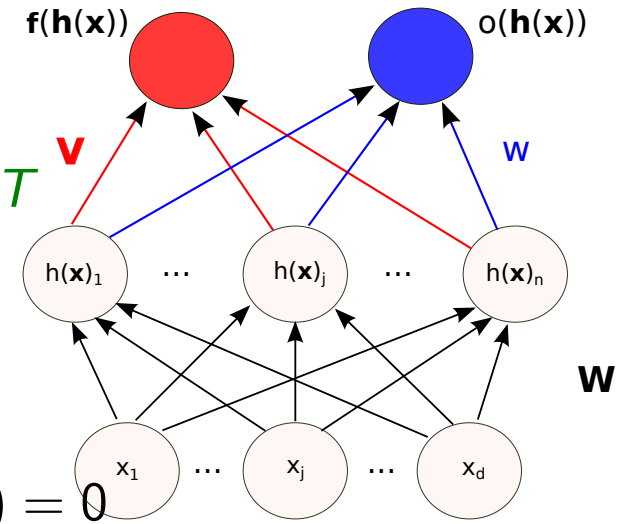
Domain-Adversarial Neural Network (DANN)

$$\min_{\mathbf{W},\mathbf{V},\mathbf{b},\mathbf{c}} \Bigg[\underbrace{\frac{1}{m}\sum_{i=1}^{m} -\log\left(f_{y_i^s}(\mathbf{x}_i^s)\right)}_{\text{Risque source}} + \lambda \underbrace{\max_{\mathbf{w},d} \left(\frac{1}{m}\sum_{i=1}^{m} \log\left(o(\mathbf{h}(\mathbf{x}_i^s))\right) + \frac{1}{m}\sum_{i=1}^{m} \log\left(1-o(\mathbf{h}(\mathbf{x}_i^t))\right) \right)}_{\text{Régularisateur d'adaptation}} \Bigg],$$

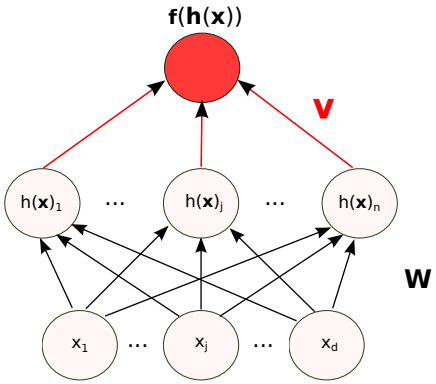
où $\lambda > 0$ pondère le terme de régularisation de l'adaptation de domaine.

Étant donné un ensemble source $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^m \sim (\mathcal{D}_S)^m$,
et un ensemble cible $T = \{(\mathbf{x}_i^t)\}_{i=1}^m \sim (\mathcal{D}_T)^m$,

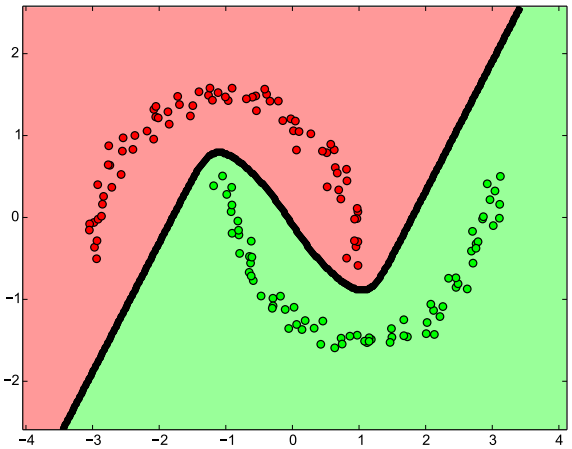
1. prendre un exemple $(\mathbf{x}^s, y^s) \in S$ et un exemple $\mathbf{x}^t \in T$
2. Mettre à jour $\mathbf{V}$ pour avoir $f_{y^s}(\mathbf{x}^s)$ proche de 1
3. Mettre à jour $\mathbf{W}$ pour avoir $f_{y^s}(\mathbf{x}^s)$ proche de 1
4. Mettre à jour $\mathbf{w}$ pour avoir $o(\mathbf{h}(\mathbf{x}^s)) = 1$ et $o(\mathbf{h}(\mathbf{x}^t)) = 0$
5. Mettre à jour $\mathbf{W}$ pour avoir $o(\mathbf{h}(\mathbf{x}^s)) = 0$ et $o(\mathbf{h}(\mathbf{x}^t)) = 1$



