مقاوم‌سازی روش‌های یادگیری متریک در مقابل داده‌های پرت

داود ذبیح زاده خواجوی[۱]، رضا منصفی[۲]

[۱] استادیار گروه فناوری اطلاعات دانشکده مهندسی، دانشگاه فناوری نوین سبزوار، سبزوار ایران، d.zabihzadeh@gmail.com

[۲] دانشیار گروه مهندسی کامپیوتر دانشکده مهندسی، دانشگاه فردوسی مشهد، ایران monsefi@um.ac.ir

چکیده: در بسیاری از الگوریتم‌های یادگیری ماشین، شناسایی الگو و داده‌کاوی نیازمند آن هستیم که شباهت یا فاصله بین داده‌ها را به روش مناسب اندازه‌گیری کنیم. بعنوان مثال کارایی الگوریتم های خوشه بندی و یا طبقه‌بند k نزدیک‌ترین همسایه به معیار فاصله/شباهت بستگی دارد. معیارهایی عمومی نظیر فاصله اقلیدسی و شباهت کسینوسی که بدون توجه به مفهوم داده‌ها میزان شباهت یا فاصله آن‌ها را مشخص می‌کنند، در بسیاری از کاربردها کارایی مناسبی ندارند. این مساله ضرورت یادگیری متریک را نشان می‌دهد. در یادگیری متریک هدف این است که با توجه به داده‌ها معیار شباهت یا فاصله بهینه به دست آید بطوری که داده‌هایی که ازنظر مفهومی و منطقی شبیه به هم هستند، به یکدیگر نزدیک می‌شوند و داده‌هایی که ازنظر مفهومی و منطقی شبیه نیستند از یکدیگر دور شوند. در این زمینه روش های زیادی ارایه شده است، اما همچنان یکی از چالش‌های مهم و جذاب، کاهش تأثیر داده پرت یا برچسب نویزی می‌باشد. در روش ارائه‌شده، مجموعه داده ورودی همزمان با یادگیری متریک، به دو بخش داده بدون خطا و بخش داده های پرت تقسیم می‌شود و یادگیری معیار فاصله تنها بر روی بخش بدون خطا انجام می شود. آزمایشات انجام شده بر روی‌داده‌های واقعی (در حضور و عدمِ حضور داده پرت و برچسب نویزی) کارایی الگوریتم ارایه شده را تایید می کند و برتری آن را نسبت به روش های همتا در مرزهای دانش در محیط های دارای نویز برچسب نشان می دهد.

کلمات کلیدی: یادگیری متریک، یادگیری متریک مقاوم، یادگیری متریک نزدیک‌ترین همسایه با حاشیه بزرگ، طبقه‌بند k نزدیک‌ترین همسایه، داده‌های پرت، برچسب نویزی

مقدمه

یادگیری متریک[۱] یکی از مسائل بنیادی در برنامه‌های کاربردی می‌باشد. عملکرد بسیاری از الگوریتم‌های یادگیری ماشین ازجمله k نزدیک‌ترین همسایه(kNN)[۲][۱] و خوشه‌بندی[۳] به معیار فاصله‌ای که برای سنجش ارتباط بین داده‌های ورودی است، بستگی دارد [۲].

هدف روش‌های یادگیری متریک، نزدیک کردن داده‌های مشابه به یکدیگر و افزایش فاصله بین داده‌های غیرمشابه می‌باشد، این امر باعث بهبود عملکرد الگوریتم یادگیری مبتنی بر متریک می شود.

اغلب روش های یادگیری متریک از داده های آموزشی بشکل محدویت های زوج و یا سه گانه استفاده می کنند، که بصورت زیر تعریف می شود:

$$S = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ should be similar}\}$$
$$D = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ should be dissimilar}\}$$
$$T = \{(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) \mid \boldsymbol{x}_i \text{ should be more closer to } \boldsymbol{x}_j \text{ than to } \boldsymbol{x}_k\}$$

به‌طورکلی این روش‌های ، مسئله بهینه‌سازی به‌صورت زیر را حل می‌کنند [۳]:

$$\underset{\boldsymbol{M}}{\text{minimize}}\ l(\boldsymbol{M}, C) + \lambda r(\boldsymbol{M}) \qquad (1)$$

در رابطه بالا $\mathbf{M}$ نشان‌دهنده متریک (مجموعه پارامترهای یادگیری) است. C مجموعه محدودیت های مساله را نشان می دهد که اغلب بصورت محدویت های زوج و سه گانه است. l تابع ضرر و r تابع تنظیم‌کننده هست که برای جلوگیری از بیش برازش[۴] تعریف‌شده است. یکی از مهم‌ترین روش‌های یادگیری متریک که تاکنون بیشتر توجه را به خود اختصاص داده است یادگیری فاصله ماهالونوبیس است که به‌صورت زیر تعریف می‌شود:

$$d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j) \qquad (2)$$

در این معیار فاصله $\boldsymbol{M} \in \boldsymbol{S}_+^d$ (مخروط ماتریس‌های متقارن $d \times d$ نیمه معین مثبت) است. این ویژگی $\boldsymbol{M}$ باعث می‌شود که متریک ماهالونوبیس چهار ویژگی نامنفی بودن، انطباق، تقارن و نامساوی مثلث را داشته باشد.

اکثر روش‌های یادگیری متریک فرضی بر وجود داده پرت و برچسب‌های نویزی قائل نشده اند. با این وجود آزمایشات مختلف تایید می کند که وجود این داده ها که باعث افت کارایی قابل ملاحظه آن ها می شود [۴]. از این رو کاهش تأثیر داده‌های پرت و برچسب‌های نویزی یکی از چالش‌های مهم در یادگیری متریک می‌باشد.

در روش ارائه‌شده، مجموعه داده ورودی همزمان با یادگیری متریک، به دو بخش داده بدون خطا و بخش داده های پرت تقسیم می‌شود و یادگیری معیار فاصله بر روی بخش بدون خطا انجام می شود. بطور دقیقتر، داده ورودی ام i $\boldsymbol{x}_i \in \mathbb{R}^d$ به دو بخش $\boldsymbol{x}_i^0$ (داده بدون خطا) و $\boldsymbol{e}_i$ (داده پرت) بصورت $\boldsymbol{x}_i = \boldsymbol{x}_i^0 + \boldsymbol{e}_i \quad i = 1,2,\dots,n$ تجزیه می شود. برای اکثر داده ها، بخش $\boldsymbol{x}_i^0$ مقدار دارد و بخش $\boldsymbol{e}_i$ برابر با $\mathbf{0}$ است، بنابراین ماتریس $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_n]$ تا حد ممکن باید تنک باشد. روش پیشنهادی برخلاف روش های معمول، یادگیری متریک را بر روی ماتریس $\boldsymbol{X}_0 = [\boldsymbol{x}_1^0, \boldsymbol{x}_2^0, \dots, \boldsymbol{x}_n^0]$ که حاوی داده های بدون نویز است، انجام می دهد. بدین ترتیب می توان انتظار داشت که متریک یادگرفته شده در مقابل نویز برچسب و داده های پرت مقاوم خواهد بود. نتیجه آزمایشات بر روی مجموعه داده های مختلف حاوی نویز برچسب، صحت این ادعا را تایید می کند.

