



گزارش پروژه پایانی

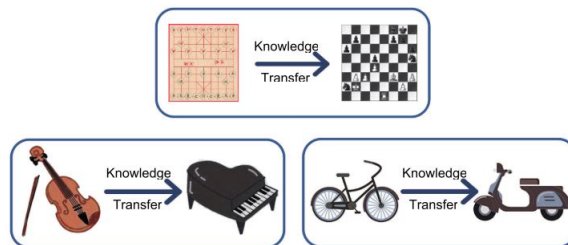
مریم سادات هاشمی
دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
m_hashemi94@comp.iust.ac.ir

چکیده

تهیه داده‌های دارای برچسب برای آموزش الگوریتم‌های یادگیری ماشین بسیار مهم است. با این حال، بدست آوردن تعداد کافی از داده‌های برچسب‌دار در کاربردهای واقعی اغلب گران و زمانبر است. بنابراین استفاده از روش‌هایی که دانش موجود در یک دامنه با برچسب‌گذاری مناسب (دامنه مبدا) را به یک دامنه با برچسب‌های معدود یا بدون برچسب (دامنه هدف) منتقل می‌کنند، موثر هستند. به این روش‌ها انتقال یادگیری، تطبیق دامنه و یا تطبیق توزیع می‌گویند. از آنجا که دامنه‌های مبدا و هدف دارای توزیع‌های مختلف هستند، روش‌های بیشمار برای کاهش واگرایی توزیع‌ها پیشنهاد شده است. در این پروژه به بررسی یک روش پارامتری به نام *BDA* و و یک روش غیرپارامتری به نام *EasyTL* می‌پردازیم.

۱ مقدمه

اگرچه یادگیری ماشین سنتی در سال‌های اخیر به موفقیت‌های بزرگی دست یافته و در بسیاری از برنامه‌های کاربردی با موفقیت مورد استفاده قرار گرفته است، اما هنوز هم برای برخی از مسائل دنیای واقعی محدودیت‌هایی دارد. سناریوی ایده‌آل برای یادگیری ماشین این است که تعداد زیادی داده‌های آموزشی دارای برچسب وجود داشته باشد به طوری که توزیع آن مشابه داده‌های تست باشد. با این حال، جمع‌آوری داده‌های آموزشی کافی در بسیاری از سناریوها بسیار گران‌قیمت، وقت‌گیر یا حتی غیرممکن است. یادگیری نیمه نظارتی^۱ می‌تواند تا حدودی این مشکل را حل کند. رویکرد نیمه نظارتی فقط به تعداد محدودی از داده‌های برچسب‌خورده نیاز دارد و از مقدار زیادی از داده‌های بدون برچسب برای بهبود دقت یادگیری استفاده می‌کند. اما در بسیاری از موارد، جمع‌آوری نمونه‌های بدون برچسب نیز دشوار است [۷].



شکل ۱: مثالی از انتقال یادگیری [۷].

^۱Semi-supervised Learning

انتقال یادگیری^۲، یک روش یادگیری ماشین برای حل مسئله فوق است. مفهوم انتقال یادگیری در ابتدا از روانشناسی ناشی شده است. طبق تئوری انتقال، انتقال یادگیری نتیجه عمومی شدن تجربه است. به این معنی که کسی که ویولن را یاد گرفته است می‌تواند پیانو را سریعتر از دیگران بیاموزد، زیرا ویولن و پیانو سازهای موسیقی هستند و ممکن است دانش مشترکی با هم داشته باشند. شکل ۱ چند مثال شهودی درباره انتقال یادگیری را نشان می‌دهد.

با الهام از توانایی‌های بشر برای انتقال دانش، انتقال یادگیری در یادگیری ماشین به این مفهوم است که دانش از یک حوزه مرتبط (به نام دامنه مبدا) برای بهبود عملکرد یادگیری در دامنه هدف منتقل می‌شود. گفتنی است، دانش منتقل شده همیشه تأثیر مثبتی در کارهای جدید نمی‌گذارد. اگر بین حوزه‌ها اشتراک کمی وجود داشته باشد، انتقال دانش می‌تواند ناموفق باشد. به عنوان مثال، یادگیری دوچرخه سواری نمی‌تواند به ما کمک کند سریعتر پیانو را یاد بگیریم. علاوه بر این، شباهت‌ها بین دامنه‌ها همیشه یادگیری را تسهیل نمی‌کند، زیرا گاهی اوقات شباهت‌ها ممکن است گمراه کننده باشد. به عنوان مثال، اگرچه اسپانیایی و فرانسوی رابطه نزدیکی با یکدیگر دارند، افرادی که اسپانیایی را یاد می‌گیرند ممکن است در یادگیری زبان فرانسه مشکلاتی از قبیل استفاده از واژگان اشتباه یا ترکیبات را تجربه کنند. این امر به این دلیل رخ می‌دهد که تجربه موفق قبلی در اسپانیایی می‌تواند در یادگیری کلمه، استفاده، تلفظ، ترکیب و غیره به زبان فرانسه اختلال ایجاد کند. در زمینه روانشناسی، این پدیده که تجربه قبلی در یادگیری کارهای جدید تأثیر منفی دارد، انتقال منفی نامیده می‌شود. به همین ترتیب، در حوزه انتقال یادگیری، اگر یادگیرنده هدف، تحت تأثیر منفی دانش منتقل شده باشد، به عنوان انتقال منفی نامیده می‌شود. این که آیا انتقال منفی اتفاق می‌افتد ممکن است به عوامل مختلفی بستگی داشته باشد از جمله ارتباط بین دامنه مبدا و هدف و ظرفیت یادگیرنده [۷].

