Reading Club Kognitive Systeme

KogSys- Sem- M2

Categorization of Transfer Learning Techniques

Emphasis on Self- Taught Learning

Christian Reißner

Sommersemester 2012

Inhalt

1. Einführung in die Thematik

Im Bereich des maschinellen Lernens gibt es bereits viele verschiedene Ansätze und Algorithmen, die zum gewünschten Ziel führen sollen. Eine noch recht junge Form des maschinellen Lernens ist das so genannte „t*ransfer learning*“, auf das in diesem Dokument besonders Wert gelegt wird. Dabei handelt es sich um eine neue Lernumgebung, bei der im Gegensatz zum traditionellen „*machine learning*“, eine Aufteilung der Quell- und Zieldomains stattfindet. Des Weiteren gibt es noch weitere Unterscheidungsmöglichkeiten und Eigenschaften mit denen der Unterbereich des transfer learnings definiert werden kann.

Um den Überblick bei dieser wachsenden Vielfalt zu bewahren oder um ein einfacheres Einarbeiten in die Semantik zu ermöglichen ist es wichtig die vorhandenen Alternativen zu kategorisieren. Daher wird in diesem Dokument eine Kategorisierung des Themas vorgestellt. Es wird ein Überblick gegeben, angefangen bei den Unterschieden zum traditionellen maschinellen Lernen bis zu den unterschiedlichen Varianten des übertragenen Lernens.

Weiter wird ein besonderes Augenmerk auf den Teilbereich *„self-taught learning„* gelegt. Das ist eine neue Lernumgebung, die unter die Kategorie transfer learning fällt. Diese Form des übertragenen Lernens beschäftigt sich mit der Einordnung von unmarkierten Daten in definierte Kategorieklassen. Zu diesem Thema werden Problemstellungen behandelt und der hinter dem Namen steckende Algorithmus wird betrachtet, außerdem wird ein Vergleich mit anderen Methoden in Augenschein genommen.

2. Ein Überblick

Um einen ersten Überblick zu schaffen und um die einzelnen Verbindungen besser darlegen zu können, wird auf eine optische Variante der Darstellung zurückgegriffen. Nachfolgend wird in einem Diagramm die Struktur und die einzelnen Teilbereiche des *„transfer learnings“* aufgezeigt. Auf die einzelnen Teilgebiete wird in den darauf folgenden Kapiteln eingegangen, besonders auf den Bereich des autodidaktischen Lernens.

Die Teilbereiche des transfer learnings auf einen Blick:

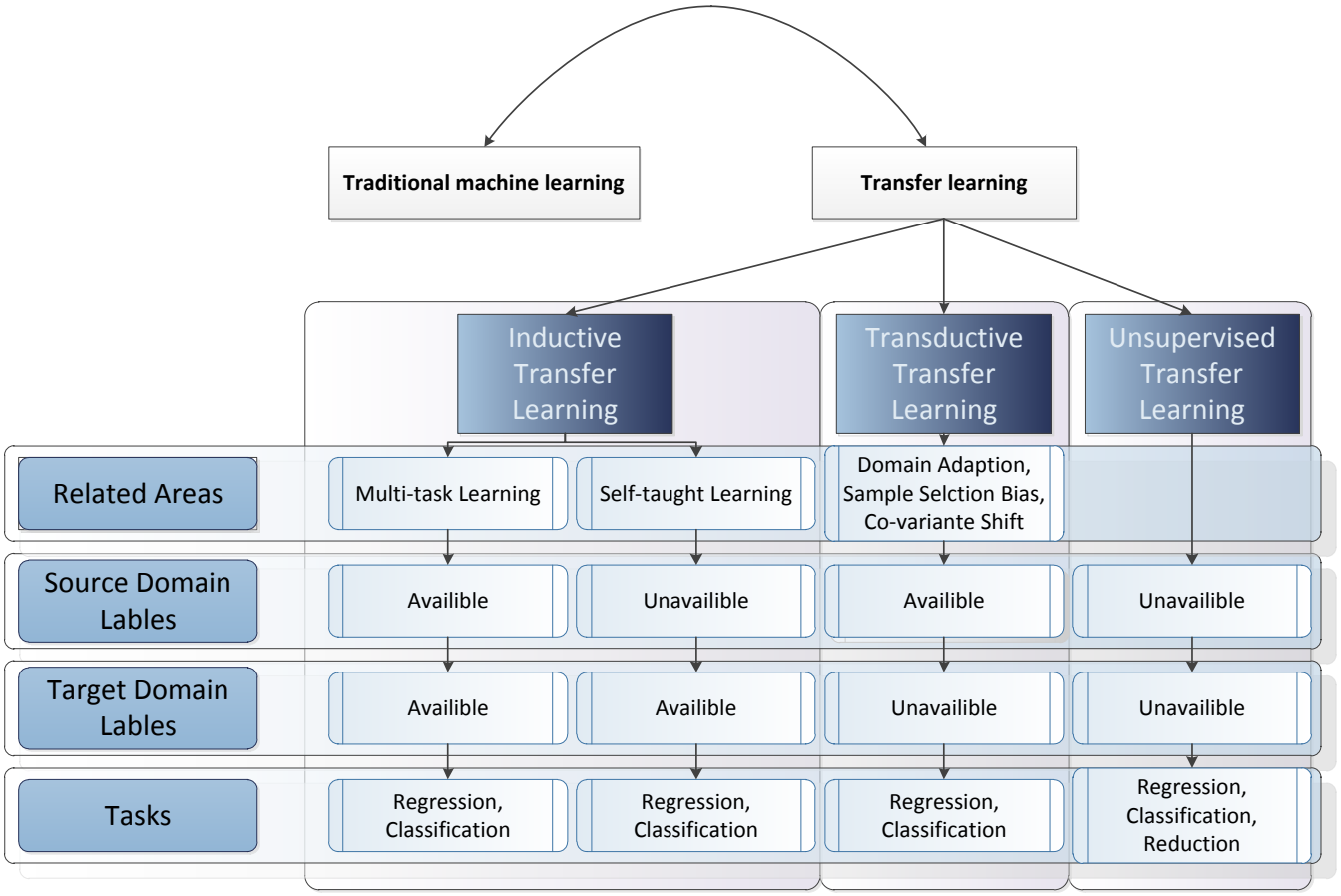


Abbildung 3.1- 1: Überblick der Transfer Learning Bereiche [1]

Auf den ersten Blick ist erkennbar, dass sich *„transfer learning“* in drei Sparten unterteilen lässt.