ادامه مقاله بصورت زیر سازماندهی شده است. در بخش ۲ کارهای انجام شده در زمینه مقاوم سازی یادگیری متریک مرور می شوند. بخش ۳ به روش پیشنهادی می پردازد. بخش ۴ به آزمایشات انجام شده برای بررسی کارایی روش پیشنهادی و تحلیل نتایج اختصاص

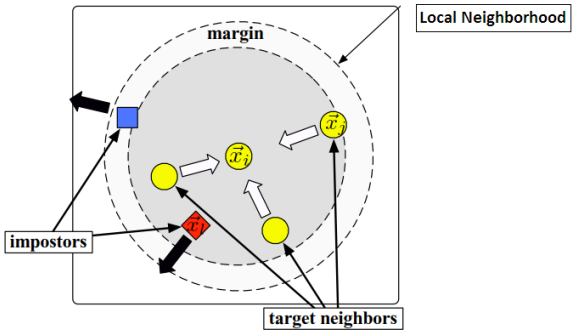
[۱] *Metric Learning*
[۲] *k-nearest neighbor classifier*
[۳] *Clustering*
[۴] *Over Fitting*

یافته است. سرانجام، جمع بندی و نتیجه گیری به همراه چند پیشنهاد برای کارهای آتی در بخش ۵ ارایه شده است.

مروری بر کارهای مرتبط

اولین کار درزمینه یادگیری فاصله ماهالونوبیس، الگوریتم MMC است [۵]. در این روش هدف ماکزیمم کردن فاصله بین داده‌های نامشابه است درحالی‌که مجموع فاصله داده‌های مشابه کمتر از یک مقدار معین باشد.

از کارهای مطرح در زمینه یادگیری متریک به روش LMNN می توان اشاره کرد که برای بهبود طبقه بند kNN طراحی شده است. با توجه به اینکه روش پیشنهادی بر روی این الگوریتم بمنظور مقاوم سازی آن اعمال شده است، این روش را با جزییات بیشتر بررسی می کنیم. برای هر داده $\boldsymbol{x}_i$ ، k نزدیکترین همسایه به آن با برچسب یکسان را همسایه های هدف (Target Neighbors) آن می نامیم. همچنین به داده هایی که در محدوده فاصله $\boldsymbol{x}_i$ با دورترین همسایه هدف آن با در نظر گرفتن یک حاشیه (margin) مناسب قرار می گیرند، داده های بَدل (imposters) گفته می شود. در LMNN، هدف کاهش فاصله هر داده با همسایه‌های هدف آن و درعین‌حال افزایش فاصله آن با داده های بدل است بطوری که داده های بدل از محدوده همسایگی تعریف شده توسط همسایه های هدف خارج شوند. در شکل زیر مفاهیم همسایه های هدف، همسایه های بدل و حاشیه نشان داده شده است.



شکل ۱- مفهوم همسایه های هدف و بدل داده $\boldsymbol{x}_i$ بهمراه همسایگی تعریف شده و حاشیه

به‌صورت کلی تابع هزینه LMNN از دو بخش اصلی بصورت زیر تشکیل می‌شود:

$$\varepsilon(\boldsymbol{M}) = (1-\mu)\varepsilon_{pull}(\boldsymbol{M}) + \mu\varepsilon_{push}(\boldsymbol{M}) \qquad (3)$$

$\epsilon_{pull}(\boldsymbol{M})$ برای نزدیک کردن هر داده به همسایه های هدف آن، بصورت زیر تعریف می شود:

$$\epsilon_{pull}(\boldsymbol{M}) = \sum_{j \rightsquigarrow i} d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 \qquad (4)$$

در رابطه بالا، $j \rightsquigarrow i$ همسایه‌های هدف $\boldsymbol{x}_i$ را نشان می‌دهند. همچنین برای بیرون راندن داده های بَدل از حاشیه، تابع زیان زیر تعریف‌شده است:

$$\epsilon_{pull}(\boldsymbol{M}) = \sum_{i, i \rightsquigarrow j} \sum_{l} (1-y_{il}) \left[1 + d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 - d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l)^2\right]_+ \qquad (5)$$

در رابطه بالا $y_{il} = 1$ اگر $y_i = y_l$ باشد در غیراینصورت برابر 0 است. با ترکیب روابط (4) و (5) به مسئله بهینه‌سازی زیر می‌رسیم.

$$\begin{aligned} \text{minimize } & (1-\mu) \sum_{i, i \rightsquigarrow j} d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 \\ & + \mu \sum_{i, j \rightsquigarrow i} \sum_{l} (1-y_{il}) \xi_{ijl} \\ \textit{subject to} \quad & \\ & 1 + d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 - d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l)^2 \leq \xi_{ijl} \quad \xi_{ijl} \geq 0 \quad \boldsymbol{M} \succcurlyeq 0 \end{aligned} \qquad (6)$$

در این رابطه $0 \leq \mu \leq 1$ امکان مصالحه بین دو جمله را فراهم می‌کند. این الگوریتم در عمل خوب کار می‌کند اما به دلیل عدم وجود تابع تنظیم‌کننده در برخی مسائل مشکل بیش‌برازش دارد.

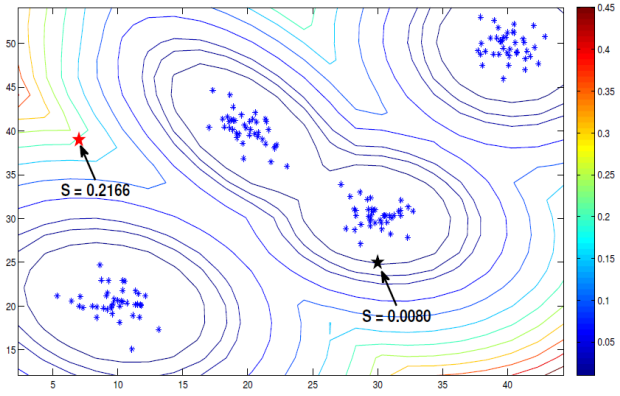
بسیاری از روش‌های یادگیری متریک فرضی بر وجود داده‌های پرت و همچنین برچسب نویزی ندارند. زمانی که داده‌ها شامل ویژگی‌های نامربوط یا نویز برچسب باشند، این روش‌ها با مشکل مواجه می‌شوند. در ادامه این بخش به چند روش که در سال‌های اخیر برای مقاوم سازی متریک در برابر داده های پرت و نویز برچسب انجام شده است، می پردازیم.

