ارائهٔ روشی جدید برای طبقه‌بندی چندبرچسبی برمبنای شبکه‌های عصبی

محسن نصیری[۱]، نگین دانشپور[۲]*

کارشناس‌ارشد دانشکدهٔ مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران[۱]

دانشیار دانشکدهٔ مهندسی کامپیوتر، دانشگاه تربیت دبیر شهید رجایی، تهران، ایران[۲]*

چکیده

طبقه‌بندی چندبرچسبی نوعی از طبقه‌بندی است که در آن نمونه‌ها می‌توانند صفر، یک یا بیش از یک برچسب داشته باشند؛ به‌عبارت‌دیگر هر نمونه به‌وسیلهٔ یک مجموعه از برچسب‌ها نمایش داده می‌شود. با توجه به پژوهش‌های اخیر، درنظرگرفتن ارتباط بین برچسب‌ها نتایج بهتری را حاصل می‌کند. در این مقاله برای درنظرگرفتن ارتباط بین برچسب‌ها، در مرحلهٔ نخست از خوشه‌بندی k- میانگین با محدودیت استفاده و در مرحلهٔ دوم برای هر خوشه یک شبکهٔ عصبی پرسپترون چندلایه درنظر گرفته شده‌است؛ درنهایت با ترکیب برچسب‌های پیش‌بینی‌شده به‌وسیلهٔ طبقه‌بندها، برچسب‌های نهایی به‌دست می‌آید. با توجه به اینکه تعداد شبکه‌های عصبی نسبت به حالت معمول افزایش و به‌تبع آن‌زمان آموزش داده‌ها بیشتر می‌شود، روش جدیدی برای کاهش ابعاد با استفاده از جمع پراکنده به‌کار برده شده‌است. با ارزیابی روش پیشنهادی بر روی مجموعه‌داده‌های موجود در مقایسه با روش‌های پیشین این نتیجه حاصل شد که روش پیشنهادی در سه مجموعه‌داده از نوع متن در بسیاری از معیارها مانند دقت، صحت و فاصلهٔ همینگ در بین الگوریتم‌ها رتبهٔ نخست را داشته است.

واژگان کلیدی: طبقه‌بندی، طبقه‌بندی چندبرچسبی، خوشه‌بندی، شبکه‌های عصبی.

Presenting a new method for multi label classification based on neural network

Mohsen Nasiri[1], Negin Daneshpour[2*]

Msc, Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran[1]

Associate Professor, Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran[2*]

Abstract

The problem of classification can be divided into two categories: single-label and multi-label. Single-label classification consists of binary and multi-class classification. In binary classification, the task is to predict one in two possible classes, such as distinguishing between spam and non-spam emails. In multi-class classification, the goal is to classify instances into more than two classes, such as identifying different species of flowers based on petal measurements. In contrast to single-label classification, multi-label classification is more complex because each instance could belong to multiple categories simultaneously. In multi-label learning, instead of assigning a single label to each instance, a set of labels is assigned. This means that each sample may have zero, one, or more than one associated label. For example, in a text classification task, a news article about technology and business might be labeled as both "Technology" and "Business". To handle multi-label classification, several approaches have been developed. One of the simplest methods is Binary Relevance (BR), which transforms the multi-label

* نویسندهٔ عهده‌دار مکاتبات

* Corresponding author

• تاریخ ارسال مقاله: ۱۴۰۳/۴/۲۴ • تاریخ پذیرش: ۱۴۰۳/۱۲/۱۴ • تاریخ انتشار: ۱۴۰۴/۶/۲۲ • نوع مطالعه: پژوهشی