- Inductive Transfer Learning
- Transductive Transfer Learning
- Unsupervised Transfer Learning

Die Aufteilung in diese drei Gruppen erfolgt auf Grund der unterschiedlichen Nutzung von Quell- und Zieldomains, sowie der Quell- und Zieltasks[1].

[1] Näheres zu dem Thema Domains und Tasks wird in Kapitel 3. Traditional Machine Learning vs. Transfer Learning geboten

Die nächste Unterteilung bilden die Einstellungsmöglichkeiten des transfer learnings. Diese gelten für jede Untergruppierung, die zuvor stattgefunden hat. Hierbei handelt es sich um:

- *Related Areas*
- *Source Domain Lables*
- *Target Domain Lables*
- *Tasks*

Realated Areas sind ähnliche Bereiche, die unter denselben Überbegriff zusammen genannt werden können. Die *Source Domain Lables* geben an, ob für die Methoden eine Quelldomain bekannt ist, also ob bereits Daten bekannt sein müssen, um die Methode anwenden zu können. Entsprechend den zuvor genannten Source Domain Lables handelt es sich bei den Target Domain Labels darum, ob eine Zieldomain mit Daten vorhanden sein muss, in welche die transferierten Daten klassifiziert werden können. Hinter dem Begriff „*Tasks*“ steht das Vorgehen der einzelnen Methoden; so ist das induktive Verfahren z.B. regressiv und klassifizierend.

Auf die einzelnen Bereiche und deren Bedeutung wird im Verlauf der Arbeit noch einmal kurz eingegangen.

3. Traditional Machine Learning vs. Transfer Learning

Vorab gibt es Definitionen und Notationen, die für den weiteren Verlauf von Bedeutung sind.

Definition einer Domain:

$D = \{\mathcal{X}, P(X)\}$ $\quad D: domain \quad \mathcal{X}: Merkmale\ Raum$

$P(X): marginale\ Wahrscheinlichkeitsverteilung\ \ und\ X = \{x_1, \ldots, x_n\} \epsilon\ \mathcal{X}$

Eine Domain ist ein Bereich oder ein Ort in dem Daten/ Wissen liegt.

Definition einer Task:

$\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$ $\quad \mathcal{T}: Task\ \ \mathcal{Y}: Kennzeichen - Raum\ (enthält\ alle\ labels)$

$P(Y|X): Funktion\ zum\ ermitteln\ eines\ bestimmten\ labels$

Eine Task ist eine bearbeitende Einheit, welche von den Daten einer Domain lernt.

3.1. Traditional Machine Learning

Bei der traditionellen Übertragung von Wissen läuft das Data Mining und die Algorithmen des maschinellen Lernens mit Vorhersagen über künftige Daten mit Hilfe statischer Modelle. Diese müssen erst mit zuvor gesammelten markierten oder unmarkierten Daten geschult werden.

Sind die Klassifikationen nur teilweise geschult wird das Problem aufgeworfen, dass es zu wenige markierte Daten gibt um einen guten Klassifikator zu bilden. Das heißt, dass im Verhältnis zu viele unmarkierte gegenüber den markierten Daten vorhanden sind.

3.2. Transfer Learning

Im Jahr 2005 hat die *"Broad Agency Announcement of Defense Advanced Research Projects Agency`s Information Processing Technoloy Office"* einen neuen Auftrag zum Thema *transfer learning* ausgestellt. Es sollte ein System geschaffen werden, das Kenntnisse und Fertigkeiten lernen kann und diese auf bekannte und neue Aufgaben anwenden kann. Sie definieren das Verfahren so, dass es zum Ziel hat, Kenntnisse aus einer oder mehreren Quellbereichen zu extrahieren und das so entstandene Wissen auf einen Zielbereich anzuwenden.

Eine formelle Definition für *transfer learning* lautet original wie folgt:

> *Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T , transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.*

Weiter ist eine Domain D als Paar $D = \{\mathcal{X}, P(X)\}$ definiert und durch die Aussage $D_S \neq D_T$ wird impliziert, dass $\mathcal{X}_S \neq \mathcal{X}_T$.

Eine Task $\mathcal{T}$ ist gleichzeitig definiert als Paar $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$ und durch die Aussage $\mathcal{T}_S \neq \mathcal{T}_T$ wird impliziert, dass $\mathcal{Y}_S \neq \mathcal{Y}_T$.

Sobald Ziel- und Quelldomains, sowie Ziel- und Quelltasks jeweils dieselben sind, handelt es sich wieder um ein Problem des traditionellen maschinellen Lernens.

3.3. Der Unterschied

Es wird die Frage geklärt, wie der Unterschied zwischen dem traditionellen maschinellen Lernen und dem autodidaktischen Lernen definiert ist.

Die grundsätzliche Unterscheidung wird auf Grund der Quell- und Zieldomains, sowie der Quell- und Zieltasks getroffen. Es wird dabei unterschieden, ob diese Bereiche nur zusammenhängend sind oder ob es sich bei zwei Bereichen um denselben handelt. Beim traditionellen maschinellen Lernen sind sowohl die Quell- und Zieldomains, als auch die Quell- und Zieltasks je ein und derselbe Bereich.