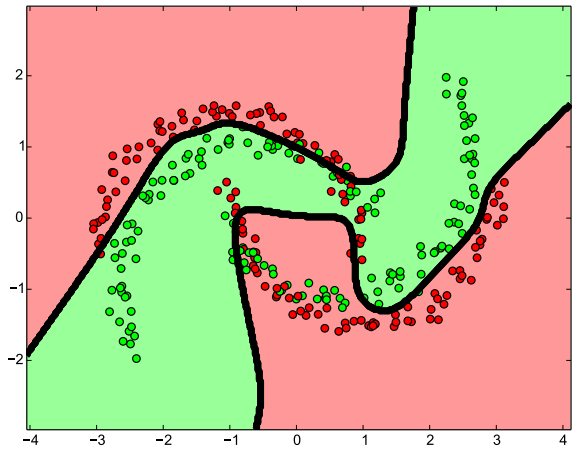
Problème jouet

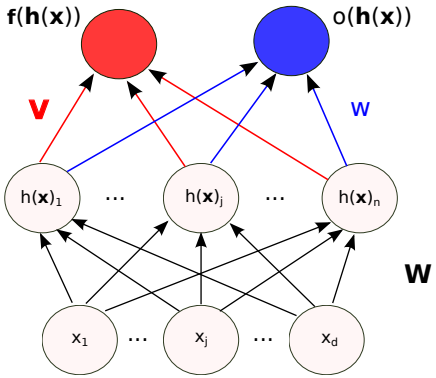


Entrainé pour classifier l'ensemble source.

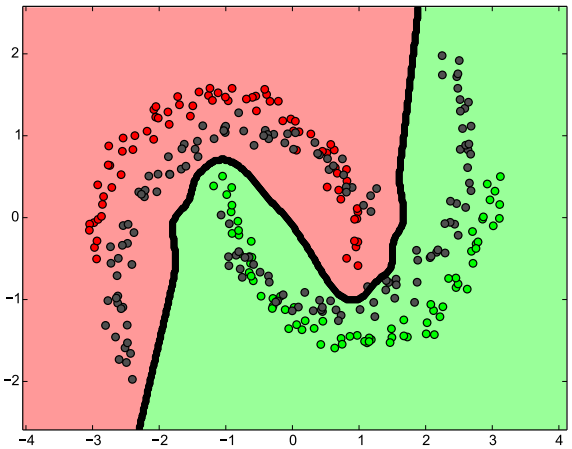


Entrainé pour classifier les domaines.

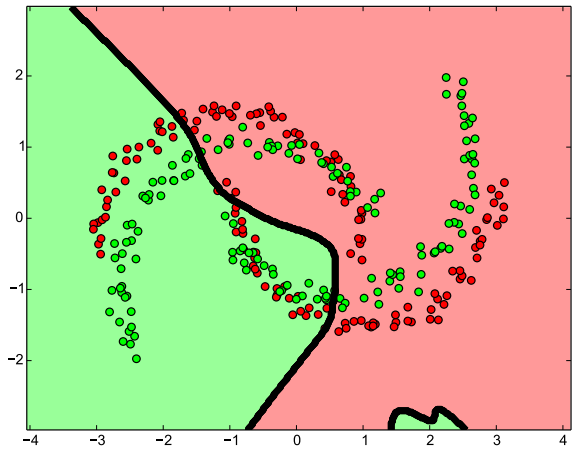




Sortie pour la classification: $\mathbf{f(h(x))}$



Sortie pour la distinction entre les domaines: $o(\mathbf{h(x)})$



Ensemble de données Amazon

Input: Critiques sur produits d'amazon (bag of words) — **Output:** Une côte positive ou négative.

Dataset	**DANN**	**NN**
books → dvd	0.201	**0.199**
books → electronics	**0.246**	0.251
books → kitchen	**0.230**	0.235
dvd → books	**0.247**	0.261
dvd → electronics	**0.247**	0.256
dvd → kitchen	0.227	0.227
electronics → books	**0.280**	0.281
electronics → dvd	**0.273**	0.277
electronics → kitchen	**0.148**	0.149
kitchen → books	**0.283**	0.288
kitchen → dvd	0.261	0.261
kitchen → electronics	0.161	0.161

Marginalized Stacked Denoising Autoencoders (mSDA)

Question

Est ce que DANN peut être combiné avec une autre méthode d'apprentissage de représentation pour l'adaptation de domaines?

Les "autoencoders" mSDA (Chen et al. 2012) fournissent une nouvelle représentation communes pour les domaines source et cible

Avec **mSDA+SVM**, Chen et al. (2012) est devenu *l'état de l'art* sur les données d'Amazon:
– Entrainé un SVM linéaire sur les représentations mSDA source+cible.

Nous avons essayé **mSDA+DANN**:
– Entrainé notre algorithme DANN sur les représentations mSDA source+cible.

