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 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 بیشماری برای کاهش واگرایی توزیع‌ها پیشنهاد شده است. در این پروژه به بررسی یک روش پارامتری به نام BDA و و یک روش غیر پارامتری EasyTL برای انجام تطبیق دامنه می پردازیم.

۱ مقدمه

لورم ایپسوم متن ساختگی با تولید سادگی نامفهوم از صنعت چاپ و با استفاده از طراحان گرافیک است. چاپگرها و متون بلکه روزنامه و مجله در ستون و سطرآنچنان که لازم است و برای شرایط فعلی تکنولوژی مورد نیاز و کاربردهای متنوع با هدف بهبود ابزارهای کاربردی می باشد.[۱]

۲ ادبیات موضوع

لورم ایپسوم متن ساختگی با تولید سادگی نامفهوم از صنعت چاپ و با استفاده از طراحان گرافیک است. چاپگرها و متون بلکه روزنامه و مجله در ستون و سطرآنچنان که لازم است و برای شرایط فعلی تکنولوژی مورد نیاز و کاربردهای متنوع با هدف بهبود ابزارهای کاربردی می باشد.[۲]

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 $\{x_{s_i},\ y_{s_i}\}_{i=1}^{n}$ و $\{x_{t_j}\}_{j=1}^{m}$ تعریف می‌شود. همچینین 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 ۱) $P_s(x_s)$ و $P_t(x_t)$ ، ۲) $P_s(y_s|x_s)$ و $P_t(y_t|x_t)$ را به حداقل می‌رساند.

۲.۳ تطبیق متوازن دامنه در یادگیری انتقال [۱]

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 مجموعه‌داده بر روی تطبیق دامنه تاثیرگذار است که در روش‌های موجود این موضوع در نظر گرفته نمی‌شود. برای حل این دو مشکل، در [1] 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 فاصله ی بین احتمال های حاشیه ای و شرطی می توانیم دامنه های مبدا و هدف را تطبیق دهیم. این موضوع را میتوانیم به صورت عبارت زیر بیان کنیم:

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

روش BDA به عبارت ۱ یک ضریب توازن به نام μ اضافه می کند که میزان اهمیت توزیع حاشیه ای و شرطی را در مسائل مختلف تنظیم و کنترل می کند. پس عبارت ۱ تبدیل می شود به:

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

برای حل عبارت ۲ از روش MMD برای تخمین توزیع های حاشیه ای و توزیع های شرطی استفاده می کنیم. بنابراین خواهیم داشت:

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

در عبارت ۳ ترم اول فاصله ی توزیع های حاشیه ای بین دامنه ها را نشان می دهد و ترم دو م هم فاصله ی توزیع های شرطی بین دامنه ها را نمایش می دهد. با بهره گیری از تکنیک های ماتریس و جبر خطی عبارت ۳ را می توان به صورت زیر نوشت:

$$\begin{gathered} min \quad tr\left(A^T X \left((1-\mu) M_0 + \mu \sum_{c=1}^{C} M_c\right) X^T A\right) + \lambda \|A\|_F^2 \\ s.t. \quad A^T X H X^T A = I, \quad 0 \leq \mu \leq 1 \end{gathered} \qquad (۴)$$

در عبارت ۴ دو ترم وجود دارد: ۱) تطبیق توزیع های حاشیه ای و شرطی به همراه ضریب توازن μ ۲) ترم مربوط به رگلاریسیون با ضریب رگلارسیون λ. در این عبارت ۲ قید هم داریم. قید اول بیان می کند که داده ی تغییر یافته ویژگی های داخلی اصل داده را حفظ می کند. قید دوم نیز بازه ی ضریب توازن را بیان می کند.

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

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

[۱] Balanced Distribution Adaptation for Transfer Learning

الگوریتم BDA

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 طبقه‌بندf.

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 ازd کوچکترین بردار برای ساخت A استفاده کنید.

۴: طبقه بند f را بر روی $\{A^T X_s,\ y_s\}$آموزش دهید.

۵: برچسب‌های $Dt: \hat{y}_t = f\,(A^T X_t)$ را به‌روز کنید.

۶: ماتریس M_cرا به‌روز کنید. (یا W_c برای *W-BDA*)

۷: طبقه بندf را برگردانید.

شکل ۱

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 می یابیم:

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

عبارت ۶ همان معادله ی 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 می آید که چرا مقدار بهینه μ را بدست نیاوردیم. نکته ی حائز اهمیت در اینجا این است که پارمتر μ یک پارامتر آزاد نیست که بتوانیم آن را تخمین بزنیم بلکه μ با توجه به توزیع داده ها تخمین زده می شود. بنابراین در واقعیت باید با توجه به کاربرد و با استفاده از روش ارزیابی متقابل مقدار بهینه μ را بدست آوریم.