Entsprechend anders ist es beim transferierten Lernen, bei dem die Domains oder die Tasks lediglich zusammenhängend, nicht aber dieselben sind.

Folgende Tabelle gibt einen Überblick über die unterschiedlichen Nutzräume:

Lern Methode		Quell- und Zieldomains	Quell- und Zieltasks
Traditionelles maschinelles Lernen		Gleich	Gleich
Transferiertes Lernen	Induktives transferiertes Lernen	Gleich	Unterschiedlich aber zusammenhängend
	Unbeaufsichtigtes transferiertes Lernen	Gleich	Unterschiedlich aber zusammenhängend
	Transduktives transferiertes Lernen	Unterschiedlich aber zusammenhängend	Gleich

Tabelle 3.3-1: Zusammenhang zwischen traditionellen maschinellen Lernen und verschiedenen Methoden des autodidaktischen Lernens [1]

Weiter unterscheiden sich die beiden Verfahren in Hinblick auf das Lernverhalten von Erkenntnissen (Tasks). Die traditionelle Variante versucht zu jeder Erkenntnis von Grund auf zu Lernen, um so verschiedene Lernziele zu erreichen.

Bei dem autodidaktischen Lernen hingegen wird versucht, aus verschiedenen früheren Erkenntnissen gemeinsam zu lernen und das so erlernte Wissen auf einen gemeinsamen Zielrahmen abbilden zu können.

Folgende Darstelleng erläutert den eben beschriebenen Unterschied:

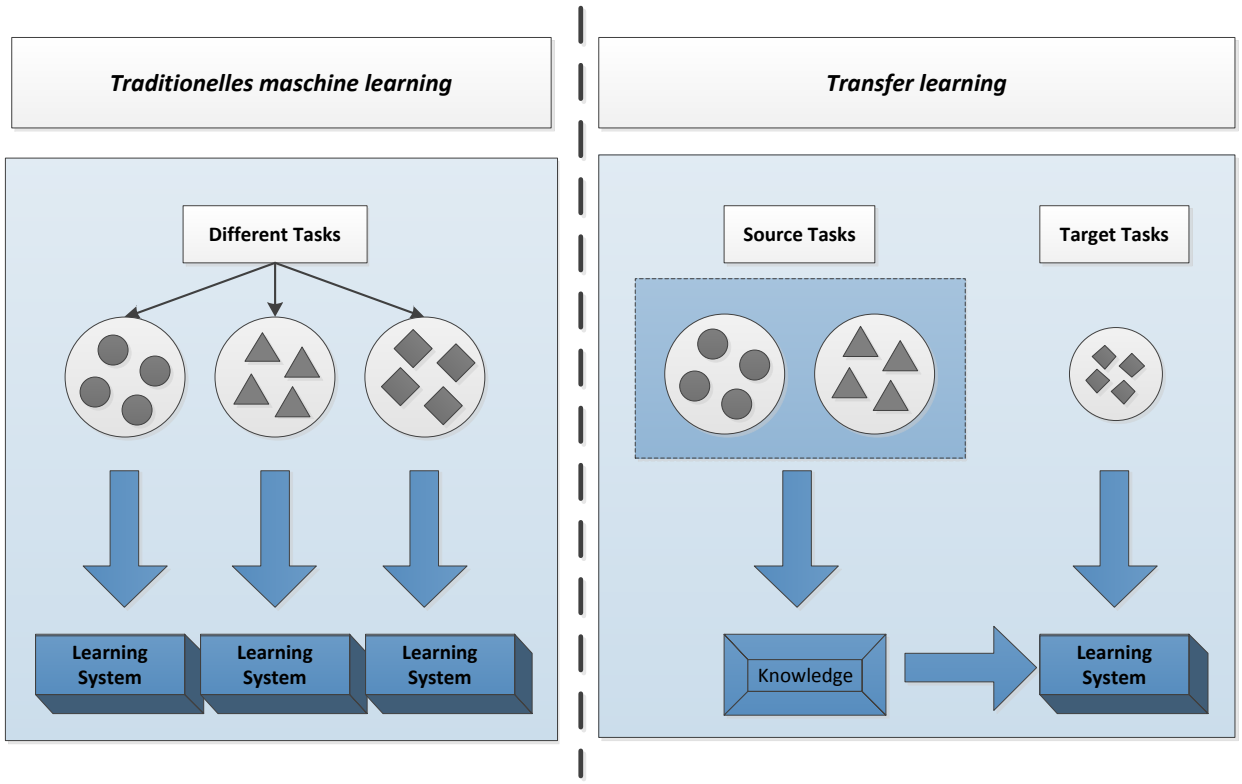


Abbildung 3.3-2: Unterschied traditionelles maschinelles Lernen und transferiertes Lernen [1]

Man kann sagen, dass die Abwicklung der Wissensübertragung zwischen Quell- und Zieldomain mit dem Wechsel vom traditionellen maschinellen Lernen zum autodidaktischen Lernen nicht mehr symmetrisch abläuft.

4. Transfer Learning categories

Für die Kategorisierung des transfer learnings werden drei Fragestellungen betrachtet:

- **Was ist zu übertragen**
- **Wie ist zu übertragen**
- **Wann ist zu übertragen**

„*Was ist zu übertragen*“ meint, welchen Teil eines Wissens kann zwischen verschiedenen Bereichen übertragen werden.

„*Wie ist zu übertragen*“, betrifft die Realisierung dieser Übertragung.

Die Frage „*Wann ist zu übertragen*“, fragt nach der am besten geeigneten Situation um Wissen zu übertragen.