Ensemble de données Amazon

Input: les critiques sur les produits (bag of words) — **Output:** Une côte positive ou négative.

Dataset	**mSDA+DANN**	**mSDA+SVM**
books → dvd	0.176	**0.175**
books → electronics	**0.197**	0.244
books → kitchen	**0.169**	0.172
dvd → books	0.176	0.176
dvd → electronics	**0.181**	0.220
dvd → kitchen	**0.151**	0.178
electronics → books	0.237	**0.229**
electronics → dvd	**0.216**	0.261
electronics → kitchen	**0.118**	0.137
kitchen → books	**0.222**	0.234
kitchen → dvd	**0.208**	0.209
kitchen → electronics	0.141	**0.138**

Conclusion

Nous avons présenté:

- Le différence entre l'apprentissage supervisé standard et l'adaptation de domaine.
- Notre approche DANN pour l'adaptation de domaine qui trouve une représentation de données $\mathbf{h}(\cdot)$ qui est:
  - Performante sur la tâche de classification de l'ensemble source S;
  - **Mais** qui est incapable de discriminer entre les exemples provenants de S et ceux provenants de T.
  - Devenu l'état de l'art sur les données d'Amazon utilisées dans la littérature.

Remerciement

- Domain-adversarial neural networks [AGL+14]
- Domain-Adversarial Training of Neural Networks [GUA+15]

Merci!

Plan

Bibliographie I


Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand, *Domain-adversarial neural networks*, arXiv preprint arXiv:1412.4446 (2014).


Mahsa Baktashmotlagh, Mehrtash Harandi, Brian Lovell, and Mathieu Salzmann, *Unsupervised domain adaptation by domain invariant projection*, Proceedings of the 2013 IEEE International Conference on Computer Vision, 2013, pp. 769–776.


Aurélien Bellet, Amaury Habrard, and Marc Sebban, *Apprentissage de bonnes similarités pour la classification linéaire parcimonieuse*, Conférence Francophone sur l'Apprentissage Automatique-CAp 2012, 2012, pp. 16–p.

Bibliographie II


Lorenzo Bruzzone and Mattia Marconcini, *Domain adaptation problems: A dasvm classification technique and a circular validation strategy*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010), no. 5, 770–787.


Lorenzo Bruzzone and Mattia Marconcini, *Domain adaptation problems: A DASVM classification technique and a circular validation strategy*, IEEE Transaction Pattern Analysis and Machine Intelligence **32** (2010), no. 5, 770–787.


John Blitzer, Ryan T. McDonald, and Fernando Pereira, *Domain adaptation with structural correspondence learning*, Conference on Empirical Methods in Natural Language Processing, 2006, pp. 120–128.

Bibliographie III


Minmin Chen, Kilian Q Weinberger, and John Blitzer, *Co-training for domain adaptation*, Advances in neural information processing systems, 2011, pp. 2456–2464.


Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant, *A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers*, ICML, 2013, pp. 738–746.


Yaroslav Gani, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, *Domain-adversarial training of neural networks*, arXiv preprint arXiv:1505.07818 (2015).


Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola, *Correcting sample selection bias by unlabeled data*, Advances in neural information processing systems, 2006, pp. 601–608.

Bibliographie IV


Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh, *Multiple source adaptation and the rényi divergence*, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2009, pp. 367–374.

Sélection de modèle

Algorithm 1 Validation inverse

Étant donné un ensemble source étiqueté $S = (\mathbf{x}_i^S, y_i^S)_{i=1}^m$ et un ensemble cible non-étiqueté $T = (\mathbf{x}_i^T)_{i=1}^m$.

1. On sépare les ensembles S et T respectivement en ensembles d'entraînement (S_{tr} et T_{tr}) et en ensembles de validation (S_V et T_V respectivement.)
2. Construire le classificateur h à l'aide de l'ensemble d'entraînement source étiqueté S_{tr} et de l'ensemble d'entraînement cible non étiqueté T_{tr}. On a alors $h \overset{\text{def}}{=} A(S_{tr} + T_{tr})$.
3. Auto-étiqueter T_{tr} à l'aide du classificateur h. Soit $h(T_{tr})$ les étiquettes prédites pour l'ensemble cible.
4. Construire un autre classificateur h_r à partir de l'ensemble d'entraînement source non étiqueté $S = (\mathbf{x}_i^S)_{i=1}^m$ et de l'ensemble d'entraînement cible auto-étiqueté $T \overset{\text{def}}{=} \{(\mathbf{x}_i^T, h(\mathbf{x}_i^T))\}_{i=1}^m$.
5. Finalement, évaluer le risque $R_{S_V}(h_r)$ du classificateur h_r sur S_V.