اصطلاح دیگری که معمولاً در انتقال یادگیری استفاده می‌شود تطبیق دامنه^۳ است. تطبیق دامنه به فرایندی گفته می‌شود که یک یا چند دامنه مبدا برای انتقال دانش و بهبود عملکرد دامنه هدف تطبیق داده می‌شوند. انتقال یادگیری غالباً به فرآیند تطبیق دامنه متکی است که سعی در کاهش تفاوت بین دامنه‌ها دارد [۶].

در این پروژه ما دو روش انتقال یادگیری را بررسی می‌کنیم. روش اول یک روش پارامتری به نام BDA ^۴ است که تطبیق دامنه انجام می‌دهد [۴]. روش دوم یک روش غیرپارامتری به نام $EasyTL$ ^۵ است که نه تنها ویژگی‌ها بلکه طبقه‌بند را نیز از دامنه مبدا به دامنه هدف منتقل می‌کند. نکته‌ی جالب توجه این است که از روش BDA می‌توان در روش $EasyTL$ بهره برد [۵].

۲ ادبیات موضوع

مسئله انتقال یادگیری براساس معیارهای زیادی دسته‌بندی می‌شود. یکی از این دسته‌بندی‌ها براساس پایداری فضای ویژگی‌ها و فضای برجسب‌های دامنه مبدا و هدف است. اگر $Y_s = Y_t$ و $X_s = X_t$ در مسئله ما برقرار باشد؛ به این مسئله یک مسئله انتقال یادگیری همگون^۶ می‌گوییم و اگر $Y_s \neq Y_t$ و $X_s \neq X_t$ برقرار باشد؛ یک مسئله انتقال یادگیری ناهمگون^۷ می‌گوییم. هر دو روش معرفی شده در [۴] و [۵] جز مسائل انتقال یادگیری همگون هستند که مبتنی بر فضای ویژگی حل می‌شوند. از این رو در این بخش به بررسی مهمترین روش‌های مبتنی بر فضای ویژگی می‌پردازیم.

روش TCA ^۸ تمرکز را بر روی به حداقل رساندن فاصله توزیع حاشیه‌ای بین دامنه‌ها گذاشته است. در این روش پارامترهای زیادی برای تنظیم کردن وجود دارد. به همین علت از لحاظ محاسباتی هزینه زیادی دارد. از طرفی چون فقط به توزیع حاشیه‌ای پرداخته است و اهمیتی برای توزیع شرطی قائل نشده است؛ عملکرد خوبی را در مسائل واقعی از خود نشان نمی‌دهد [۳]. روش JDA ^۹ برای تطبیق دامنه از هر دو توزیع حاشیه‌ای و توزیع شرطی استفاده می‌کند اما اهمیت هر دو توزیع را یکسان در نظر می‌گیرد و خاصیت تعمیم‌دهی در مسائل کاربردی و واقعی را ندارد [۲]. روش GFK ^{۱۰} یک روش مبتنی بر کرنل است که به بهره برداری از ساختارهای کم بعدی موجود در داده می‌پردازد. رویکردی که این مدل دارد از نظر محاسباتی سودمند است زیرا تنها یک پارامتر دارد که نیاز به تنظیم دارد و آن پارامتر هم تعداد بعدها و ویژگی است. اما این روش زمانی که فاصله‌ی توزیع‌های حاشیه‌ای زیاد است عملکرد خوبی ندارد [۱].

^۲ Transfer Learning
^۳ Domain Adaptation
^۴ Balanced Distribution Adaptation
^۵ Easy Transfer Learning
^۶ homogeneous transfer learning
^۷ heterogeneous transfer learning
^۸ Transfer Component Analysis
^۹ Joint Distribution Adaptation
^{۱۰} Geodesic Flow Kernel

۳ شرح روش پیشنهادی مقالات

در این بخش ابتدا مسئله تطبیق دامنه را به صورت ریاضی بیان می‌کنیم. سپس مشکلات و چالش‌هایی که در حل این مسئله وجود دارد را بیان خواهیم کرد. در ادامه نیز توضیح خواهیم داد که هر کدام از مقاله‌ها چه روشی را برای حل این مشکلات و چالش‌ها ارائه داده‌اند.