۲.۲.۳ W–BDA

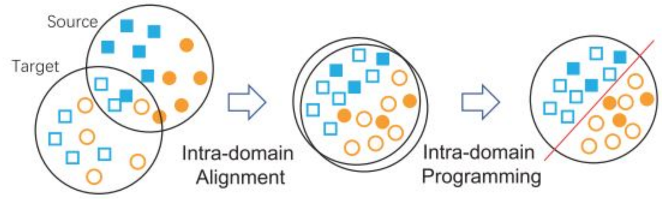
روش W-BDA همانند روش WBDA است با این تفاوت که با در نظر گرفتن نامتوازن بودن کلاس ها در داده ها تخمین دقیق تری را برای احتمال شرطی تعریف می کند. همانطورکه می دانیم در واقعیت ما نمیتوانیم احتمال $P(y|x)$ را به صورت مستقیم تخمین بزنیم به همین دلیل با استفاده از قانون بیز به جای آن احتمال $P(x|y)$ را تخمین میزنیم. روش W-BDA برای تخمین احتمال پیشین داده های موجود در دامنه مبدا و هدف به ترتیب از دو ضریب α_s و α_t استفاده می کند که با استفاده از آن ها تناسب کلاس های هر دامنه را متوازن می کند. اگر بخواهیم این مسئله را به صورت ریاضی بیان کنیم خواهیم داشت:

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

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

$$\begin{aligned}
&min\quad tr\left(A^TX\left((1-\mu)\,M_0+\mu\sum_{c=1}^{C}W_c\right)X^TA\right)+\lambda\|A\|_F^2\\
&\qquad\qquad s.t.\quad A^TXHX^TA=I,\quad 0\le\mu\le1
\end{aligned} \tag{۸}$$

به طور خلاصه الگوریتم دو روش BDA و W-BDA را میتوانید در شکل ۱ مشاهده کنید.



شکل ۲: فرآیند روش $EasyTL$

۳.۳ انتقال یادگیری آسان با استفاده از ساختارهای درون دامنه [۲]

بیشتر روش‌های سنتی و عمیق در انتقال یادگیری، روش‌های پارامتری است که باید طی فرآیندی بسیار گران قیمت و وقت گیر پارامترهای آن را تنظیم کند. ارزیابی متقابل، که متداول‌ترین استراتژی برای انتخاب مدل‌ها و تنظیم پارامترها است، در انتقال یادگیری قابل استفاده نیست زیرا اغلب داده‌های برچسب خورده در دامنه هدف وجود ندارد. اگرچه روش‌های اخیر AutoML می‌تواند به طور خودکار پارامترها را از طریق هرس درخت، boosting 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 مقاله [۲] الگوریتمی پیشنهاد شده است که روشی ساده اما کارآمد را پیشنهاد می‌دهد که علاوه بر این‌که 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 هایپرپارامترها ، 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 ۲). 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 طبقه بند دامنه مبدا را به دامنه هدف منتقل کند. قبل از اینکه به سراغ جزئیات این روش بریم، مفهومی را به نام ماتریس احتمال M معرفی می کنیم. سطر های این ماتریس تعداد کلاس ها و ستون های ان برابر با تعداد نمونه های موجود در مجموعه داده است. هر درایه از این ماتریس نشان می دهد که چقدر احتمال دارد یک نمونه متعلق به کلاس آن سطر باشد. بنابراین جمع احتمال های یک ستون در این ماتریس برابر یک است.

$$\boldsymbol{M} = \begin{array}{cc} & \begin{array}{ccccc} x_1^t & x_2^t & x_3^t & \dots & x_n^t \end{array} \\ \begin{array}{c} c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} & \begin{bmatrix} 0.1 & 0.2 & 0.5 & \cdots & \cdots \\ 0.2 & 0.4 & 0.1 & \cdots & \cdots \\ 0.3 & 0.2 & 0.2 & \cdots & \cdots \\ 0.4 & 0.2 & 0.2 & \cdots & \cdots \end{bmatrix} \end{array}$$

شکل ۳: یک مثال از ماتریس احتمال M

در روش *intra-domain programming* به جای اینکه مستقیما

را بیاموزد، تلاش می کند تا مارتیس احتمال M را بدست آورد. از این رو تابع هزینه به صورت زیر تعریف می شود:

$$\mathbf{J} = \sum_{j}^{n_t} \sum_{c}^{C} D_{cj} M_{cj} \tag{۹}$$

در عبارت ۹ ماتریس D ماتریس فاصله است که درایه ی D_{cj} نشان دهنده ی فاصله ی 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