problem into multiple independent binary classification tasks—one for each label. Although this approach is easy to implement, it treats each label independently and ignores possible relationships among them. However, in real-world applications, labels are often correlated; for instance, in medical diagnosis, certain diseases frequently appear together. In another approach, Label Powerset (LP), considers label dependencies by treating each unique combination of labels as a separate class. While this method captures relationships between labels, it suffers from scalability issues while dealing with a large number of labels, as the number of possible label combinations increases exponentially. To address these challenges, the proposed method incorporates k-means constraint clustering to group both labels and features prior to applying classification. In the first step, clustering is performed to group similar labels together, ensuring that label correlations are preserved. This also helps to mitigate the issue of imbalanced classification, where certain labels may be underrepresented in the dataset. Once the labels are being clustered, a separate multi-layer neural network would be assigned to each cluster. Instead of using a single large neural network for all labels, multiple smaller networks would be trained for different label clusters. This approach enhances learning efficiency and improves accuracy by focusing on relevant label groups. However, using multiple classifiers increases computational costs and training time. To mitigate this issue, a scatter-add dimension reduction technique is applied. Using scatter-add, attributes are efficiently assigned to the input of each neural network, ensuring that each classifier receives only the relevant feature subset. Each neural network then predicts labels within its designated cluster. Eventually, the predictions from all classifiers are combined to generate the final multi-label output for each instance. To evaluate the effectiveness of the proposed method, experiments were conducted on various text datasets. The results were compared with traditional multi-label classification methods, including Binary Relevance and Label Powerset. The evaluation has been based on several performance metrics, such as accuracy, precision, and hamming-loss. The results demonstrated that the proposed approach achieved superior performance across multiple datasets, ranking first in several evaluation criteria. Notably, it outperformed existing methods by a margin of approximately 1% in accuracy. These findings suggest that clustering-based multi-label classification using k-means constraint clustering and multi-layer neural networks is a promising approach. By leveraging label correlations and reducing dimensionality, the proposed method effectively improves classification performance while addressing issues such as label imbalance and computational inefficiency. Future research may further explore optimization techniques to reduce training time while maintaining high accuracy.

Keywords: Classification, Multi-Label Classification, Clustering, Neural Networks.

۱-مقدمه

طبقه‌بندی[1] فرایند یافتن مدلی[2] (یا تابعی) است که طبقات[3] یا مفاهیم داده را توصیف و جدا می‌کند. این مدل برای پیش‌بینی برچسب طبقهٔ اشیایی که برچسب طبقه برای آن‌ها ناشناخته است، استفاده می‌شود[۱]. مسئلهٔ طبقه‌بندی را به دو دستهٔ کلی طبقه‌بندی تک‌برچسبی[4] و طبقه‌بندی چندبرچسبی[5] می‌توان تقسیم کرد. در طبقه‌بندی رایج هر نمونه به یک برچسب مرتبط می‌شود؛ درحالی‌که در دنیای واقعی هر نمونه ممکن است به چند برچسب به‌طور هم‌زمان مربوط شود. از این جهت طبقه‌بندی چندبرچسبی به‌وجود آمده‌است[۲]. طبقه‌بندی چندبرچسبی طیف وسیعی از حوزه‌ها [۲۵-۲۶] را مانند «دسته‌بندی متن»[6] که هر متن یا خبر ممکن است به بیش از یک دسته تعلق داشته باشد، شامل می‌شود[۳]. ازجمله کاربردهای دیگر این نوع طبقه‌بندی می‌توان به دسته‌بندی محصولات[7] [۴]، پیشنهاد برچسب (هشتگ)[8] در شبکه‌های اجتماعی[۵]، طبقه‌بندی احساسات موسیقی[9] [۶] و تفسیر تصویر[10] [۷] اشاره کرد. در بسیاری از مقالات صفت‌های نمونه با متغیر x و برچسب‌هایی که قرار است برای نمونه‌ها پیش‌بینی شود، با متغیر y نشان داده شده‌است و درنهایت برای حل مسئلهٔ طبقه‌بندی چندبرچسبی باید زیرمجموعه‌ای از برچسب‌ها را به نمونهٔ مورد نظر نسبت داد (برای مثال برچسبی را که با نمونه مرتبط است با یک، در غیر این‌صورت با صفر نمایش دهیم) که باید در فضای 2^y جست‌وجو شود[۸، ۹، ۱۰]. این مسئله نمایی است و زمانی که تعداد برچسب‌ها بسیار باشد، مشکل تشدید می‌شود.