در مقاله [۶] برای بهبود عملکرد FCM[۵] نرم دوم اقلیدسی با تابع فاصله نرمالیزه شده $1 - exp(-\beta\|x_i - v_c\|^2)$ جایگزین شده است. متریک فاصله که در روش FCM استفاده می‌شود فقط فاصله اقلیدسی بین هر داده و مرکز خوشه را در نظر می‌گیرد و از پراکندگی فاصله بین داده‌ها در یک خوشه صرف‌نظر می‌کند. این روش تأثیر داده‌های دورافتاده را کاهش می‌دهد ولی از خود داده‌ها و میزان پراکندگی آن‌ها استفاده نمی‌کند و زمانی که خوشه‌های ناهمگن داشته باشیم، نتایج مطلوبی نمی‌دهد.

در برخی از روش ها برای مقاوم سازی متریک بر روی کاهش رتبه[۶] ماتریس $\boldsymbol{M}$ تاکید شده است. از جمله این روش‌ها می توان به [۷] اشاره کرد که از تابع trace بر روی ماتریس $\boldsymbol{M}$ برای تُنُک کردن خروجی و از نُرم ترکیبی یک و دو برای تُنُک کردن ورودی استفاده می‌کند. مقاله [۸] هم که توسعه روش LMNN است، از نُرم ترکیبی یک و دو ماتریس $\boldsymbol{M}$ برای جلوگیری از بیش برازش داده‌ها و همچنین تُنُک کردن آن بهره می برد.

[۹] از KL[۷] و همچنین احتمالات پیشین و پسین برای تشخیص داده های پرت استفاده می‌کند. بدین ترتیب که اگر تفاوت بین توزیع پیشین و پسین بر اساس داده ورودی $\boldsymbol{x}_n$ و مجموعه مدل‌های $\boldsymbol{M} \in \mathcal{M}$ از یک حد آستانه (T_{out}) بیشتر باشدآن داده را داده دورافتاده در نظر می‌گیرد. رابطه (7) و شکل ۹ بیانگر این موضوع می‌باشد. در این رابطه $P(\boldsymbol{M}|\boldsymbol{x}_n)$ نشان‌دهنده احتمال مدل $\boldsymbol{M}$ با توجه به داده ورودی $\boldsymbol{x}_n$ می باشد.

$$\begin{aligned} S(\boldsymbol{x}_n, \mathcal{M}) &= H_m(\boldsymbol{x}_n) = KL\big(P(\boldsymbol{M}|\boldsymbol{x}_n), P(\boldsymbol{M})\big) \\ &= \int_{\boldsymbol{M} \in \mathcal{M}} P(\boldsymbol{M}|\boldsymbol{x}_n)\, log \frac{P(\boldsymbol{M}|\boldsymbol{x}_n)}{P(\boldsymbol{M})} d\boldsymbol{M} \end{aligned} \qquad (7)$$



شکل ۲- داده قرمزرنگ، نشان‌دهنده داده پرت و داده مشکی، نشان‌دهنده داده مناسب می‌باشد که با استفاده از KL تشخیص داده‌شده‌اند[۹].

در مقاله [۱۰] برخلاف روش‌های دیگر که فرض را بر درستی انتخاب مجموعه‌های مشابه S و مجموعه‌های غیرمشابه D می‌گذارند، این احتمال را در نظر می‌گیرد که احتمال نادرست بودن مجموعه‌ها وجود دارد. در این روش به‌نوعی وزن دهی به قیدها انجام‌شده است و درنهایت تبدیل به مسئله محدب شده و با بهینه‌سازی هموارشده نیستروف حل می‌شود.

[۱۱] از ایده استفاده از داده‌های کمکی و بدون برچسب و ترکیب کردن آن با داده‌های آموزش بمنظور جلوگیری از بیش برازش بهره می برد. داده‌های کمکی به داده‌هایی گفته می‌شود که از منبع و تابع توزیع متفاوتی نسبت به هدف به دست می‌آید. در [۱۲] نشان داده شده است که استفاده از این داده‌ها در یادگیری متریک مفید است. روش کار، ساختن

---

۵ *Fuzzy C-Means Clustering*

۶ *rank*

۷ *Kullback–Leibler divergence*

متریک‌های دسته‌جمعی[۸] از داده‌های محدود آموزش و داده‌های کمکی می‌باشد. سپس متریک تجمیع شده[۹] از ترکیب داده‌های آموزش و کمکی بدست می آید.

در [۱۳] فرایند شناسایی مقاوم به دو فاز تشخیص داده های پرت و فاز شناسایی تقسیم شده است. بخش اول از رگرسیون خطی وزن‌دار برای یادگیری متریکی که داده‌های پرت و نویز را تشخیص می‌دهد، استفاده می کند. در بخش دوم با استفاده از متریک یادگرفته شده، مجموعه داده بزرگ به مجموعه کوچک با استفاده از معیار نزدیکترین همسایه فیلتر می شود، سپس یک بازنمایی تنک[۱۰] با استفاده از تکنیک حداقل مربعات نامنفی[۱۱] محاسبه می شود.

در [۱۴] برای یادگیری متریک مقاوم در برابر نویز، تابعی هدفی مشابه [۱۵] ارایه شده است که بجای نُرم ۲ از نُرم ۱ استفاده می کند. با این وجود بعلت نامحدب بودن مساله بهینه سازی، جواب مساله پیشنهاد شده به مقداردهی اولیه متریک بسیار حساس است. همچنین بدلیل اینکه در تابع نُرم ۱ ضرر بصورت خطی رُشد می کند، این روش چندان برای مجموعه داده های حاوی نویز برچسب و داده های پرت مناسب نیست.

در [۱۶] روش معروف NCA[۱۲] در محیط های حاوی نویز برچسب مقاوم شده است. برای این منظور ابتدا تاثیر برچسب نویزی برروی مشتق تابع درست نمایی (likelihood) نشان داده شده و این روش سعی میکند با مدل سازی تابع احتمال درست بودن برچسب، تاثیر داده هایی که با احتمال بالا نویزی هستند را کاهش دهد.

در [۱۷] یادگیری متریک با استفاده از استنتاج بیزی برای مقاوم سازی ارایه شده است. دلیل این مطلب حساسیت بیشتر ماتریس فاصله در روش های تخمین نقطه ای به مثال های آموزشی عنوان شده است. این روش در واقع توسعه بیزی الگوریتم معروف LMNN است و و باعث مقاوم پذیری بهتر آن در برابر نویزبرچسب شده است.