[۲] Easy Transfer Learning By Exploiting Intra-domain Structures

صورت زیر محاسبه می شود:

$$D_{cj} \ = \ \left|\left|x_t^j \ - \ h_c\right|\right| \tag{۱۰}$$

$$h_c = \frac{1}{\left|\Omega_s^{(c)}\right|} \sum_i^{n_s} x_i^s . \, \mathbb{I}\left(y_i^s = c\right) \tag{۱۱}$$

اکنون برای ان که ماتریس احتمال M را بدست آوریم باید تابع هزینه معادله ۹ را به حداقل برسانید. بدین منظور قید هایی برای این تابع نعریف می شود. اولین قید است است که جمع احتمال هر ستون از ماتریس M برابر با یک باشد. قید دومی که نعریف می شود این است که حداقل باید برای هر کلاس یک نمونه وجود داشته باشد و قید سوم هم همانطور که می دانیم مقدار هر درایه از ماتریس M باید یک عدد در بازه ی صفر تا یک باشد. بنابراین به صورت ریاضی داریم:

$$\begin{aligned}
&min \quad \mathbf{J} = \sum_j^{n_t} \sum_c^{C} D_{cj} M_{cj} \quad s.t \\
&0 \leq \ M_c j \leq 1, \\
&\sum_c^{C} M_{cj} = 1, \ \forall \ j \, \epsilon \, 1, \ \ldots, \ n_t, \\
&\sum_j^{n_t} M_{cj} \geq 1, \ \forall \ c \, \epsilon \, 1, \ \ldots, C
\end{aligned} \tag{۱۲}$$

برای حل عبارت ۱۲ از پکیچ PuLP در پایتون استفاده می کنیم و در نهایت ماتریس M بدست می آید. در گام آخر برچسب نمونه x_j^t به شکل زیر محاسبه می گردد:

$$y_j^t = \underset{r}{argmax} \, \frac{M_{rj}}{\sum_c^C M_{cj}} \qquad for \ \ r \in \{1, \ldots, C\} \tag{۱۳}$$

نکته ی قابل توجه در این روش این است که هیچ پارامتری در این روش وجود ندارد که نباز ب تنظیم کردن داشته باشد. به همین علت است که به این روش، روش غیر پارامتری می گوییم.

۲.۳.۳ intra-domain alignment

۴ پیاده‌سازی و نتایج

همان طور که در [۲] بیان شده است در روش $EasyTL$ باید دو قسمت زیر را پیاده‌سازی کنیم:

۱. *Intra-domain programming*
۲. *Intra-domain alignment*

بخش *Intra-domain programming* شامل ۳ مرحله نیز می‌باشد:

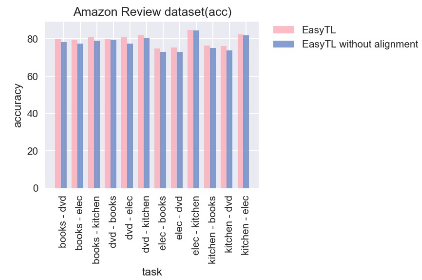
۱. محاسبه‌ی بردار مراکز کلاس‌های دامنه‌ی مبدا h_c: این قسمت در تابع ***get_class_center(Xs, Ys)*** پیاده‌سازی شده است.
۲. محاسبه‌ی ماتریس فاصله D: این قسمت در تابع ***get_distance_matrix(Xt, class_center)*** پیاده‌سازی شده است.
۳. بدست آوردن ماتریس احتمال M با استفاده از معادله‌ی فلان و بدست آوردن برچسب دامنه هدف: این قسمت در تابع ***solve_LP(C, nt, Dcj)*** پیاده‌سازی شده است.

سپس از نتایج این سه تابع استفاده می‌کنیم و آن‌ها را در تابع ***intra_domain_programming(Xs, Ys, Xt, Yt)*** با هم ترکیب می‌کنیم.در بخش *Intra-domain alignment* کافی است تنها معادله‌ی فلان را پیاده‌سازی کنیم. بدین منظور از تابع ***intra_domain_alignment(Xs, Xt)*** استفاده می‌کنیم.