بسیاری از مقالات برای حل این مشکل، به خوشه‌بندی برچسب‌ها روی آوردند[۹، ۱۰، ۱۱] که این امر موجب کاهش فضای مسئله می‌شود؛ زیرا به‌جای بررسی تمامی برچسب‌ها و مقایسهٔ احتمال هر یک، خوشه‌های آن‌ها بررسی می‌شود.

[1] Classification
[2] Model
[3] Class
[4] Single label classification
[5] Multi-label classification
[6] Text categorization
[7] product categorization
[8] hashtag suggestion
[9] music emotion classification
[10] image annotation

در این مقاله نیز در ابتدا برچسب‌ها بر اساس داده و میزان ارتباط بین برچسب‌ها خوشه‌بندی می‌شوند. برای درنظرگرفتن ارتباط بین برچسب‌ها از خوشه‌بندی k- میانگین با محدودیت[1] بر اساس تکرار برچسب‌ها با یکدیگر و ماتریس شباهت استفاده شده‌است؛ سپس راه‌کاری برای حل مشکل نمایی مد نظر قرار می‌گیرد؛ همچنین برای بهبود دقت و سرعت در انتخاب برچسب‌ها ایدهٔ کاهش بعد بر اساس جمع پراکنده[2]استفاده شده‌است. مواردی که در این مقاله انجام شده به‌اختصار در زیر آورده شده‌است:

- خوشه‌بندی برچسب‌ها و صفت‌ها بر اساس ماتریس مشابهت
- کاهش بعد لایهٔ ورودی شبکهٔ عصبی
- متناسب‌کردن داده‌ها برای استفاده از جمع پراکنده و درنهایت کاهش بعد لایهٔ خروجی شبکهٔ عصبی

در ادامه، بخش دوم این مقاله به تشریح کارهای انجام‌شده می‌پردازد. در بخش سوم، الگوریتم مورد نظر با جزئیات معرفی می‌شود. نتایج مقایسات و آزمایش‌های انجام‌گرفته الگوریتم پیشنهادی و سایر روش‌ها در بخش چهارم به‌تفصیل بیان شده و در انتها و در بخش پنجم نیز به نتیجه‌گیری و پیشنهادهایی برای توسعهٔ الگوریتم مقاله پرداخته شده‌است.