روش پیشنهادی

اگر داده‌های ورودی را به‌صورت $\boldsymbol{X} \in \mathbb{R}^{d*n}$ که n تعداد داده‌های ورودی و d ابعاد آن می‌باشد، در نظر بگیریم. می‌توان داده‌های ورودی $\boldsymbol{X}$ را به دو بخش $\boldsymbol{X_0}, \boldsymbol{E} \in \mathbb{R}^{d*n}$ طبق رابطه (9) تقسیم کرد که $\boldsymbol{X_0}$ را بخش داده‌های بدون خطا و $\boldsymbol{E}$ را بخش مقادیر پرت یا خطا در نظر گرفت.

$$\boldsymbol{X} = \boldsymbol{X_0} + \boldsymbol{E} \tag{8}$$

بدلیل این‌که بیشتر داده ها حاوی نویز برچسب نیستند، ماتریس E باید تُنُک باشد. برای تنک سازی این ماتریس می توان نُرم یک آن را مینیمم کرد که بصورت زیر تعریف می شود:

$$\|\boldsymbol{E}\|_{1,1} = \|vec(\boldsymbol{E})\|_1 = \sum_{i=1}^{d}\sum_{j=1}^{n}|e_{ij}| \tag{9}$$

با توجه به رابطه (1) مسئله بهینه‌سازی متریک معمولاً از دو بخش تابع ضرر که شامل ماتریس متریک $\boldsymbol{M}$ و مجموعه محدودیت‌های C است، و جمله تنظیم کننده برای جلوگیری از بیش برازش تشکیل‌شده است. در روش پیشنهادی برای آموزش ماتریس متریک $\boldsymbol{M}$ داده های خالص $\boldsymbol{X_0} \in \mathbb{R}^{d*n}$ جایگزین داده‌های ورودی $\boldsymbol{X} \in \mathbb{R}^{d*n}$ می شود. همچنین تنک سازی ماتریس $\mathbf{E}$ را بعنوان جمله تنظیم کننده در نظر میگیریم. به این ترتیب به مساله بهینه سازی زیر می‌رسیم:

$$\min_{\boldsymbol{M}\succcurlyeq\boldsymbol{0},\boldsymbol{X_0}} \; l(\boldsymbol{M}, C, \boldsymbol{X_0}) + \lambda r(\boldsymbol{M}, \boldsymbol{E}) \tag{10}$$

به دلیل آنکه می‌توان ماتریس مقادیر پرت را طبق رابطه (8) به‌صورت $\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{X_0}$ نوشت، بنابراین با محاسبه مقدار ماتریس $\boldsymbol{X_0}$، مقدار ماتریس $\boldsymbol{E}$ محاسبه می‌شود و این متغیر را به‌عنوان مجهول در نظر نمی‌گیریم. ماتریس متریک $\boldsymbol{M}$ نیمه معین مثبت متقارن است و می‌توان آن را به‌صورت $\boldsymbol{M} = \boldsymbol{W}\boldsymbol{W}^T$ تجزیه کرد که $\boldsymbol{W} \in \mathbb{R}^{\boldsymbol{d}\times\boldsymbol{r}}$ و r رتبه ماتریس $\boldsymbol{M}$ را نشان می‌دهد. همچنین طبق رابطه (2) و با استفاده از انتقال خطی $\boldsymbol{x}_i' = \boldsymbol{W}^T\boldsymbol{x}_i$، داده‌ها در فضای جدید به دست می‌آیند. رابطه (10) بیانگر این موضوع است که برای به دست آوردن ماتریس متریک $\boldsymbol{M}$ و درنهایت ماتریس انتقال خطی $\boldsymbol{W}$ از مقادیر بدون خطا $\boldsymbol{X_0}$ استفاده می شود. نتایج آزمایشات تایید می کند که این کار باعث بهبود قابل ملاحظه در کارایی متریک یادگرفته شده می شود. مسئله پیشنهادی نسبت به متغیرهای $\boldsymbol{X_0}$ و $\boldsymbol{M}$ نامحدب می‌باشد و برای حل مسئله ارائه‌شده از چارچوب BCD[۱۳] معرفی‌شده استفاده‌شده است. این روش متغیرها را به ۲ زیرگروه مجزا $\boldsymbol{\theta}\chi = (\boldsymbol{M}, \boldsymbol{X_0})$ تقسیم می‌کند بطوری‌که هر زیر مسئله بطور مجزا محدب است. مساله اصلی به‌صورت تکراری و با در نظر گرفتن تنها متغیرهای هر گروه در یک‌زمان، حل می شود.

اعمال روش پیشنهادی بر روی الگوریتم LMNN

با اعمال روش پیشنهادی بر روی الگوریتم LMNN به مساله بهینه سازی زیر می‌رسیم:

$$\begin{aligned}\text{minimize } & (1-\mu)\sum_{i,i\rightsquigarrow j} d_{\boldsymbol{M}}(\boldsymbol{x}_{0i}, \boldsymbol{x}_{0j})^2 \\ & +\mu\sum_{i,j\rightsquigarrow l}\sum_{l}(1-y_{il})\xi_{ijl} + \lambda_e\|\boldsymbol{E}\|_{1,1} \\ subject\ to & \\ & 1 + d_{\boldsymbol{M}}(\boldsymbol{x}_{0i}, \boldsymbol{x}_{0j})^2 - d_{\boldsymbol{M}}(\boldsymbol{x}_{0i}, \boldsymbol{x}_{0l})^2 \le \xi_{ijl} \\ & \boldsymbol{X} = \boldsymbol{X_0} + \boldsymbol{E}, \xi_{ijl} \ge 0 \quad \boldsymbol{M} \succcurlyeq 0\end{aligned} \tag{11}$$

همانطورکه مشاهده می شود در روش پیشنهادی برای یادگیری متریک از بخش بدون خطای داده ($\boldsymbol{X_0}$) بهره می برد. همچنین ابر پارامتر λ_e برای تنظیم میزان اهمیت تُنُک بودن ماتریس $\mathbf{E}$ به مساله اضافه شده است. برای حل رابطه نامحدب (11) با بهره گرفتن از روش BCD می‌توان متغیرها $\boldsymbol{\theta}\chi = (\boldsymbol{M}, \boldsymbol{X_0})$ را به دو زیر مسئله محدب تقسیم کرد و به‌صورت تکراری حل نمود. اثبات‌شده است که در هر بروزرسانی، تابع هزینه هیچ‌وقت افزایش نمی‌یابد [۱۸]. تابع هزینه (11) به‌صورت بلاکی محدب می‌باشد درنتیجه در هر تکرار تمامی زیر مسائل محدب می‌باشند [۱۹]. مقادیر اولیه μ و λ_e بااعتبار سنجی متقاطع و $\boldsymbol{X_0} = \boldsymbol{X}$ به دست می‌آید. گام‌های زیر به ترتیب برای حل تابع (11) انجام می‌پذیرد.