۱.۳ تعریف ریاضی مسئله تطبیق دامنه

ما یک دامنه‌ی برچسب خورده به نام دامنه مبدا و یک دامنه برچسب نخورده به نام دامنه هدف داریم که به ترتیب به صورت $\{x_{t_j}\}_{j=1}^m$ و $\{x_{s_i}, y_{s_i}\}_{i=1}^n$ تعریف می‌شود. همچنین فرضیاتی را در مورد فضای ویژگی‌ها $X_s = X_t$ و فضای برچسب‌ها $Y_s = Y_t$ و احتمال حاشیه‌ای $P_s(x_s) \neq P_t(x_t)$ و احتمال شرطی $P_s(y_s|x_s) \neq P_t(y_t|x_t)$ در نظر می‌گیریم. بنابراین هدف انتقال یادگیری این است که فضای برچسب‌های دامنه هدف Y_t را به کمک دامنه مبدا D_s یاد بگیرد. تطبیق دامنه تلاش می‌کند که با کاهش واگرایی احتمال‌های حاشیه‌ای و احتمال‌های شرطی بین دامنه‌ی مبدا و هدف، مسئله انتقال یادگیری را حل کند به عبارت دیگر واگرایی بین $(1, P_s(x_s))$ و $(2, P_t(x_t))$ و $P_s(y_s|x_s)$ و $P_t(y_t|x_t)$ را به حداقل می‌رساند.

۲.۳ تطبیق متوازن توزیع در انتقال یادگیری^{۱۱}

بسیاری از روش‌های تطبیق توزیع موجود، یکی از توزیع‌های حاشیه‌ای یا شرطی و یا هر دو را تطبیق می‌دهند. در مقاله‌های اخیر ثابت شده است که استفاده از هر دو توزیع عملکرد بهتری را نتیجه می‌دهد. اما روش‌های کنونی تاثیر هر دو توزیع را یکسان در نظر می‌گیرند در حالی که وقتی مجموعه داده‌ها بسیار متفاوت باشند، به این معنی است که توزیع حاشیه‌ای اهمیت بیشتری دارد و وقتی مجموعه داده‌ها مشابه هستند، به این معنی است که توزیع شرطی به توجه بیشتری نیاز دارد. از این رو، لازم است از هر دو توزیع با وزنی مناسب با اهمیت آن‌ها برای تطبیق استفاده نمود. علاوه بر این، نامتوازن بودن مجموعه داده بر روی تطبیق دامنه تاثیرگذار است که در روش‌های موجود، این موضوع در نظر گرفته نمی‌شود. برای حل این دو مشکل، در [۴] دو روش پیشنهاد شده است. روش اول BDA است که نه تنها می‌تواند توزیع‌های حاشیه‌ای و شرطی بین حوزه‌ها را تطبیق دهد، بلکه اهمیت این دو توزیع را به صورت متوازن تنظیم می‌کند. روش دوم W-BDA است که مشکل نامتوازن بودن مجموعه داده را حل می‌کند. W-BDA می‌تواند هنگام انجام تطبیق توزیع، وزن هر کلاس را به طور انطباقی تغییر دهد.

۱.۲.۳ BDA

همانطور که توضیح دادیم در تطبیق دامنه، با کاهش واگرایی یا فاصله‌ی بین احتمال‌های حاشیه‌ای و شرطی می‌توانیم دامنه‌های مبدا و هدف را تطبیق دهیم. این موضوع را می‌توانیم به صورت عبارت زیر بیان کنیم:

$$Dist(D_s, D_t) \approx Dist(P(x_s), P(x_t)) + Dist(P(y_s|x_s), P(y_t|x_t)) \quad (۱)$$

روش BDA به عبارت ۱ یک ضریب توازن به نام μ اضافه می‌کند که میزان اهمیت توزیع حاشیه‌ای و شرطی را در مسائل مختلف تنظیم و کنترل می‌کند. پس عبارت ۱ تبدیل می‌شود به:

$$Dist(D_s, D_t) \approx (1 - \mu)Dist(P(x_s), P(x_t)) + \mu Dist(P(y_s|x_s), P(y_t|x_t)) \quad (۲)$$

برای حل عبارت ۲ از روش MMD^{۱۲} برای تخمین توزیع‌های حاشیه‌ای و توزیع‌های شرطی استفاده می‌کنیم. بنابراین خواهیم داشت:

$$D(D_s, D_t) \approx (1 - \mu) \left\| \frac{1}{n} \sum_{i=1}^n x_{s_i} - \frac{1}{m} \sum_{j=1}^m x_{t_j} \right\|_H^2 + \mu \sum_{c=1}^C \left\| \frac{1}{n_c} \sum_{x_{s_i} \in D_s^{(c)}} x_{s_i} - \frac{1}{m_c} \sum_{x_{t_j} \in D_t^{(c)}} x_{t_j} \right\|_H^2 \quad (۳)$$

در عبارت ۳ ترم اول فاصله‌ی توزیع‌های حاشیه‌ای بین دامنه‌ها را نشان می‌دهد و ترم دوم هم فاصله‌ی توزیع‌های شرطی بین دامنه‌ها را نمایش می‌دهد. با بهره‌گیری از تکنیک‌های ماتریس و جبر خطی عبارت ۳ را می‌توان به صورت زیر نوشت^{۱۳}:

$$\min_{s.t.} \quad tr \left(A^T X \left((1 - \mu) M_0 + \mu \sum_{c=1}^C M_c \right) X^T A \right) + \lambda \|A\|_F^2 \quad (۴)$$

$$s.t. \quad A^T X H X^T A = I, \quad 0 \leq \mu \leq 1$$

^{۱۱}Balanced Distribution Adaptation for Transfer Learning

^{۱۲}Maximum Mean Discrepancy

^{۱۳}لطفاً برای مشاهده جزئیات بیشتر عبارت ۴ به مقاله‌ی اصلی مراجعه نمایید.