4.1.1. Inductive Transfer Learning

Definition:

Given a source domain D_S and a learning task T_S, a target domain $D_T$and a learning task T_T , inductive transfer learning aims to help improve the learning of the target predictive function $f_T\ (\cdot)$ in D_T using the knowledge in D_S and T_S, where $T_S \neq T_T$.

Bei der induktiven Übertragung von Wissen sind die Quell- und Zielbereich des bekannten Wissens verschieden. Es macht also nichts aus, wenn Quellbereich und Zielbereich verschieden sind.

Des Weiteren sind in diesem Fall bereits bekannte Daten im Zielbereich verfügbar. Ob Daten auch im Quellbereich bekannt sind führt hingegen zu einer weiteren Unterscheidung dieser Kategorie, in „*multi- task learning*“ und „*self- taught learning*“.

Beim *multi- task learning* sind bereits eine große Menge von markierten Daten im Quellbereich verfügbar. Also sind beim bei dieser Methode in beiden Domains Daten bekannt, was wiederum notwendig ist, um den Anspruch an das Verfahren zu erfüllen. Dieser ist nämlich, beim Übertragen von Wissen möglichst performant zu sein. Dies wird dadurch erreicht, dass diese Variante aus Quell- und Zielbereichen gleichzeitig lernt und Erfahrungen sammelt.

Das Thema self- taught learning wird im späteren Verlauf der Arbeit (*Kapitel 6. Self- Taugth Learning*) genauer analysiert und wird daher an dieser Stelle nicht weiter betrachtet.

4.1.2. Transductive Transfer Learning

Definition:

> *Given a source domain D_S and a corresponding learning task T_S, a target domain D_T and a corresponding learning task T_T, transductive transfer learning aims to improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge inD_S and T_S, where $D_S = D_T$ and $T_S = T_T$. In addition, some unlabeled target-domain data must be available at training time.*

Bei der transduktiven Wissensübertragung sind die Quell- und Zieltasks gleich, da Quell- und Zieldomains unterschiedlich sind. Es sind keine gelabelten Daten im Zielbereich verfügbar, stattdessen sind viele gelabelten Daten im Quellbereich verfügbar.

Es gibt noch zwei weitere Unterscheidungsmöglichkeiten beim *„transductive transfer learning“*. Diese sind *„Domain Adaption“* und *„Sample Selction Bias/ Covariance Shift“*. Bei der Domain Adaption ist der Feature-Raum zwischen der Quell- und Zieldomain unterschiedlich $\mathcal{X}_S \neq \mathcal{X}_T$. Die zweite Unterscheidung ist ein gemeinsamer Feature-Raum zwischen den Domains $\mathcal{X}_S = \mathcal{X}_T$, aber mit Eigenschaften der Eingangsparameter mit $P(\mathcal{X}_S) \neq P(\mathcal{X}_T)$.

4.1.3. Unsupervised Transfer Learning

Definition:

> *Given a source domain D_S with a learning task T_S, a target domain D_T and a corresponding learning task T_T , unsupervised transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S , where $T_S \neq T_T$ and $\mathcal{Y}_S$and $\mathcal{Y}_T$ are not observable.*

Beim letzten Teilbereich, sind sowohl im Quell-, als auch im Zielbereich keine gelabelten Daten vorhanden. Der Fokus sitzt hier darauf, nicht überwachtes Lernen im Ziel-Task zu lösen.

Die Eigenschaften des *„unsupervised transfer learnings“* sind ähnlich denen des induktiven Lernens, denn der Ziel-Task ist verschieden zum Quell-Task, aber trotzdem zusammenhängend.

5. Verschiedene Ansätze

Im Bereich „*transfer learning*“ gibt es neben den oben vorgestellten Einstellungen noch verschiedene Ansätze, welchen in den einzelnen Einstellungen nachgegangen wird.

Folgende Ansätze werden Unerschieden:

- Instance transfer
- Feature- representation- transfer
- Parameter- transfer
- Relational- knowledge- transfer

Diese vier Ansätze kommen, verschieden vertreten, in den drei Sparten des *transfer learnings* vor. Jeder dieser Ansätze verfolgt ein bestimmtes Ziel und hat entsprechende Anforderungen an die Gegebenheiten.

Nachfolgende Tabelle gibt einen Überblick, welche Ansätze in welchem Bereich vorkommen:

	Inductive Transfer Learning	**Transductive Transfer Learning**	**Unsuperviced Transfer Learning**
Instance transfer	✓	✓	
Feaute- representation-transfer	✓	✓	✓
Parameter- transfer	✓		
Relational- knowledge-transfer	✓		

Tabelle 4.1.3-1: Ansätze in den verschiedenen Einstellungen [1]

5.1. Instance- transfer

Obwohl die Daten der Quelldomain nicht direkt wieder verwendet werden können, gibt es Teile der Daten, die zusammen mit den Daten der Zieldomain verwendet werden können. Durch einen iterativen Ansatz wird versucht die Quelldaten zu gewichten und damit die „schlechten“ Quelldaten zu reduzieren und die „guten“ Quelldaten noch förderlicher für die Zieldaten zu machen.

Zu diesem Ansatz gibt es auch schon verschiedene Ansätze wie zum Beispiel den *„TrAdaBoost“* Algorithmus oder andere Methoden, die einem heuristischen Ansatz nachgehen .Der *„TrAdaBoost„* Algorithmus ist ein *„boosting“* Algorithmus[2] und behandelt dieses Problem. Er aktualisiert falsch klassifizierte Quelldaten.