۱- در مرحله s، ماتریس نیمه مثبت معین $\boldsymbol{M}_s$ با استفاده از $\boldsymbol{X}_{0,s-1}$ محاسبه می شود.

۲- محاسبه ماتریس داده‌های بدون خطا $\boldsymbol{X}_{0,s}$ با ثابت در نظر گرفتن ماتریس $\boldsymbol{M}_s$.

۳- تکرار مراحل ۱ و ۲ تا زمانی که مساله به مقدار بهینه همگرا شود یا به تعداد تکرار موردنظر برسد.

یادگیری ماتریس متریک $\boldsymbol{M}_s$

در این مرحله ماتریس $\boldsymbol{X}_{0,s}$ ثابت فرض می شود و تنها متغیر $\mathbf{M}$ را لحاظ می کنیم. مشابه الگوریتم LMNN می توان این مرحله را با استفاده از روش زیر گردایان نزولی بشکل زیر حل کرد:

$$\boldsymbol{G}_t = \frac{\partial\varepsilon}{\boldsymbol{M}_t} = (1-\mu)\sum_{i,\,j\rightsquigarrow i}\boldsymbol{C}_{ij} + \mu\sum_{(i,j,l)\in\mathcal{N}_t}(\boldsymbol{C}_{ij} - \boldsymbol{C}_{il}) \tag{12}$$

$$\boldsymbol{M}_{t+1} = \boldsymbol{M}_t - \tau\boldsymbol{G}_t \tag{13}$$

در رابطه بالا $\boldsymbol{C}_{ij} = (\boldsymbol{x}_{0i,s-1} - \boldsymbol{x}_{0j,t-1})(\boldsymbol{x}_{0i,s-1} - \boldsymbol{x}_{0j,s-1})^T$ است و τ نشان دهنده نرخ یادگیری یا گام حرکت است.

ماتریس متریک $\boldsymbol{M}_t$ باید نیمه مثبت معین باقی بماند. برای محقق شدن این امر، تجزیه مقادیر ویژه $\boldsymbol{M}_t = \boldsymbol{V}\Delta\boldsymbol{V}^T$ را به دست می‌آوریم. $\mathbf{V}$ ماتریس بردارهای ویژه می‌باشد و $\boldsymbol{\Delta}$ ماتریس قطری مقادیر ویژه که $\boldsymbol{\Delta} = \boldsymbol{\Delta}^{-}+\boldsymbol{\Delta}^{+}$ می‌باشد. با صفر در نظر گرفتن مقادیر ویژه منفی می‌توان تصویر $\boldsymbol{M}_t$ را بر روی منحنی نیمه مثبت معین به‌صورت زیر به دست آورد:

$$\rho_s(\boldsymbol{M}_t) = V\Delta^{+}V^T \tag{14}$$

یادگیری داده‌های بدون خطا $\boldsymbol{X}_{0,s}$

فرض کنید در تکرار s قرار داریم. در مرحله قبل ماتریس متریک $\boldsymbol{M}_s$ محاسبه شد و با ثابت در نظر گرفتن آن می‌توان ماتریس داده‌های بدون خطا $\boldsymbol{X}_{0,s}$ را به دست آورد. ماتریس

---

[۸] *ensemble*

[۹] aggregated

[۱۰] *Sparse Representation*

[۱۱] *Non-negative Least Squares*

[۱۲] *Neighborhood Components Analysis*

[۱۳] *Block-Coordinate Descent*

$\boldsymbol{M}_s$ ثابت است، بنابراین قید مثبت بودن آن از رابطه (11) حذف می‌گردد و می توان تابع هزینه را بشکل ساده تر زیر نوشت:

$$\epsilon(\boldsymbol{X}_0) = L(\boldsymbol{X}_{0,s}) + \lambda_e \|\boldsymbol{X} - \boldsymbol{X}_0\|_1 \tag{15}$$

$$\begin{aligned} \boldsymbol{L}(X_0) = (1-\mu) \sum_{i,\, j \rightsquigarrow i} d_{\boldsymbol{M}}(x_{0i}, x_{0j})^2 \\ + \mu \sum_{i,\, j \rightsquigarrow i} \sum_{l} (1 - y_{il}) \left[1 + d_{\boldsymbol{M}}(\boldsymbol{x}_{0i}, \boldsymbol{x}_{0j})^2 \right. \\ \left. - d_{\boldsymbol{M}}(\boldsymbol{x}_{0i}, \boldsymbol{x}_{0l})^2 \right]_+ \end{aligned} \tag{16}$$

اگر در هر مرحله برای محاسبه بردار گرادیان، مجموعه سه‌تایی $\mathcal{N}_k = (i, j, l)$ که شامل قیدهای فعال می‌باشند، را در نظر بگیریم آن‌گاه $[Z]_+ = [Z]$ و در نتیجه بردار گرادیان $\nabla_i L(\boldsymbol{x}_{0i})$ را می‌توان با مشتق‌گیری بصورت زیر بدست آورد:

$$\begin{aligned} \nabla_i L(\boldsymbol{x}_{0i}) = (1-\mu) \left(\sum_{j \rightsquigarrow i} 2\boldsymbol{M}\boldsymbol{x}_{0i} - 2\boldsymbol{M}\boldsymbol{x}_{0j} \right. \\ \left. + \sum_{i \rightsquigarrow j} -2\boldsymbol{M}\boldsymbol{x}_{0j} + 2\boldsymbol{M}\boldsymbol{x}_{0i} \right) \\ + \mu \left(\sum_{(j \rightsquigarrow i) \in N} 2\boldsymbol{M}\boldsymbol{x}_{0i} - 2\boldsymbol{M}\boldsymbol{x}_{0j} \right. \\ + \sum_{(i \rightsquigarrow j) \in N} -2\boldsymbol{M}\boldsymbol{x}_{0j} + 2\boldsymbol{M}\boldsymbol{x}_{0i} \\ + \sum_{(i \nrightarrow l) \in N} 2\boldsymbol{M}\boldsymbol{x}_{0i} - 2\boldsymbol{M}\boldsymbol{x}_{0l} \\ \left. + \sum_{(l \nrightarrow i) \in N} -2\boldsymbol{M}\boldsymbol{x}_{0l} + 2\boldsymbol{M}\boldsymbol{x}_{0i} \right) \end{aligned} \tag{17}$$