ورودی: ماتریس ویژگی X_s و X_t ، بردار برچسب مبدا y_s ، تعداد بعد ها d ، ضریب توازن μ ، پارمتر رگلاریسیون λ .

خروجی: ماتریس تبدیل A و طبقه‌بند f .

- ۱: یک طبقه‌بند پایه را روی X_s آموزش دهید و پیش‌بینی را روی X_t اعمال کنید تا برچسب‌های آن را بدست آورید. ماتریس $X = [X_s, X_t]$ را بسازید. ماتریس‌های M_0 و M_c را مقداردهی اولیه کنید. (یا W_c برای W -BDA)
- ۲: مراحل زیر را تا وقتی که همگرایی رخ دهد ادامه دهید.
- ۳: معادله (۵) را حل کنید. (یا معادله (۸) برای W -BDA) و از d کوچکترین بردار برای ساخت A استفاده کنید.
- ۴: طبقه‌بند f را بر روی $\{A^T X_s, y_s\}$ آموزش دهید.
- ۵: برچسب‌های $A^T X_t = f(A^T X_t)$ را به‌روز کنید.
- ۶: ماتریس M_c را به‌روز کنید. (یا W_c برای W -BDA)
- ۷: طبقه‌بند f را برگردانید.

شکل ۲

در عبارت ۴ دو ترم وجود دارد: ۱) تطبیق توزیع‌های حاشیه‌ای و شرطی به همراه ضریب توازن μ (۲) ترم مربوط به رگلاریسیون با ضریب رگلاریسیون λ . در این عبارت ۲ قید هم داریم. قید اول بیان می‌کند که داده‌ی تغییر یافته خصوصیات داخلی اصل داده را حفظ می‌کند. قید دوم نیز بازه‌ی ضریب توازن را نشان می‌دهد.

برای حل عبارت ۴ از ضرائب لاگرانژ استفاده می‌کنیم. بنابراین اگر ضرائب لاگرانژ $\phi = \{\phi_1, \phi_2, \dots, \phi_d\}$ باشد، خواهیم داشت:

$$L = tr \left(A^T X \left((1 - \mu) M_0 + \mu \sum_{c=1}^C M_c \right) X^T A \right) + \lambda \|A\|_F^2 + tr \left((I - A^T X H X^T A) \phi \right) \quad (۵)$$

برای بدست آوردن ماتریس تبدیل A کافی است از عبارت ۵ نسبت به A مشتق بگیریم و برابر صفر بگذاریم. در این صورت به عبارت زیر دست می‌یابیم:

$$\left(X \left((1 - \mu) M_0 + \mu \sum_{c=1}^C M_c \right) X^T + \lambda I \right) A = X H X^T A \phi \quad (۶)$$

عبارت ۶ همان معادله‌ی معروف بردار و مقدار ویژه است. پس با پیدا کردن d بردار ویژه با کمترین مقدار ویژه خواهیم توانست ماتریس تبدیل A را بدست آوریم. در اینجا سوالی پیش می‌آید که چرا مقدار بهینه μ را بدست نیاوردیم. نکته‌ی حائز اهمیت در اینجا این است که پارامتر μ یک پارامتر آزاد نیست که بتوانیم آن را تخمین بزنیم بلکه μ با توجه به توزیع داده‌ها تخمین زده می‌شود. بنابراین در واقعیت باید با توجه به کاربرد و با استفاده از روش ارزیابی متقابل^{۱۴} مقدار بهینه μ را بدست آوریم.

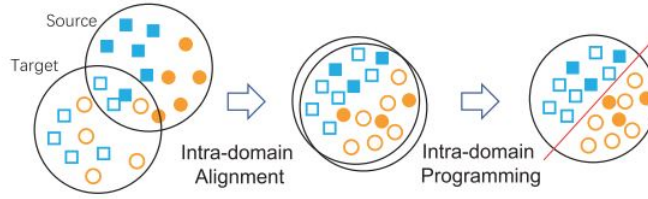
۲.۲.۳ W-BDA

روش W-BDA^{۱۵} همانند روش BDA است با این تفاوت که با در نظر گرفتن نامتوازن بودن کلاس‌ها در داده‌ها تخمین دقیق‌تری را برای احتمال شرطی تعریف می‌کند. همانطور که می‌دانیم در واقعیت ما نمی‌توانیم احتمال $P(y|x)$ را به صورت مستقیم تخمین بزنیم به همین دلیل با استفاده از قانون بیز به جای آن احتمال $P(x|y)$ را تخمین می‌زنیم. روش W-BDA برای تخمین احتمال پیشین داده‌های موجود در دامنه مبدا و هدف به ترتیب از دو ضریب α_s و α_t استفاده می‌کند که با استفاده از آن‌ها تناسب کلاس‌های هر دامنه را متوازن می‌کند. اگر بخواهیم این مسئله را به صورت ریاضی بیان کنیم خواهیم داشت:

$$\begin{aligned} Dist(P(y_s|x_s), P(y_t|x_t)) &= \|P(y_s|x_s) - P(y_t|x_t)\|_H^2 \\ &= \left\| \frac{P(y_s)}{P(x_s)} P(y_s|x_s) - \frac{P(y_t)}{P(x_t)} P(y_t|x_t) \right\|_H^2 \\ &= \|\alpha_s P(y_s|x_s) - \alpha_t P(y_t|x_t)\|_H^2 \end{aligned} \quad (۷)$$