[2] Boosting algorithm = automatische Klassifizierung, der mehrere schlechte Klassifikatoren zu einem guten verschmilzt

5.2.Feature- representation- transfer

Dieser Ansatz sucht, ganz wie der Name sagt, eine gute Featuredarstellung, welche die Domaindivergenz, sowie den Klassifikationsfehler reduziert. Welche Strategie hierzu schlussendlich eingesetzt wird hängt vom Verhalten der Quelldomain ab. Gibt es dort viele bereits markierte Daten können *„supervised learning methods“* eingesetzt werden um eine Featuredarstellung zu erstellen, ähnlich ist dazu das *„common feature learning“*. Sind keine markierten Daten im Quellbereich vorhanden, sind *„unsupervised learning methods“* die bessere Variante.

5.3.Parameter- transfer

Hier geht es darum, gemeinsame Parameter der Quell- und Zieldomain Modelle zu entdecken, welche für das *transfer learning* genutzt werden können. Ein effizienter Algorithmus für die Umsetzung ist der *„MT-IVM“* Algorithmus [6]. Dieser Algorithmus basiert auf dem Gaußprozess[3] und versucht mehrere Taskparameter zu finden.

Weitere Ansätze beschäftigen sich ebenfalls mit dem Gaußprozess um Korrelationen zu induzieren oder nutzen dazu eine hierarchische Bayes-Grundlage.

5.4.Relational- knowledge- transfer

Anders als die vorherigen drei Ansätze beschäftigt sich dieser Ansatz mit dem Problem bei von verbundenen Domains im Bereich *transfer learning*. Es wird die Verbindung von Daten zwischen Quell- und Zieldomain übertragen. Diese Problembehandlung entgegned man mit *„statistical relational learning techniques“*, einen Algorithmus zu diesem Fall stellt der *„TAMAR“* Algorithmus dar, der mit Hilfe von *„Markov Logic Networks“* (MLNs) solche Verbindungen überträgt.

[3] Gaußprozess = verallgemeinerte mehrdimensionale Gaußverteilung über unendlich viele Zufallszahlen.

6. Self- taught learning

6.1.Erläuterung

Hinter dem Begriff „*self- taught learning*“ versteckt sich ein neues „*machine learning*“- Framework, das unmarkierte Daten in bekannte Klassifikations- Tasks ordnet. Selbst aus sehr großen Datenmengen soll es möglich sein Daten zu erkennen und zu sortieren. Der dazu eingesetzte Algorithmus muss dieses Erkennen durchführen können, ohne auf bereits bekannte Daten zugreifen zu können.

Dieser Ansatz hat das Potential das „Lernen“ deutlich zu verbessern und auch günstiger zu gestalten. Zu diesem Zweck werden im folgendem als Beispiel Bilder von Elefanten und Nashörnern untersucht, da es schwierig ist, zu diesem Beispiel markierte Bilder zu finden.

Anhand des in gerade erwähnten Beispiels lässt sich noch ein weiteres Beispiel einbringen, das mehr anwendungsorientiert ist. Es geht darum Elefanten und Nashörner zu klassifizieren und darum, welche Art von Wissens- Ressource dazu verwendet wird.

Abbildung 6.1-3: Verschiedene learning Verfahren in der im Vergleich [2]

In der Abbildung erkennt man von oben nach unten gehend, die hierarchische Struktur des *transfer learnings*. Die orangefarben hinterlegten Bereiche, enthalten markierte Daten, entsprechend enthalten die nicht farbig hinterlegten Bereiche keine markierten Daten.

Superviced Classification:

Hier wird das Verfahren, das auf der einen Seite für das Erkennen von Nashörnern dient übertragen, um in der anderen Domain das Erkennen auf Bildern von Elefanten anzuwenden.

Semi-superviced Learning:

Das Verfahren versucht Bilder aus einem Trainingsset, das nur aus gleichen Bildern besteht die auch in der anderen Domain vorkommen, zu erkennen und zuzuordnen.

Transfer learning:

Beim transfer learning wird versucht das Wissen auf markierte Daten mit einem anderen, aber dennoch ähnlichem Kontext zu übertragen.

Self-taugth learning:

Es wird versucht aus beliebigen Fotos eine Verbindung anhand der Merkmale zu erkennen und diese zu übertragen, solche Daten können auch zufällig heruntergeladene Bilder aus dem Internet sein.

Nachfolgende Tabelle gibt hierzu noch einmal einen Überblick über die vorhandenen Trainingsmengen:

P = es liegen nur passende Bilder in der Domain vor
M = es handelt sich um markierte Bilder
B = es liegen beliebige Bilder vor
U = es liegen unmarkierte Bilder vor

Verfahren	**Quelldomain**			
	P	B	M	U
Supervised Classification			★	
Semi-superviced Learning	★			★
Transfer Learning		★	★	
Self-taught Learning		★		★

Tabelle 6.1-1: Überblick der Trainingsmengen bei versch. Verfahren

Die Tabelle zeigt noch einmal auf, wie die Trainingsmengen der oben aufgeführten Verfahren bearbeitet sind.

Beim Supervised Classification handelt es sich um gleiche Daten in der jeweiligen Domain, aber diese sind nur ähnlich zu den Daten, der jeweils anderen Domain. Deshalb wurde auf die Aussage, ob die Bilder passend oder beliebig sind, an dieser Stelle in der Tabelle verzichtet.

6.2.Einordnung in die „transfer learning“- Hierarchie

Da es sich beim autodidaktischen Lernen um eine Methode handelt, bei der bereits Wissen in der Zieldomain bekannt ist, muss die erste Eingliederung im Bereich der induktiven Transfer Learning sein.

Eine weitere, tiefer gehende Einordnung wird nun durch das Wissen innerhalb der Zieldomain vorgenommen. Da beim *self- taught learning* keine markierten Daten in dieser Domain bekannt sein müssen, unterscheidet sich das Verfahren vom *multi- task learning* und bildet einen eignen Zweig.