در رابطه بالا، $i \nrightarrow l$ بیان‌کننده بدل بودن داده l برای i می‌باشد. همچنین بردار گرادیان $\nabla_{\mathrm{i}} \epsilon(\mathrm{x}_{0\mathrm{i}})$ را می‌توان از رابطه زیر محاسبه کرد:

$$\frac{\partial \epsilon(\boldsymbol{x}_{0i,j})}{\partial \boldsymbol{x}_{0i,j}} = \begin{cases} \dfrac{\partial \boldsymbol{L}(x_{0i,j})}{\partial x_{0i,j}} + \lambda_e \mathrm{sign}(x_{i,j} - x_{0i,j}) & , |x_{i,j} - x_{0i,j}| > 0 \\ \dfrac{\partial \boldsymbol{L}(x_{0i,j})}{\partial x_{0i,j}} + \lambda_e & , x_{i,j} - x_{0i,j} = 0, \dfrac{\partial L(x_{0i,j})}{\partial x_{0i,j}} < -\lambda_e \\ \dfrac{\partial \boldsymbol{L}(x_{0i,j})}{\partial x_{0i,j}} - \lambda_e & , x_{i,j} - x_{0i,j} = 0, \dfrac{\partial L(x_{0i,j})}{\partial x_{0i,j}} > \lambda_e \\ 0 & x_{i,j} - x_{0i,j} = 0, -\lambda_e \le \dfrac{\partial L(x_{0i,j})}{\partial x_{0i,j}} \le \lambda_e \end{cases} \tag{18}$$

در حالت اولیه $\mathrm{x}_{\mathrm{i,j}} = \mathrm{x}_{0\mathrm{i,j}}$ است و در رابطه (18) فقط $\mathrm{x}_{0\mathrm{i,j}}$ که $\left|\frac{\partial \mathrm{L}(\mathrm{x}_{0\mathrm{i,j}})}{\partial \mathrm{x}_{0\mathrm{i,j}}}\right| > \lambda_{\mathrm{e}}$ دارند، می‌تواند مقادیر غیر صفر داشته باشد. بنابراین ماتریس خطا ($\boldsymbol{E}$) تنک می‌باشد. درنهایت با توجه به گرادیان محاسبه‌شده در رابطه، ماتریس $\mathbf{X}_{0,\mathrm{t}}$ با روش نزول گردایانی از رابطه زیر به دست می‌آید:

$$\boldsymbol{X}_{0,t} = \left(\boldsymbol{X}_{0,t-1} - \alpha \boldsymbol{\nabla} \epsilon(\boldsymbol{X}_{0,t})\right) \tag{19}$$

که در رابطه بالا α نرخ یادگیری را نشان می دهد. در الگوریتم ۱ مراحل روش پیشنهادی بطور خلاصه نشان داده شده است.

الگوریتم ۱- شبه کد روش پیشنهادی

ورودی:
مجموعه آموزشی شامل n نمونه $\mathbf{x}_{\mathrm{i}} \epsilon \mathbb{R}^{\mathrm{d}}$ و برچسب‌های آن
در حالت اولیه داریم : $\mathbf{x}_{0\mathrm{i}} = \mathbf{x}_{\mathrm{i}} \quad , \forall\, \mathrm{i} = 1, \ldots, \mathrm{n}$

خروجی:
ماتریس نیمه معین مثبت $\boldsymbol{M}$.

الگوریتم :

۱- انتخاب λ_{e} و μ مناسب.
۲- محاسبه $\mathbf{G}_{\mathrm{t}}$ و با استفاده از رابطه (12)
۳- محاسبه $\mathbf{M}_{\mathrm{t}}$ از رابطه (14) و ثابت بودن ماتریس $\mathbf{X}_0$ در تمامی تکرارها
۴- تکرار مراحل ۲ و ۳ تا زمانی که به کمینه سراسری نسبت به $\mathbf{M}_{\mathrm{s}}$ همگرا شود یا $\mathbf{t}$ به حداکثر تکرار برسد.
۵- محاسبه $\frac{\partial \epsilon(\mathrm{x}_{0\mathrm{i,j}})}{\partial \mathrm{x}_{0\mathrm{i,j}}}$ از رابطه (18) و $\mathbf{X}_{0,\mathrm{t}}$ از رابطه (19) با ثابت بودن ماتریس $\boldsymbol{M}_s$
۶- تکرار مرحله ۵ تا زمانی که به کمینه سراسری نسبت به $\mathbf{X}_{\mathrm{s}}$ همگرا شود یا t به حداکثر تکرار برسد.
۷- تکرار مراحل ۲ تا ۶ و بررسی شرایط همگرایی و برگشت به گام ۲ در صورت همگرا نشدن

آزمایش‌ها و تحلیل نتایج

جهت ارزیابی کارایی الگوریتم ارائه‌شده و مقایسه آن با سایر روش‌ها از ۸ پایگاه داده با اندازه و پیچیدگی های مختلف استفاده می‌کنیم. مشخصات این مجموعه داده ها در جدول گزارش شده است.

در پایگاه داده‌هایی که تعداد ویژگی‌ها زیاد می‌باشد، با استفاده از PCA ابعاد داده‌ها را قبل از اعمال الگوریتم کاهش می‌دهیم. نتایج از میانگین چندین بار اجرا با تخصیص ۷۰٪ از داده‌ها به داده‌های آموزش و ۳۰٪ داده‌های تست، به دست می‌آید. تعداد همسایه‌های هدف برای یادگیری الگوریتم پیشنهادی و LMNN در تمامی گزارش‌ها $\mathrm{k}=3$ و پارامتر $\mu = 0.5$ می‌باشد. همسایه‌های هدف در فضای ورودی (در صورت نیاز پس از کاهش ابعاد با PCA) بر اساس فاصله اقلیدسی به دست می‌آید. مقدار مناسب پارامتر λ_{e} برای هر پایگاه داده با استفاده از روش اعتبار سنجی متقاطع 5-fold حاصل می‌شود.

طبقه بند kNN از فاصله اقلیدسی و LMNN از متریک حاصل‌شده و روش ارائه‌شده از داده‌های بدون خطا X_0 داده‌های آموزش و ماتریس متریک برای مقایسه استفاده می‌کند. همچنین از روش RNCA برای مقایسه با روش ارائه‌شده استفاده‌شده است.

نتایج گزارش‌شده در این بخش بر اساس درصد خطای طبقه‌بندی است. طبقه بند استفاده‌شده، kNN با $\mathrm{k} = 1$ می‌باشد. نویز برچسب با نرخ‌های متفاوت و همچنین داده پرت برای ارزیابی روش‌های مورد مقایسه، استفاده‌شده است.

جدول -۱- ویژگی‌های مجموعه داده‌های مورداستفاده در آزمایش‌ها.