^{۱۴} Cross Validation

^{۱۵} Weighted Balanced Distribution Adaptation



شکل ۳: فرآیند روش EasyTL [۵].

بنابراین با توجه به توضیحات بالا در روش W-BDA کافی است معادله ۴ را به معادله ۱۶ زیر تغییر دهیم^{۱۶}:

$$\begin{aligned} \min \quad & \text{tr} \left(A^T X \left((1 - \mu) M_0 + \mu \sum_{c=1}^C W_c \right) X^T A \right) + \lambda \|A\|_F^2 \\ \text{s.t.} \quad & A^T X H X^T A = I, \quad 0 \leq \mu \leq 1 \end{aligned} \quad (8)$$

حل معادله ۸ مانند معادله ۴ است.

به طور خلاصه الگوریتم دو روش BDA و W-BDA را می‌توانید در شکل ۲ مشاهده کنید.

۳.۳ انتقال یادگیری آسان با استفاده از ساختارهای درون دامنه^{۱۷}

بیشتر روش‌های سنتی و عمیق در انتقال یادگیری، روش‌های پارامتری است که باید طی فرآیندی بسیار گران قیمت و وقت گیر پارامترهای آن تنظیم شود. ارزیابی متقابل، که متداول‌ترین استراتژی برای انتخاب مدل‌ها و تنظیم پارامترها است، در انتقال یادگیری قابل استفاده نیست زیرا اغلب داده‌های برجسته خورده در دامنه هدف وجود ندارد. اگرچه روش‌های اخیر AutoML می‌تواند به طور خودکار پارامترها را از طریق هرس درخت و شبکه‌های عصبی تنظیم کنند اما آن‌ها قادر به مدیریت توزیع‌های مختلف بین دامنه‌ها نیستند و به طور معمول مدت زمان زیادی برای همگرایی لازم دارند. در [۵] الگوریتمی پیشنهاد شده است که روشی ساده اما کارآمد را پیشنهاد می‌دهد که علاوه بر این که مشکلات ذکر شده را حل می‌کند، این امکان را فراهم می‌کند که بتوان الگوریتم یادگیری انتقال را بر روی دستگاه‌های کوچک که دارای منابع محاسباتی محدود هستند نیز اجرا کرد. اسم این روش را EasyTL نامیدند. EasyTL قادر است بدون نیاز به انتخاب مدل و تنظیم هایپر پارامترها، انتقال دانش را در دامنه‌ها انجام دهد. EasyTL با بهره برداری از ساختارهای درون دامنه، هم انتقال ویژگی‌ها و هم انتقال طبقه‌بند را به صورت غیر پارامتری می‌آموزد. برای انتقال ویژگی‌ها روشی به نام *intra-domain alignment* و برای انتقال طبقه‌بند روشی به نام *intra-domain programming* را معرفی کرده است (شکل ۳). در ادامه به توضیح جزئیات هر کدام از این روش‌ها می‌پردازیم.

۱.۳.۳ intra-domain programming

در این بخش می‌خواهیم بررسی کنیم که روش *intra-domain programming* چگونه می‌تواند طبقه‌بند دامنه مبدا را به دامنه هدف منتقل کند. قبل از این که به جزئیات این روش بپردازیم، مفهومی را به نام ماتریس احتمال M معرفی می‌کنیم. سطرهای این ماتریس تعداد کلاس‌ها و ستون‌های آن برابر با تعداد نمونه‌های موجود در مجموعه داده است. هر درایه از این ماتریس نشان می‌دهد که چقدر احتمال دارد یک نمونه متعلق به کلاس آن سطر باشد. بنابراین جمع احتمال‌های یک ستون در این ماتریس برابر یک است. (شکل ۴)

روش *intra-domain programming* به جای این که مستقیماً y_t را بیاموزد، تلاش می‌کند تا ماتریس احتمال M را بدست آورد. از این رو تابع هزینه به صورت زیر تعریف می‌شود:

$$\mathbf{J} = \sum_j^{n_t} \sum_c^C D_{cj} M_{cj} \quad (9)$$

در عبارت ۹ ماتریس D فاصله است که درایه D_{cj} نشان‌دهنده فاصله نمونه x_j^t تا مرکز c امین کلاس در دامنه مبدا است. اگر h_c را به عنوان مرکز c امین کلاس در دامنه مبدا در نظر بگیریم؛ در این صورت D_{cj} و h_c به ترتیب به صورت