Folgendes Diagramm veranschaulicht die Einordnung in die „transfer learning“- Hierarchie:

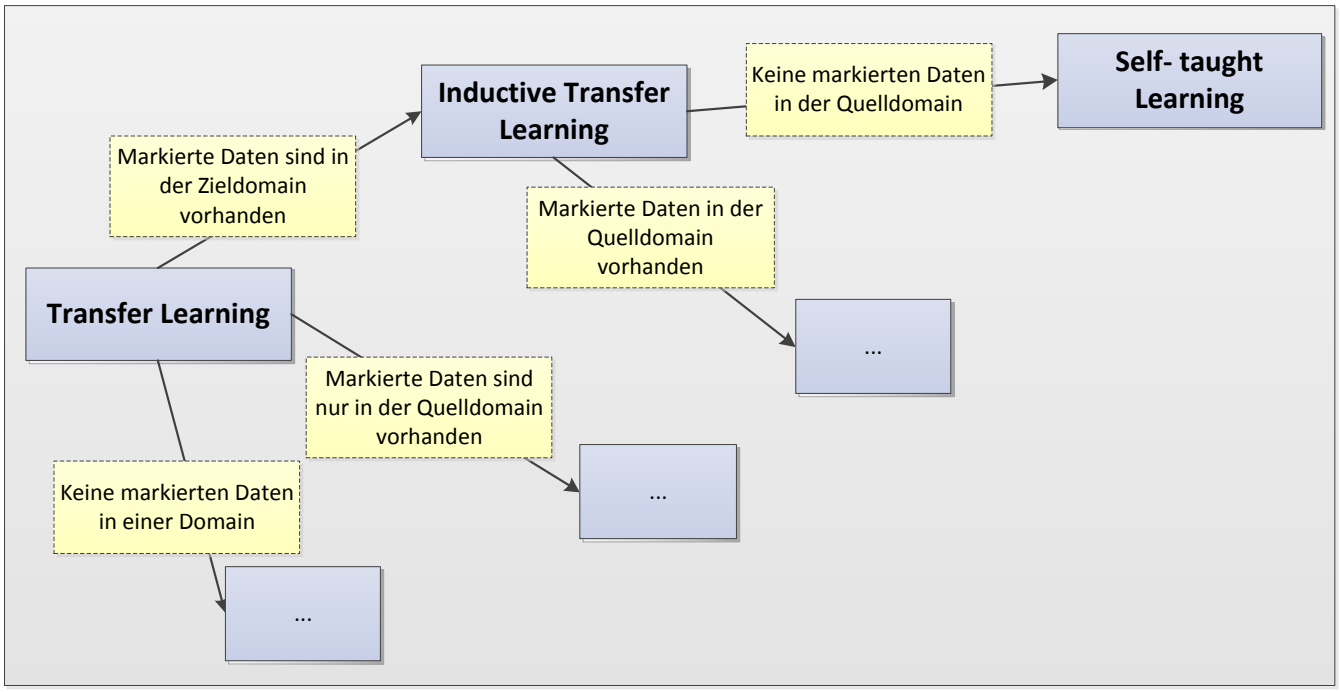


Abbildung 6.2-4: Einordnung in die „transfer learning“- Hierarchie [1]

6.3. Die Problemstellung bei Self- taught learning

Beim autodidaktischen Lernen ist ein Trainingsset mit einer endlichen *Datenmenge* mit m Elementen i.i.d.[4] in einer Distribution gegeben. Diese Datenmenge besteht aus je paarweise auftretenden Kombinationen aus einem *markierten Eingangs- Feature- Vektor* und dem *korrespondierenden Klassenlabel*.

Weiter ist eine Menge mit k Elementen aus unmarkierten Daten gegeben. Dabei ist entscheidend, dass diese Daten nicht aus derselben Distribution kommen wie die markierten Daten.

Klar ist, dass die markierten und unmarkierten Daten nicht komplett irrelevant untereinander sind, die unmarkierten Daten können der Klassifizierungseinheit weiterhelfen.

Es wird davon ausgegangen, dass die beiden Datensätze vom selben Eingabetyp sind, also Dokumente, Audio Dateien, Bilder etc. Wird nun ein autodidaktischer Algorithmus auf diese Daten angewandt, wirft dieser eine Hypothese aus. Diese Hypothese versucht die Beziehung, die zwischen den Daten besteht zu imitieren und auf andere Daten zu übertragen, um diese wiederum kategorisieren zu können. Durch diese Kategorisierung können neue Daten dann den bekannten Distributionen zugeteilt werden.

[4] i.i.d. = *independent and identically distributed, sowohl unabhängig als auch identisch verteil.*

6.4.„Self- taught learning“- Algorithmus

Der Algorithmus, der zum autodidaktischen Lernen eingesetzt werden soll, basiert auf dem *Sparse Coding*- Algorithmus von Olshausen & Field (1996).

Eine modifizierte Version dieses Algorithmus gab die Erkenntnis für das Lernen *von Hight- level Represantation*. Genauer gesagt, bedeutet das:

Gibt es unmarkierte Daten: $\{\boldsymbol{x_u^{(1)}}, \dots, \boldsymbol{x_u^{(k)}}\}$ **mit** $\boldsymbol{x_u^{(i)}} \in \mathbb{R}^{\boldsymbol{n}}$

Diese unmarkierten Daten werden genutzt um eine prägnantere Darstellung der Eingänge auf einem etwas höheren Niveau zu lernen.