۲- پیشینهٔ پژوهش

روش‌ها و الگوریتم‌های طبقه‌بندی چندبرچسبی معمول را می‌توان از چندین دیدگاه متفاوت دسته‌بندی و با هم مقایسه کرد. یک دیدگاه این است که ارتباط بین برچسب‌ها را در نظر بگیریم یا خیر. از جمله الگوریتم‌هایی که این ارتباط را در نظر نمی‌گیرند، الگوریتم ارتباط دودویی[3] است[۱۲]. این روش از یک گروه از طبقه‌بندهای جدا که هر کدام مسئولیت پیش‌بینی برچسب خود را دارند، تشکیل شده‌است. از مزایای این الگوریتم، ساده و شهودی‌بودن آن است؛ اما دستهٔ دوم ارتباط بین برچسب‌ها را در نظرمی‌گیرند. از جملهٔ این الگوریتم‌ها، طبقه‌بند زنجیره‌ای است. در این رویکرد ترتیب قرارگیری برچسب‌ها بسیار مهم و دشوار است. جون و همکاران برای ترتیب قرارگیری، شرط آنتروپی[4] را در نظر گرفتند[۱۳]. باید بین هر زوج برچسب با استفاده از قانون احتمال آنتروپی را حساب کرد. آن ترتیب که مقدار بیشتری داشت، ترتیب مناسب‌تری است. دستهٔ دیگری از الگوریتم‌هایی که ارتباط بین برچسب‌ها را در نظر می‌گیرند، طبقه‌بندهای گروهی هستند؛ از جملهٔ این الگوریتم‌ها می‌توان به k زیرمجموعه برچسب فعال اشاره کرد. وانگ و همکاران در مقالهٔ خود[۱۴] برای حل مشکل نمایی با درنظرگرفتن مجموعه‌هایی متشکل از k برچسب راه‌کاری پیشنهاد دادند. در این روش هر k برچسب (برای مثال k=۲) به‌صورت یک مجموعه‌اند. تمام زیرمجموعه‌های این مجموعه kعضوی بررسی و برای هر یک از این زیرمجموعه‌ها نامی در نظر گرفته می‌شود. از دیگر الگوریتم‌های مطرح می‌توان به ترکیب طبقه‌بندها بر اساس الگوریتم تکاملی اشاره کرد. بسیاری از روش‌های ارائه‌شدهٔ گروهی طبق انتخاب، تصادفی بوده و به ساختار داده توجه ندارند؛ درحالی‌که مویانو روشی پیشنهاد می‌دهد که به ساختار داده یعنی رابطهٔ بین برچسب‌ها و عدم تعادل در درجهٔ برچسب‌ها اشاره دارد[۱۵]. در این مقاله هر طبقه‌بند مسئولیت زیرمجموعه‌ای از برچسب‌ها را برعهده دارد. در این الگوریتم با استفاده از ابزارهای الگوریتم‌های تکاملی مانند تابع سازگاری[5]، عمل‌گر جهش [6]و عمل‌گر متقاطع[7] می‌توان دقت را افزایش داد. روش دیگری که نتایج بسیار خوبی در این زمینه داشته است، درنظرگرفتن ارتباط نمونه‌ها و صفت‌ها با یکدیگر و استفاده از روش k-نزدیکترین همسایه[8] است؛ همچنین در این مقاله به ارتباط بین برچسب‌ها با این فرض که برچسب‌های با ارتباط زیاد، خروجی مشابهی تولید می‌کند، توجه شده‌است. رویکرد دیگری که به‌تازگی منتشر شده، روش ترکیبی MLDE است[۱۸]. در این رویکرد، به‌صورت پویا طبقه‌بندهایی که بهترین دقت را دارند، انتخاب می‌شوند. طبق ادعای مقاله، تنها از طبقه‌بندهای دودویی استفاده شده و ممکن است با تغییر طبقه‌بندها حتی بتوان به نتایج بهتری دست یافت. مشکل پیچیدگی زمان، حافظه و برچسب‌های دنباله[9] در طبقه‌بندی چندبرچسبی با تعداد برچسب بالا بسیار شدیدتر است. برای حل این مشکل به‌طورمعمول راه‌کارها قبل از اجرای الگوریتم اصلی، پیش‌پردازش انجام می‌دهند. در حالت کلی می‌توان به سه رویکرد اشاره کرد[۸]؛ روش یک در مقابل همه[10]، حین فرایند آموزش برای هر برچسب یک طبقه‌بند، احتمال آن برچسب را در مقابل سایر برچسب‌ها محاسبه می‌کند. این روش از نظر دقت از بهترین‌ها محسوب می‌شود؛ اما چون زمان اجرای الگوریتم بسیار بالاست، در مواقعی که منابع محدود باشد، استفاده از این الگوریتم غیرممکن است. رویکرد دوم، روش‌های مبتنی بر درخت[11] است. ابتدا برچسب‌ها را به مجموعه‌های مجزا تقسیم می‌کنند؛ سپس بر اساس درخت تصمیم ایجادشده، برچسب‌گذاری نهایی را انجام می‌دهند. از مزایای این الگوریتم‌ها می‌توان به‌سرعت بالا در آموزش داده‌ها و پیش‌بینی برچسب‌ها در زمان اجرا اشاره کرد. از معایب آن مشکل انتشار در درخت و خطا در پیش‌بینی برچسب‌های دنباله است. رویکرد سوم روش‌های مبتنی بر یادگیری روش‌ها نیز تا اندازه‌ای سرعت بالایی در آموزش دارند.