^{۱۶} لطفاً برای دیدن جزئیات ماتریس W_c در معادله ۸ به مقاله اصلی مراجعه نمایید.
^{۱۷} Easy Transfer Learning By Exploiting Intra-domain Structures

$$M = \begin{matrix} & \begin{matrix} x_1^t & x_2^t & x_3^t & \dots & x_n^t \end{matrix} \\ \begin{matrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{matrix} & \begin{bmatrix} 0.1 & 0.2 & 0.5 & \dots & \dots \\ 0.2 & 0.4 & 0.1 & \dots & \dots \\ 0.3 & 0.2 & 0.2 & \dots & \dots \\ 0.4 & 0.2 & 0.2 & \dots & \dots \end{bmatrix} \end{matrix}$$

شکل ۴: یک مثال از ماتریس احتمال M.

زیر محاسبه می‌شود:

$$D_{cj} = \left\| x_t^j - h_c \right\| \quad (10)$$

$$h_c = \frac{1}{|\Omega_s^{(c)}|} \sum_i^{n_s} x_i^s \cdot \mathbb{I}(y_i^s = c) \quad (11)$$

اکنون برای آن که ماتریس احتمال M را بدست آوریم باید تابع هزینه معادله ۹ را به حداقل برسانیم. بدین منظور قیدهایی برای این تابع تعریف می‌شود. اولین قید این است که جمع احتمال هر ستون از ماتریس M برابر با یک باشد. قید دوم این است که حداقل باید برای هر کلاس یک نمونه وجود داشته باشد و قید سوم هم همانطور که می‌دانیم مقدار هر درایه از ماتریس M باید یک عدد در بازه صفر تا یک باشد. بنابراین به بیان ریاضی داریم:

$$\begin{aligned} \min \quad & \mathbf{J} = \sum_j^{n_t} \sum_c^C D_{cj} M_{cj} \quad s.t \\ & \sum_c^C M_{cj} = 1, \quad \forall j \in 1, \dots, n_t, \\ & \sum_j^{n_t} M_{cj} \geq 1, \quad \forall c \in 1, \dots, C, \\ & 0 \leq M_{cj} \leq 1 \end{aligned} \quad (12)$$

برای حل عبارت ۱۲ از کتابخانه PuLP در پایتون استفاده می‌کنیم و در نهایت ماتریس M بدست می‌آید. در گام آخر برچسب نمونه x_j^t به شکل زیر محاسبه می‌گردد:

$$y_j^t = \underset{r}{argmax} \frac{M_{rj}}{\sum_c^C M_{cj}} \quad for \quad r \in \{1, \dots, C\} \quad (13)$$

نکته قابل توجه در این روش این است که هیچ پارامتری در این روش وجود ندارد که نیاز به تنظیم کردن داشته باشد. به همین علت است که این روش، یک روش غیر پارامتری است.

۲.۳.۳ intra-domain alignment

همان‌طور که در شکل ۳ مشخص است؛ هدف از *intra-domain alignment* تطبیق ویژگی‌ها بین دامنه‌ی مبدا و هدف است که باعث کم شدن واگرایی بین دامنه‌ها می‌شود. روش‌های زیادی برای این کار وجود دارد مانند BDA و GFK. اما در اینجا ما از روش CORAL^{۱۸} استفاده می‌کنیم که هم از لحاظ محاسباتی کارآمد است و هم یک روش غیرپارامتری است. فرمولی که این روش برای تطبیق دامنه‌ها معرفی می‌کند به صورت زیر است:

$$z^r = \begin{cases} x^r \cdot (cov(x^s) + E_s)^{-\frac{1}{2}} \cdot (cov(x^t) + E_t)^{\frac{1}{2}} & if \quad r = s \\ x^r & if \quad r = t \end{cases} \quad (14)$$

الگوریتم روش EasyTl را به طور خلاصه می‌توانید در شکل ۵ مشاهده کنید.

CORrelation ALignment^{۱۸}

ورودی: ماتریس ویژگی x_s و x_e برای Ω_s و Ω_e و بردار برجسب y_s برای Ω_s .

خروجی: بردار برجسب پیش بینی شده برای دامنه هدف.

۱: (اختیاری) انجام intra-domain alignment.

۲: بدست آوردن ماتریس احتمال M و محاسبه y_e با حل معادله (۱۲).

۳: بردار برجسب y_e برای Ω_e .

شکل ۵

۴ پیاده سازی و نتایج

همان طور که در [۵] بیان شده است در روش EasyTL باید دو بخش *Intra-domain programming* و *Intra-domain alignment* را پیاده سازی کنیم. بخش *Intra-domain programming* شامل ۳ مرحله می باشد:

۱. محاسبه ی بردار مراکز کلاس های دامنه ی مبدا h_c با استفاده از معادله ی ۱۱: این قسمت در تابع $get_class_center(X_s, Y_s)$ پیاده سازی شده است.

۲. محاسبه ی ماتریس فاصله D با استفاده از معادله ی ۱۰: این قسمت در تابع $get_distance_matrix(X_t, class_center)$ پیاده سازی شده است.