Es ergibt sich folgendes Optimierungsproblem ⊙:

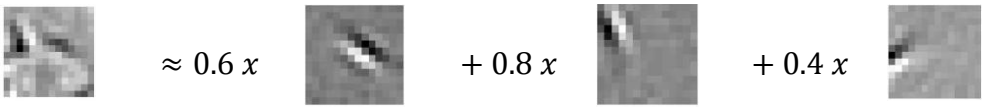
$$\boldsymbol{minimize_{b,a}} \quad \sum_i ||\boldsymbol{x_u^{(i)}} - \sum_j \boldsymbol{a_j^{(i)} b_j}||_2^2 + \boldsymbol{\beta}\, ||\boldsymbol{a^{(i)}}||_1 \quad \text{mit} \quad ||\boldsymbol{b_j}||_2 \leq \boldsymbol{1}, \ \forall \boldsymbol{j} \in \boldsymbol{1}, \dots, \boldsymbol{s}$$

<u>Die Optimierungsvariablen sind:</u>

Die *Basisvektoren* $\boldsymbol{b} = \{\boldsymbol{b_1}, \boldsymbol{b_2}, \dots, \boldsymbol{b_s}\}$ **mit** $\boldsymbol{b_j} \in \mathbb{R}^{\boldsymbol{n}}$

beschreiben ein Bildausschnitt und bilden damit einen robusten Kantendetektor.

<u>Dazu ein Beispiel:</u>



$\approx 0.6\, x$ $+ 0.8\, x$ $+ 0.4\, x$

x	b_{142}	b_{381}	b_{497}

Tabelle 6.4-1: Beispiel für Basisvektoren

Links wird ein Bildausschnitt angezeigt und rechts davon die dazu sparse gewichtete Kombination aus Basen.

Die *Aktivitäten* $\boldsymbol{a} = \{\boldsymbol{a^{(1)}}, \dots, \boldsymbol{a^{(k)}}\}$ **mit** $\boldsymbol{a^{(i)}} \in \mathbb{R}^{\boldsymbol{s}}$

gewichten die linear Kombinationen, die sich aus dem ersten Teil des oben aufgeführten Terms.

Die Anzahl der Basen s kann dabei viel größer sein, als die Anzahl der Eingangsdimension n.

<u>Es gibt dabei folgende Verbindungen:</u>

$\boldsymbol{a_j^{(i)}}$ **ist die Aktivität der Basis** $\boldsymbol{b_j}$ **für die eingehenden Daten** $\boldsymbol{x_u^{(i)}}$

Durch den Optimierungsansatz entstehen folgende zwei Effekte:

i. Der erste quadratische Term sorgt dafür, dass zu jedem Eingang $x_u^{(i)}$ eine gewichtete linear Kombination der Basis b_j konstruiert wird.
ii. Der zweite Teil der Funktion sorgt dafür, dass die Aktivitäten eine niedrige L_1- Norm einhalten, also dass die Aktivitäten „sparse[5]“ werden. Die meisten Elemente werden also null.

Das Problem ist konvex über die Teilmengen der Variablen a und b, wenn auch nicht zusammenhängend. Die Optimierung der Aktivitäten ist ein L_1 regularisiertes least squares problem[6] und die Optimierung der Basisvektoren b ist eine L_2 regularisiertes least squares problem. Diese Teilprobleme lassen sich effizient beheben und zusammen mit dem Optimierungsproblem iterativ optimieren.

Mit diesem Sparse Coding Algorithmus als Grundlage, werden als nächstes besondere Merkmale von Daten betrachtet und in den Algorithmus integriert. Bei diesen Merkmalen handelt es sich unter anderem um Strichstärken bei Schriften, Anzahl der Kanten in einem Bild oder gemeinsame Töne in Sprachen. Um diese Features $\hat{a}\left(x_l^{(i)}\right)$ mit in die Suchergebnisse einfließen zu lassen werden diese wie folgt berechnet:

$$\hat{\boldsymbol{a}}\left(\boldsymbol{x}_{\boldsymbol{l}}^{(\boldsymbol{i})}\right) = \mathbf{arg\,min}_{\boldsymbol{a}^{(i)}} \,||\, \boldsymbol{x}_{\boldsymbol{l}}^{(\boldsymbol{i})} - \sum_{\boldsymbol{j}} \boldsymbol{a}_{\boldsymbol{j}}^{(\boldsymbol{i})} \boldsymbol{b}_{\boldsymbol{j}} ||_2^2 \; + \boldsymbol{\beta} \, ||\boldsymbol{a}^{(\boldsymbol{i})}\,||_1$$

Diese Funktion approximiert und drückt $x_l^{(i)}$ als eine dünnbesetzte Linearkombination zu der Basis b_j aus.

Der dünnbesetzte Vektor $\hat{a}\left(x_l^{(i)}\right)$ ist nun der neue Repräsentant von $x_l^{(i)}$. Er enthält nun zusätzlich die Feature Eigenschaften.

Mit dieser Erweiterung ergibt sich für das obige Beispiele aus Abbildung 6.1 folgender Feature-Vektor für das Eingangsbild x:

$$\hat{a} \in \mathbb{R}^{512} \quad mit \quad \hat{a}_{142} = 0.6, \qquad \hat{a}_{381} = 0.8, \qquad \hat{a}_{497} = 0.4$$

[5] Sparce = Eine dünn-oder schwach besetzt, es entstehen so viele Null-Einträge, sodass diese genutzt werden können.
[6] Methode der kleinsten Quadrate = Standardverfahren zur Ausgleichsrechnung, es wird zu einer Datenpunktwolke eine Kurve gesucht.