[1] k-means constraint
[2] scatter_add
[3] Binary Relevance
[4] Conditional entropy

[5] Fitness function
[6] Mutation operator
[7] Crossover operator
[8] k-nearest neighbor
[9] tail labels
[10] one vs rest
[11] tree based method

(جدول-۱): دسته‌بندی روش‌های طبقه‌بندی
(Table-1): classification methods in 2017 to 2023

الگوریتم	سال	رویکرد	مزایا
Br Binary relevance[12]	۲۰۱۷	برای هر برچسب یک طبقه‌بند جدا در نظر می‌گیرد و به ارتباط برچسب‌ها توجه ندارد	ساده و شهودی بودن
Cc Classifier chain[13]	۲۰۱۹	بین هر زوج برچسب با استفاده از قانون احتمال، آنتروپی را حساب می‌کند.	ارتباط بین برچسب‌ها را در نظر می‌گیرد
Hierarchical Neural networks[16]	۲۰۱۴	بر اساس الگوریتم پرسپترون و انتشار مجدد در داده-های آموزش	از نظر پیچیدگی محاسباتی برای مجموعه با تعداد زیاد برچسب داده کارآمد است
k-labelset ACkEL[14]	۲۰۲۰	در این رویکرد از طبقه‌بندهای متفاوت استفاده شده و هر طبقه‌بند مسئولیت زیرمجموعه‌ای از برچسب‌ها را برعهده دارد.	-بهبودیافتهٔ روش lp است. - زمانی که تعداد برچسب‌ها و نمونه‌ها زیاد باشد مناسب است.
Bonsai[8]	۲۰۲۰	برچسب‌ها بر اساس ارتباطشان خوشه‌بندی می‌شوند و سپس درخت تصمیم ساخته می‌شود و درنهایت زیرمجموعهٔ مطلوبی از برچسب‌ها برگزیده می‌شود.	-نیازی به متعادل‌بودن بین برچسب‌ها ندارد؛ بنابراین اعتبار در تمامی داده‌ها برقرار است. -در مجموعه‌های داده با برچسب بالا به ویژه آن دسته که میانگین برچسب کمی دارند خوب عمل می‌کند
Evolutionary algorithm Ensemble generation[15]	۲۰۲۰	الگوریتم از ابزارهای الگوریتم‌های تکاملی مانند تابع سازگاری عمل‌گر جهش و عمل‌گر متقاطع استفاده کرده‌است.	داده‌های با تکرار کم را نیز پوشش می‌دهد و نادیده نمی‌گیرد.
CLML[17]	۲۰۲۲	انتخاب صفت‌های خاص با استفاده از ارتباط نمونه‌ها و صفت‌ها و استفاده از k-نزدیکترین همسایه	با توجه به انتخاب صفت‌ها و درنظرگرفتن ارتباط برچسب‌ها دقت نسبت به سایر روش‌ها بالاتر است.
MLDE[18]	۲۰۲۳	انتخاب بهترین طبقه‌بند پایه و ترکیب پیش‌بینی آن‌ها برای به‌دست‌آمدن برچسب نهایی	به‌دلیل اینکه یک رویکرد و نگرش کلی است می-توان با تغییر طبقه‌بندها و جزئیات به نتایج بهتری دست یافت

۳- طرح پیشنهادی

همان‌طورکه بیان شد، در طبقه‌بندی چندبرچسبی هدف آن است که زیرمجموعه‌ای (برحسب معمول کوچک) از تعداد زیادی برچسب به نمونه نسبت داده شود. در رویکرد پیشنهادی ابتدا برچسب‌ها خوشه‌بندی، سپس برای هر یک از خوشه‌های برچسب یک شبکهٔ عصبی در نظر گرفته می‌شود؛ درنهایت با ترکیب برچسب‌های پیش‌بینی‌شده به‌وسیلهٔ شبکه‌های عصبی، برچسب‌های نهایی به‌دست می‌آیند؛ همچنین راه‌کاری برای کاهش بعد شبکه‌های عصبی با استفاده از جمع پراکنده در نظر گرفته شده‌است. در ادامه به توضیح هر یک از موارد پرداخته شده‌است.