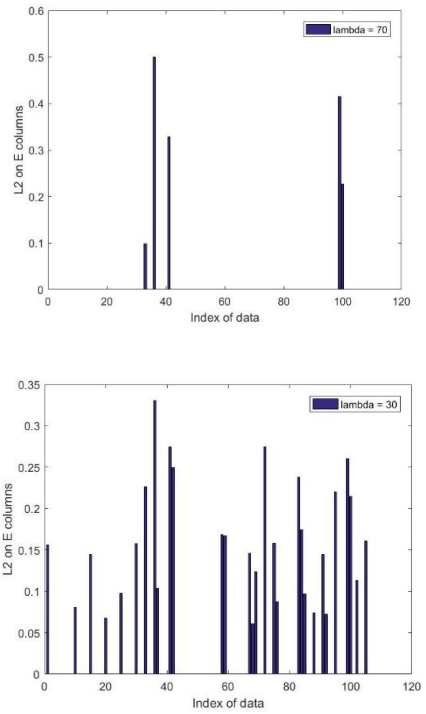
مجموعه داده	تعداد داده‌ها (آزمون / آموزش)	تعداد ویژگی‌ها	تعداد ویژگی‌های کاهش یافته	تعداد کلاس‌ها
ionosphere	۲۴۵/۱۰۷	۳۴	۳۴	۲
Heart	۱۸۹/۸۲	۱۳	۱۳	۲
balance	۴۳۷/۱۸۹	۴	۴	۳
wine	۱۰۶/۴۶	۱۳	۱۳	۳
iris	۱۰۵/۴۵	۴	۴	۳
Olivetti face	۲۸۰/۱۲۰	۲۰۰	۲۰۰	۴۰
MNIST	۲۱۰۰/۹۰۰	۷۸۴	۱۶۴	۱۰

نتایج آزمایش‌ها

تأثیر پارامتر λ_e بر تنک بودن ماتریس خطا

در حالت اولیه $\boldsymbol{X} = \boldsymbol{X}_0$ می‌باشد و طبق رابطه (18) مقادیر گرادیانی که از قدر مطلق λ_e بیشتر باشد، می‌تواند در جهت خلاف بردار گرادیان پیش برود. به عبارتی با تنظیم دقیق این پارامتر، بخش‌هایی از داده x_i که شامل مقادیر پرت است به e_i منتقل می‌شود.

شکل ۱۹ تأثیر اندازه پارامتر λ_e بر انتخاب بخش‌هایی که شامل مقادیر پرت می‌باشند در مجموعه داده Olivetti face را نشان می‌دهد. در این شکل از نرم l_2 روی هر ستون $\boldsymbol{e}_i \quad \forall\ i = 1,2,..,n$ برای نمایش آن استفاده شده است. نتایج این آزمایش تایید می کند اندازه پارامتر λ_e در تعیین تنک بودن ماتریس E نقش مهمی دارد. هر چه λ_e مقدار بیشتری داشته باشد، میزان تنک‌سازی بیشتر خواهد بود. اگر λ_e به صورتی اختیار شود که ماتریس $e_{ij} = 0\ \forall\ i = 1,..,n\ ,j = 1,..,d$ یا به عبارتی تمامی مقادیر آن برابر صفر باشد، مدل پیشنهادی مشابه روش LMNN می‌باشد. با انتخاب λ_e مناسب می‌توان از بیش برازش بر روی داده‌های آموزش نیز جلوگیری کرد.



شکل۳- تأثیر اندازه پارامتر λ_e در تنک بودن ماتریس خطای $\boldsymbol{E}$.

آزمایش بر روی مجموعه داده‌ها

ارزیابی روش‌ها بر روی داده های آزمون، بر مدل ارائه‌شده، k نزدیک‌ترین همسایه انجام‌شده است. درروش پیشنهادی نقطه شروع $\boldsymbol{X}_0 = \boldsymbol{X}$ در نظر می‌گیریم. در تولید داده پرت از dd_tools toolbox [۲۰] که از توصیف داده‌های هر کلاس و استفاده از شعاع و مرکز هر کلاس و اضافه کردن مقدار تصادفی در هر بعد داده پرت بهره می‌برد، استفاده می‌کنیم. از نرخ داده پرت ۱۰٪ در تمامی مجموعه داده‌ها جهت مقایسه روش‌ها استفاده می‌کنیم. نویز برچسب با نرخ ۱۰٪ ، ۲۰٪ و ۳۰٪ بر روی داده‌های آموزش استفاده‌شده است. هدف اصلی در آزمایش‌ها، بررسی مقاوم بودن و کارایی روش معرفی‌شده در حضور داده پرت و برچسب نویزی است.

در جدول های زیر نتایج آزمایشات بر روی مجموعه داده های مختلف نشان داده شده است. همانطورکه مشاهده می شود روش ارائه‌شده در اکثر مجموعه‌های داده، کارایی بهتری از 1-NN و LMNN دارد. در مقایسه با روش RNCA، در نویز برچسب با نرخ کمتر روش ارائه‌شده کارایی بهتری دارد. روش معرفی‌شده، برخلاف تابع هزینه روش LMNN که در برابر داده با برچسب‌های نویزی کارایی مناسبی ندارد، با اعمال جمله تنظیم کننده مناسب در تابع هزینه نسبت به حضور برچسب نویزی مقاوم می‌باشد.

جدول ۲-بررسی و مقایسه روش ارائه‌شده با سایر روش‌ها در حضور نویز برچسب با نرخ‌های متفاوت در مجموعه داده‌های UCI.

مجموعه داده	1-NN	LMNN	RNCA	Proposed method	نویز برچسب (٪)
ionosphere	۱۷,۹۸	۱۶,۸۹	۱۶,۰۰	**۱۳,۸۰**	۱۰
	۱۹,۵۸	۱۹,۰۱	۱۷,۵۰	**۱۴,۹۵**	۲۰
	۲۲,۸۶	۲۲,۴۳	۱۹,۰۰	**۱۸,۹۸**	۳۰
Heart	۲۳,۰۶	۲۰,۳۸	**۱۹,۹**	۲۰,۲۱	۱۰
	۲۵,۰۷	۲۴,۶۷	۲۴,۰۰	**۲۰,۷۸**	۲۰
	۲۸,۹۸	۲۶,۴۱	۲۸,۵۰	**۲۱,۳۸**	۳۰
balance	۲۸,۳۵	۱۷,۴۳	**۱۶,۰۰**	۱۶,۳۷	۱۰
	۳۲,۵۵	۲۷,۵۵	**۲۲,۵**	۲۴,۹۶	۲۰
	۳۶,۶۷	۳۵,۴۵	**۳۲,۰۰**	۳۴,۶۷	۳۰
wine	۲۴,۹۷	۵,۸۳	۲,۱۰	**۲,۰۴**	۱۰
	۲۹,۰۰	۸,۷۸	**۴,۵۰**	۶,۵۵	۲۰
	۳۸,۰۱	۱۴,۸۷	**۷,۵۰**	۹,۸۰	۳۰
iris	۸,۴۵	۸,۸۶	۳,۵۰	**۳,۴۰**	۱۰
	۱۱,۸۷	۱۲,۲۲	۷,۹۰	**۷,۸۸**	۲۰
	۱۸,۵۶	۱۸,۴۰	**۱۴,۰۰**	۱۴,۵۱	۳۰

جدول ۳-مقایسه روش ارائه‌شده با سایر روش‌ها در حضور نویز برچسب با نرخ‌های متفاوت در مجموعه داده‌های تصویر.