Durch diese Erweiterung des Sparse Coding Algorithmus ergibt sich folgender autodidaktischer Algorithmus:

Eingabe:	Markierte Trainingsdaten	$T = \{\left(\mathrm{x}_\mathrm{l}^{(1)}, y^{(1)}\right), \left(\mathrm{x}_\mathrm{l}^{(2)}, y^{(2)}\right), \dots, \left(\mathrm{x}_\mathrm{l}^{(\mathrm{m})}, y^{(m)}\right)\}$
	Unmarkierte Daten	$\{\mathrm{x}_\mathrm{u}^{(1)}, \mathrm{x}_\mathrm{u}^{(2)}, \dots, \mathrm{x}_\mathrm{u}^{(\mathrm{k})}\}$
Ausgabe:	Geschulter Klassifikator für die Klassifizierungseinheit.	
Algorithmus:	Benutzen der unmarkierten Daten $\{\mathrm{x}_\mathrm{u}^{(\mathrm{i})}\}$, behebt das Optimierungsproblem ⊙ um die Basisvektoren b zu erhalten. Berechnen der Features für die Klassifizierungseinheit, die ein neues Trainingsset $\hat{T} = \left\{\left(\hat{\mathrm{a}}\left(\mathrm{x}_\mathrm{l}^{(\mathrm{i})}\right), y^{(i)}\right)\right\}_{i=1}^{m}$ mit $\hat{a}\left(x_l^{(i)}\right) = \arg\min_{a^{(i)}} \| x_l^{(i)} - \sum_j a_j^{(i)} b_j \|_2^2 + \beta \|a^{(i)}\|_1$ markierten Daten erzeugt. Einem Klassifikator $\mathcal{C}$ das Erlernte, bezüglich des markierten Trainingssets $\hat{\mathrm{T}}$, beibringen.	
Rückgabe:	Den gelernten Klassifikator $\mathcal{C}$.	

6.5. Vergleich zu anderen Methoden

PCA

Einer der am häufigsten eingesetzten Algorithmen für das Lernen von unmarkierten Daten ist die Hauptkomponentenanalyse (PCA). Der Algorithmus dient dazu umfangreiche Datensätze zu strukturieren, zu vereinfachen und zu veranschaulichen, indem eine Vielzahl statistischer Variablen durch eine geringere Zahl möglichst aussagekräftiger Linearkombinationen (die „Hauptkomponenten“) genähert wird.

Das interessante an PCA ist, das die Hauptkomponenten eine Lösung für das Optimierungsproblem darstellen, sehr ähnlich wie bei dem autodidaktischem Algorithmus in Kapitel 5.4. PCA ist dahin gehende sehr praktisch, da obiges Optimierungsproblem mit einfacher numerischer Software gelöst werden kann, auch weitere Funktionen können auf Grund der orthogonalen Einschränkung einfach berechnet werden. Der PCA gehört zu der Familie der exponentiellen Algorithmen [3].

7. Zusammenfassung

Es gibt innerhalb der „*transfer learning*“ Methodik, einige verschiedene Richtungen, die sich voneinander unterscheiden. Ein neuer Trend, der vom traditionellen „*maschine learning*“ abweicht und neue Wege geht. In diesem Bereich ist seit dessen erstem Erscheinen schon einiges geschehen und viele Erkenntnisse wurden gewonnen. Alles spricht dafür, dass sich dieser Trend fortsetzt.

Die einzelnen Bereiche gehen ihren eigenen Weg um an das gewünschte Ziel zu gelangen. Speziell der Bereich autodidaktisches Lernen bzw. „*self- taught learning*“ bietet einen interessanten Ansatz und versucht über komplett ungelabelten Daten zu lernen. Mit einem vorgeschlagenen Algorithmus, der in dieser Arbeit bereits vorgestellt wurde, wird versucht ein möglichst gutes Ergebnis zu erlangen.

8. Fazit

Im Bereich „*transfer learning*“ gibt es viele Unterkategorien, die sich auf Grund ihrer Eigenschaften voneinander unterscheiden. Jeder dieser Bereiche verfolgt auf seine eigene Weise das gemeinsame große Ziel, die Übertragung von Wissen so effizient und mit möglichst wenig Verlust von Wissen umzusetzen. Dieses Ziel zu erreichen gilt es durch Forschung in den einzelnen Bereichen voran zu treiben. Das Potential der einzelnen Bereiche ist noch nicht erreicht, was in Zukunft noch zu spannenden Erkenntnissen führen wird.

Insbesondere der Bereich autodidaktisches Lernen wirkt in diesem Umfeld faszinierend, da auf sehr interessante Art und Weise versucht wird Wissen zu übertragen. In diesem Bereich wird es sicher noch spannend weiter gehen, um die bisherigen Ergebnisse noch performanter und mit besseren Ergebnissen durchführen zu können.

Die Informationen dieser Arbeit stammen hauptsächlich aus den Dokumenten *„ A Survey on Transfer Learning“ [1]* und *„Self- taught learning: transfer learning from unlabeld data“ [2]*, welche als Grundlage dieser Arbeit dienten.

9. Anhang

9.1. Abbildungsverzeichnis:

9.2. Literaturverzeichnis:

[1] Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 22, NO. 10, OCTOBER 2010 pages 1345 - 1359

[2] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, Andrew Y. Ng. Self- taught learning: transfer learning from unlabeld data. Proceedings of the 24th International Conference on Machine Learning (ICML'07)

[3] Honglak Lee, Rajat Raina, Alex Teichman, Andrew Y. Ng .(2009) Exponential Family Sparse Coding with Applications to Self- taught Learning. pages 1113 - 1119.

[4] Nigam, K., McCallum, A., Thrun, S., & Mitchell, T.(2000). Text classication from labeled and unlabeled documents using EM. Machine Learning, 39, 103{134.

[5] W. Dai, Q. Yang, G. Xue, and Y. Yu, “Boosting for TransferLearning,” Proc. 24th Int’l Conf. Machine Learning, pp. 193-200, June 2007.

[6] N.D. Lawrence and J.C. Platt, “Learning to Learn with the Informative Vector Machine,” Proc. 21st Int’l Conf. Machine earning, July 2004.