۳-۱- خوشه‌بندی k-میانگین با محدودیت

در این خوشه‌بندی ابتدا با پیش‌پردازش در قسمت ۳-۱-۱ ارتباط بین برچسب‌ها (صفت‌ها) به‌دست می‌آید؛ سپس برچسب‌های با ارتباط بیشتر در یک خوشه قرار می‌گیرند. تفاوت این خوشه‌بندی با k-میانگین متداول در این است که می‌توان بیشینه[1] و کمینهٔ[2] تعداد برچسبی را که در هر خوشه قرار می‌گیرد، تعیین کرد.

۳-۱-۱-پیش‌پردازش

این مرحله شامل یافتن ارتباط اولیهٔ صفت‌ها با یکدیگر، برچسب‌ها با یکدیگر و خوشه‌بندی آن‌هاست. خوشه‌بندی k- میانگین با محدودیت بر اساس ماتریس شباهت به‌دست می‌آید و به همین دلیل مناسب مجموعه‌داده‌هایی است که ارتباط بین برچسب‌ها به نسبت سایرین بالاتر است. در ادامه به توضیحات این موارد پرداخته شده‌است.

۳-۱-۱-۱- ارتباط بین برچسب‌ها و خوشه‌بندی آن‌ها

می‌توان برچسب‌ها را در ماتریس دو بعدی دودویی نمایش داد. هر سطر ماتریس بیان‌کنندهٔ برچسب‌های هر نمونه است. در صورتی که آن نمونه آن برچسب را داشته باشد مقدار آن یک و در غیر این صورت صفر است.

$$Y=\begin{bmatrix}1 & 1 & 0\\ 1 & 1 & 0\\ 0 & 1 & 1\end{bmatrix}$$

برای مثال در ماتریس بالا سطر نخست بیان‌کنندهٔ نمونه نخست و ستون نخست و دوم آن یک است؛ پس این نمونه برچسب نخست (y1) و برچسب دوم (y2) را دارد و ستون سوم صفر است؛ پس این نمونه فاقد برچسب سوم (y3) است؛ همچنین سطرهای دوم و سوم ماتریس بیان‌کنندهٔ برچسب‌های نمونه دوم و سوم هستند.

[1] maximum
[2] minimum

ضرب ترانهادهٔ ماتریس Y در ماتریس Y ماتریس شباهت Z را به‌وجود می‌آورد که بیان‌کنندهٔ ارتباط جبری بین برچسب‌هاست:

$Z= Y^t Y$

برای مثال داریم:

$$Z=\begin{bmatrix} x & 2 & 0 \\ 2 & x & 1 \\ 0 & 1 & x \end{bmatrix}$$

در ماتریس شباهت Z قطر اصلی ارتباط هر برچسب با خودش است که در خوشه‌بندی تأثیری ندارد. در مورد سایر درایه‌ها Z_{ij} ارتباط برچسب i و برچسب j را بیان می‌کند؛ برای مثال در ماتریس بالا Z_{12} برابر دو است که بیان‌کنندهٔ آن است که برچسب‌های یک و دو، دو مرتبه در دو نمونهٔ متفاوت هم‌زمان حضور داشته‌اند. پس این دو برچسب، ارتباط قوی‌تری نسبت به برچسب سوم دارند؛ زیرا $Z_{31} = Z_{13} = 0$ که بیان‌کنندهٔ آن است که برچسب یک و سه در هیچ نمونه‌ای هم‌زمان رخ نداده‌اند و در خوشه‌های متفاوتی قرار خواهند گرفت.