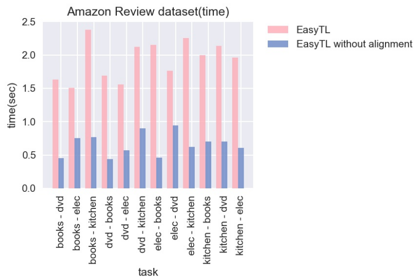
Accuracy on Amazon Review dataset

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 %

(آ)



(ب)



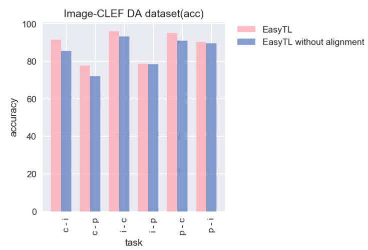
(ج)

شکل ۴: نتایج روش *EasyTL* بر روی مجموعه داده *Amazon Review*

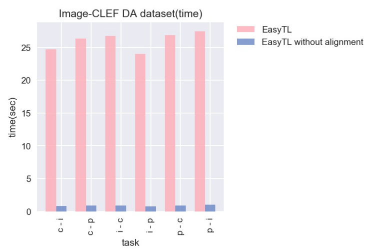
Accuracy on Image-CLEF DA dataset

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 %

(آ)



(ب)



(ج)

شکل ۵: نتایج روش *EasyTL* بر روی مجموعه داده *Image-CLEF DA*

روش *EasyTL* را می‌توانیم به دو صورت اجرا کنیم. در یک حالت فضای دامنه‌ها را بایکدیگر تراز نمی‌کنیم و فقط بخش *Intra-domain programming* را اجرا می‌کنیم و در حالت دیگر ابتدا فضای دامنه‌ها را به یکدیگر تراز می‌کنیم و سپس طبقه‌بند موجود در دامنه مبدا را به دامنه هدف منتقل می‌کنیم. این روش را بر روی ۴ مجموعه‌داده آزمایش می‌کنیم و نتایج را مقایسه می‌کنیم. این ۴ مجموعه داده به شرح زیر هستند:

۱. *Amazon Review* یک مجموعه داده تجزیه و تحلیل احساسات است که شامل بررسی‌های 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.

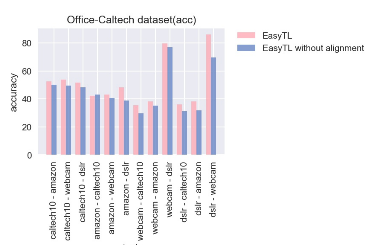
۴. *Office-Home* شامل ۱۵،۵۰۰ تصویر از ۶۵ دسته از ۴ حوزه، Clipart Art ، Product و دنیای واقعی است.

نتایج اجرای روش EasyTL را بر روی ۴ تا مجموعه‌داده ذکر شده را در شکل‌های ۴ ، ۵ ، ۶ و ۷ می‌توانید مشاهده کنید. از مقایسه‌ی شکل‌های قسمت (آ) در تمامی شکل‌های ۴ ، ۵ ، ۶ و ۷ با نتایج مقاله به این نتیجه می‌رسیم که دقت‌هایی که بدست آوردیم کاملا مطابق با دقت‌های مقاله است. از شکل‌های قسمت (ب) و (ج) در تمامی شکل‌های ۴ ، ۵ ، ۶ و ۷ می‌توانیم به

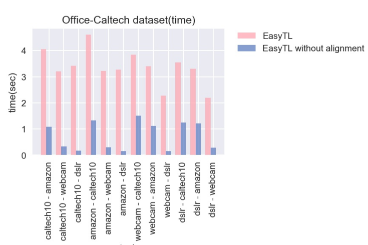
Accuracy on Office-Caltech dataset

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

(آ)



(ب)



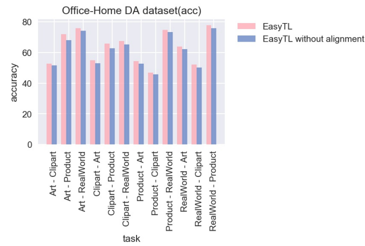
(ج)

شکل ۶: نتایج روش *EasyTL* بر روی مجموعه داده *Office-Caltech*

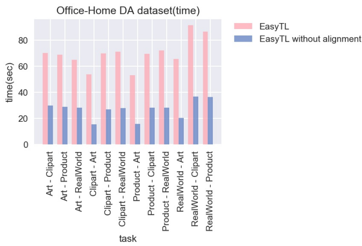
Accuracy on Office-Home DA dataset

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 - Clipart	52.3 %	50.1 %
RealWorld - Product	77.9 %	76.0 %

(آ)



(ب)



(ج)

شکل ۷: نتایج روش *EasyTL* بر روی مجموعه داده *Office-Home*

یک نتیجه‌گیری کلی برسیم و آن این است که دقت در حالتی که روش *EasyTL* را بدون تراز کردن اجرا می‌کنیم با حالتی که تراز کردن را در نظر نمی‌گیریم 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 می‌توانیم 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 مصالحه‌ای رو به رو هستیم که با توجه به کاربرد باید تصمیم بگیریم که از کدام روش استفاده کنیم.

مراجع

[1] 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.

[2] 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.