۳. بدست آوردن ماتریس احتمال M با استفاده از معادله ی ۱۲ و بدست آوردن برجسب دامنه هدف: این قسمت در تابع $solve_LP(C, nt, D_{cj})$ پیاده سازی شده است.

سپس از نتایج این سه تابع استفاده می کنیم و آن ها را در تابع $intra_domain_programming(X_s, Y_s, X_t, Y_t)$ با هم ترکیب می کنیم. در بخش *Intra-domain alignment* کافی است تنها معادله ی ۱۴ را پیاده سازی کنیم. بدین منظور از تابع $intra_domain_alignment(X_s, X_t)$ استفاده می کنیم.

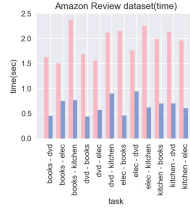
روش EasyTL را می توانیم به دو صورت اجرا کنیم. در یک حالت فضای دامنه ها را بایکدیگر تراز نمی کنیم و فقط بخش *Intra-domain programming* را اجرا می کنیم و در حالت دیگر ابتدا فضای دامنه ها را به یکدیگر تراز می کنیم و سپس طبقه بند موجود در دامنه مبدا را به دامنه هدف منتقل می کنیم. این روش را بر روی ۴ مجموعه داده آزمایش می کنیم و نتایج را مقایسه می کنیم. این ۴ مجموعه داده به شرح زیر هستند:

۱. *Amazon Review* یک مجموعه داده تجزیه و تحلیل احساسات است که شامل بررسی های مثبت و منفی چهار نوع محصول است: لوازم آشپزخانه، دی وی دی، الکترونیک و کتاب.

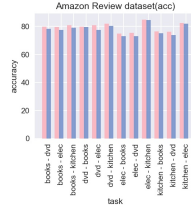
۲. *Office-Caltech* شامل ۱۰ کلاس از تصاویر در آمازون، DSLR، وب کم و Caltech است.

۳. *Image-CLEF DA* شامل ۱۲ دسته تصویر متعلق به ۳ حوزه است: Caltech، ImageNet و Pascal.

۴. *Office-Home* شامل ۱۵،۵۰۰ تصویر از ۶۵ دسته از ۴ حوزه، Clipart Art، Product و دنیای واقعی است.



(د) مقایسه زمان



(ج) مقایسه دقت

Accuracy on Amazon Review dataset

Task	EasyTL	EasyTL without alignment
books - dvd	79.8 %	78.4 %
books - elec	79.7 %	77.5 %
books - kitchen	80.9 %	79.2 %
dvd - books	79.9 %	79.5 %
dvd - elec	80.8 %	77.4 %
dvd - kitchen	82.0 %	80.4 %
elec - books	75.0 %	73.0 %
elec - dvd	75.3 %	73.1 %
elec - kitchen	84.9 %	84.6 %
kitchen - books	76.5 %	75.1 %
kitchen - dvd	76.3 %	73.8 %
kitchen - elec	82.5 %	82.0 %

(ب) نتایج پیاده‌سازی

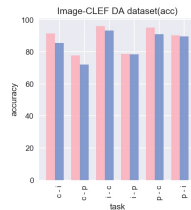
Method	EasyTL(c)	EasyTL
B → D	78.4	79.8
B → E	77.5	79.7
B → K	79.2	80.9
D → B	79.5	79.9
D → E	77.4	80.8
D → K	80.4	82.0
E → B	73.0	75.0
E → D	73.1	75.3
E → K	84.6	84.9
K → B	75.2	76.5
K → D	73.8	76.3
K → E	82.0	82.5

(آ) نتایج مقاله

شکل ۶: نتایج روش EasyTL بر روی مجموعه داده Amazon Review



(د) مقایسه زمان



(ج) مقایسه دقت

Accuracy on Image-CLEF DA dataset

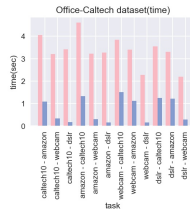
Task	EasyTL	EasyTL without alignment
c - i	91.5 %	85.5 %
c - p	77.7 %	72.0 %
i - c	96.0 %	93.3 %
i - p	78.7 %	78.5 %
p - c	95.0 %	91.0 %
p - i	90.3 %	89.5 %

(ب) نتایج پیاده‌سازی

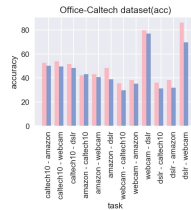
ID	Task	EasyTL(c)	EasyTL
1	C → I	85.5	91.5
2	C → P	72.0	77.7
3	I → C	93.3	96.0
4	I → P	78.5	78.7
5	P → C	91.0	95.0
6	P → I	89.5	90.3
-	AVG	85.0	88.2