مجموعه داده‌ها	1-NN	LMNN	Proposed method	نویز برچسب (٪)
Olivetti face	۱۳,۲۴	۱۲,۹۸	**۸,۰۲**	۱۰
	۱۸,۸۶	۲۵,۳۶	**۱۰,۲۵**	۲۰
	۲۲,۵۰	۳۸,۴۵	**۱۵,۹۵**	۳۰
MNIST	۲۰,۲۲	۱۹,۵۷	**۱۶,۹۶**	۱۰
	۲۲,۶۷	۲۱,۸۲	**۱۷,۲۳**	۲۰
	۳۳,۷۸	۲۶,۱۲	**۲۱,۲۸**	۳۰

در آزمایش بعدی میزان مقاوم بودن روش پیشنهادی در برابر داده پرت بررسی شده است. در جدول نتایج این آزمایش گزارش شده است. همانطور که مشاهده می شود مدل ارائه‌شده دارای تخمین خطای کمتری است و در مقایسه با روش LMNN و 1-NN کارایی بیشتری دارد.

جدول ۴- بررسی و مقایسه روش ارائه‌شده با مدل LMNN و طبقه بند K نزدیک‌ترین همسایه، شامل ۱۰٪ داده پرت.

مجموعه داده	1-NN	LMNN	Proposed method
ionosphere	۱۷٫۳۵	۱۵٫۴۵	**۱۳٫۴۵**
Heart	۲۷٫۰۰	۲۴٫۰۱	**۲۲٫۴۸**
balance	۲۹٫۹۸	۱۵٫۶۶	۱۵٫۶۶
wine	۲۶٫۱۳	۶٫۹۰	**۶٫۱۰**
iris	۵٫۸۷	۶٫۳۴	**۴٫۹۷**
Olivetti face	۶٫۸۲	۵٫۳۷	**۴٫۳۷**
MNIST	۳۳٫۰۰	۲۲٫۵۶	**۲۱٫۶۷**

نتیجه‌گیری، پیشنهاد و توصیه‌های آتی

اکثر روش‌های یادگیری متریک عملکرد مناسبی در مقابل داده‌های شامل مقادیر پرت و برچسب نویز ندارند. ازجمله این روش‌ها می‌توان به روش محبوب LMNN اشاره کرد. برای حل این مشکل، روش ارائه‌شده، متریک موردنظر را با قسمتی از داده‌ها که بدون تأثیرات ذکرشده است، آموزش می‌دهد. این روش با استفاده از نرم یک خطا داده های پَرت را شناسایی می کند. برای حل مسئله بهینه‌سازی، از روش BCD استفاده‌شده است. کارایی روش معرفی‌شده، بر روی‌داده‌های UCI و تصویر در شرایط مختلف موردبررسی قرار گرفت. بر اساس نتایج آزمایش‌ها، روش ارائه‌شده در حضور داده‌های پرت و برچسب نویزی نسبت به روش LMNN برتری قابل‌توجهی دارد. به‌طورکلی نتایج آزمایش‌ها روی‌داده‌های مختلف، برتری روش پیشنهادی را به خصوص در حضور برچسب نویزی نشان می‌دهد.

برخی توصیه‌های برای کارهای آینده در این زمینه عبارتند از:

۱- اعمال مدل ارائه‌شده بر روی سایر روش‌های یادگیری متریک.

۲- تبدیل مدل ارائه‌شده به‌صورت آنلاین برای اجرا بر روی پایگاه داده‌های بزرگ‌تر

مراجع:

[۱] T. Cover, and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21-27, 1967.

[۲] F. Wang, and J. Sun, “Survey on distance metric learning and dimensionality reduction in data mining,” *Data mining and knowledge discovery*, vol. 29, no. 2, pp. 534-564, 2015.

[۳] A. Bellet, A. Habrard, and M. Sebban, “A survey on metric learning for feature vectors and structured data,” *arXiv preprint arXiv:1306.6709*, 2013.

[۴] S. Theodoridis, A. Pikrakis, K .Koutroumbas, and D. Cavouras, *Introduction to pattern recognition: a matlab approach*: Academic Press, 2010.

[۵] E. P. Xing, M. I. Jordan, S. J. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information." pp. 521.۵۲۸-

[۶] D.-M. Tsai, and C.-C. Lin, “Fuzzy C-means based clustering for linearly and nonlinearly separable data,” *Pattern recognition*, vol. 44, no. 8, pp. 1750-1760, 2011.

[۷] D. Lim, G. Lanckriet, and B. McFee, "Robust structural metric learning." pp. 6.۶۲۳-۱۵

[۸] M. P. Kumar, P. H. Torr, and A. Zisserman, "An invariant large margin nearest neighbour classifier." pp. 1-8.

[۹] E. Hasanbelliu, K. Kampa, J. C. Principe, and J. T. Cobb, "Online learning using a Bayesian surprise metric." pp. 1-8.

[۱۰] K. Huang, R. Jin, Z. Xu, and C.-L. Liu, “Robust metric learning by smooth optimization,” *arXiv preprint arXiv:1203.3461*, 2012.

[۱۱] Z.-J. Zha, T. Mei, M. Wang, Z. Wang, and X.-S. Hua, "Robust distance metric learning with auxiliary knowledge".

[۱۲] S. C. Hoi ,W. Liu, and S.-F. Chang, “Semi-supervised distance metric learning for collaborative image retrieval and clustering,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 6, no. 3, pp. 18, 2010.

[۱۳] R. He, B. Hu, W.-S .Zheng, and Y. Guo, "Two-stage sparse representation for robust recognition on large-scale database".

[۱۴] H. Wang, F. Nie, and H. Huang, "Robust Distance Metric Learning via Simultaneous L1-Norm Minimization and Maximization." pp. 1836-1844.

[۱۵] S. Xiang ,F. Nie, and C. Zhang, “Learning a Mahalanobis distance metric for data clustering and classification,” *Pattern Recogn.*, vol. 41, no. 12, pp. 3600-3612, 2008.

[۱۶] D. Wang, and X. Tan, "Robust distance metric learning in the presence of label noise".

[۱۷] D. Wang, and X. Tan, “Robust Distance Metric Learning via Bayesian Inference,” *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1542-1553, 2018.

[۱۸] J. Kim, Y. He, and H. Park, “Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework,” *Journal of Global Optimization*, vol. 58, no. 2, pp. 285-319, 2014.

[۱۹] Y. Xu, and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758-1789, 2013.

[۲۰] D. Tax, “Data description toolbox dd tools 1.9. 0,” 2012.