(آ) نتایج مقاله

شکل ۷: نتایج روش EasyTL بر روی مجموعه داده Image-CLEF DA



(د) مقایسه زمان



(ج) مقایسه دقت

Accuracy on Office-Caltech dataset

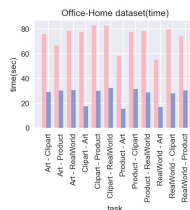
Task	EasyTL	EasyTL without alignment
caltech10 - amazon	52.6 %	50.1 %
caltech10 - webcam	53.9 %	49.5 %
caltech10 - dsir	51.6 %	48.4 %
amazon - caltech10	42.3 %	43.0 %
amazon - webcam	43.1 %	40.7 %
amazon - dsir	48.4 %	38.9 %
webcam - caltech10	35.4 %	29.7 %
webcam - amazon	38.2 %	35.2 %
webcam - dsir	79.6 %	77.1 %
dsir - caltech10	36.1 %	31.2 %
dsir - amazon	38.3 %	31.9 %
dsir - webcam	86.1 %	69.5 %

(ب) نتایج پیاده‌سازی

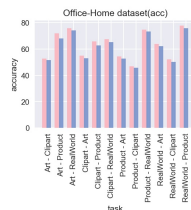
Method	EasyTL(c)	EasyTL
C → A	50.1	52.6
C → W	49.5	53.9
C → D	48.4	51.6
A → C	43.0	42.3
A → W	40.7	43.1
A → D	38.9	48.4
W → A	29.7	35.4
W → C	35.2	38.2
W → D	77.1	79.6
D → C	31.2	36.1
D → A	31.9	38.3
D → W	69.5	86.1

(آ) نتایج مقاله

شکل ۸: نتایج روش EasyTL بر روی مجموعه داده Office-Caltech



(د) مقایسه زمان



(ج) مقایسه دقت

Accuracy on Office-Home dataset

Task	EasyTL	EasyTL without alignment
Art - Clipart	52.8 %	51.5 %
Art - Product	72.1 %	68.1 %
Art - RealWorld	75.9 %	74.2 %
Clipart - Art	55.0 %	53.1 %
Clipart - Product	65.9 %	62.9 %
Clipart - RealWorld	67.6 %	65.3 %
Product - Art	54.4 %	52.8 %
Product - Clipart	46.9 %	45.8 %
Product - RealWorld	74.7 %	73.5 %
RealWorld - Art	63.8 %	62.2 %
RealWorld - Product	52.3 %	50.1 %
RealWorld - RealWorld	77.9 %	76.0 %

(ب) نتایج پیاده‌سازی

ID	Task	EasyTL(c)	EasyTL
7	Ar → Cl	51.6	52.8
8	Ar → Pr	68.1	72.1
9	Ar → Rw	74.2	75.9
10	Cl → Ar	53.1	55.0
11	Cl → Pr	62.9	65.9
12	Cl → Rw	65.3	67.6
13	Pr → Ar	52.8	54.4
14	Pr → Cl	45.8	46.9
15	Pr → Rw	73.5	74.7
16	Rw → Ar	62.2	63.8
17	Rw → Cl	50.2	52.3
18	Rw → Pr	76.0	78.0
-	AVG	61.3	63.3

(آ) نتایج مقاله

شکل ۹: نتایج روش EasyTL بر روی مجموعه داده Office-Home

نتایج اجرای روش EasyTL را بر روی ۴ تا مجموعه داده ذکر شده را در شکل های ۶، ۷، ۸ و ۹ می توانید مشاهده کنید.

از مقایسه ی شکل های قسمت (آ) و (ب) در تمامی شکل های ۶، ۷، ۸ و ۹ با نتایج مقاله به این نتیجه می رسیم که دقت هایی که بدست آوردیم کاملاً مطابق با دقت های مقاله است. از نمودارهای قسمت (ج) و (د) در تمامی شکل های ۶، ۷، ۸ و ۹ می توانیم به یک نتیجه گیری کلی برسیم که دقت در حالتی که روش EasyTL را با تراز کردن اجرا می کنیم بیشتر از حالتی است که تراز کردن را در نظر نمی گیریم. اگر چه دقت این دو با یکدیگر تفاوت چندانی ندارد اما از لحاظ زمان و سرعت، روش EasyTL بدون تراز کردن بسیار سریعتر است و نیاز به زمان کمتری دارد. بنابراین در کاربردهایی که دقت برای ما از اهمیت بالایی برخوردار است می توانیم از روش EasyTL با تراز کردن استفاده کنیم و در کاربردهایی که سرعت برای ما مهم است روش EasyTL را بدون تراز کردن استفاده کنیم. به عبارت دیگر در اینجا با یک مصالحه^{۱۹} رو به رو هستیم که با توجه به کاربرد باید تصمیم بگیریم که از کدام روش استفاده کنیم.

مراجع

- [1] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012.
- [2] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *2013 IEEE International Conference on Computer Vision*, pages 2200–2207, 2013.
- [3] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.
- [4] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen. Balanced distribution adaptation for transfer learning. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1129–1134. IEEE, 2017.
- [5] J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang. Easy transfer learning by exploiting intra-domain structures. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1210–1215. IEEE, 2019.
- [6] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.
- [7] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 2020.

trade-off^{۱۹}

پیوست

کد مقاله پیاده‌سازی شده در نوت بوک EasyTL.ipynb قرار دارد. تمامی موارد لازم برای اجرای کد در نوت بوک توضیح داده شده است.

لینک پروژه در گیت هاب.