اگر ماتریس Z به‌عنوان ورودی به تابع خوشه‌بند ارسال شود، درنهایت یک آرایه برمی‌گرداند که هر عنصر آن بیان‌کنندهٔ شماره خوشهٔ آن برچسب است؛ برای مثال اگر ماتریس بالا به‌عنوان ورودی تابع باشد و k را برابر دو قرار دهیم، خروجی آن به‌صورت آرایهٔ زیر است:

$[1 \quad 1 \quad 0]_{1*3}$

درایه‌های نخست و دوم یک هستند و این یعنی شماره خوشهٔ برچسب نخست و دوم یک است و هردو برچسب در خوشهٔ یک قرار دارند؛ همچنین درایهٔ سوم صفر، که نشان‌دهندهٔ آن است که برچسب سوم در خوشهٔ شمارهٔ صفر قرار دارد.

۳-۱-۲-ارتباط بین صفت‌ها و خوشه‌بندی آن‌ها

صفت‌ها (ویژگی‌ها) را نیز مانند برچسب‌ها می‌توان به‌صورت ماتریس دو بعدی نمایش داد. هر سطر ماتریس بیان‌کنندهٔ صفت‌های هر نمونه است؛ برای مثال ماتریس زیر بیان‌کنندهٔ صفت‌های سه نمونه است. (هر سطر صفت‌های یک نمونه را نشان می‌دهد)

$$X=\begin{bmatrix} 0.5 & 0 & 1 & 10 \\ 1 & 6.2 & 3.1 & 0 \\ 0 & 0 & 4 & 6 \end{bmatrix}$$

درایهٔ X_{ij} مقدار j اُمین صفت از i امین نمونه را بیان می‌کند؛ برای مثال در ماتریس بالا X_{22} بیان می‌کند که مقدار صفت سوم از نمونهٔ دوم برابر 3.1 است.

همان‌طورکه مشخص است، تنها تفاوت در این است که برچسب‌ها به‌صورت دودویی (صفر یا یک) بودند؛ درصورتی‌که صفت‌ها مقادیر متفاوتی دارند؛ به همین علت قبل از ضرب ماتریس صفت در ترانهادهٔ آن و به‌دست‌آوردن ماتریس Z که ارتباط بین صفت‌ها با هم را بیان می‌کند، داده‌ها نرمال‌سازی شده و در بین بازه صفر و یک قرار می‌گیرند. برای این کار از رابطهٔ (۱) استفاده شده‌است:

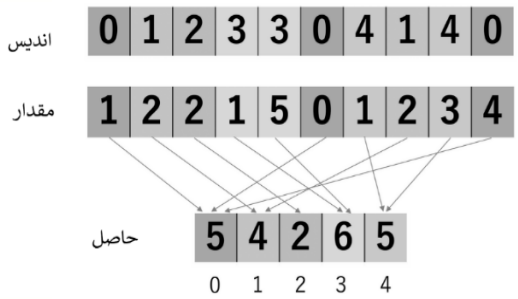
$$x_i' = \frac{xi - min(x)}{max(x) - min(x)} \qquad (۱)$$

۳-۲-کاهش بُعد لایه‌های شبکهٔ عصبی

در مرحلهٔ قبل، برچسب‌ها و صفت‌ها خوشه‌بندی شدند. برای تبدیل‌شدن خوشه به‌عنوان گره شبکه، دو راه‌کار وجود دارد؛ راه نخست درنظرگرفتن مرکز هر خوشه به‌عنوان نمایندهٔ آن خوشه است. این رویکرد با اینکه در پیچیدگی زمانی مناسب‌تر است، اما دقت کافی را ندارد؛ زیرا تأثیر تمام صفت‌ها در گره‌ها اعمال نمی‌شود؛ اما راه‌کار دوم رویکرد جمع پراکنده است که در قسمت ۳-۲-۱ توضیح داده شده‌است.

۳-۲-۱-رویکرد جمع‌کردن پراکنده

با توجه به این‌که در این مقاله برای کاهش ابعاد در لایهٔ ورودی و در لایهٔ خروجی شبکهٔ عصبی از رویکرد «جمع‌کردن پراکنده»[1] استفاده شده، بهتر است در این بخش به توضیح این روش پرداخته شود. سه آرایه با نام‌های اندیس، مقدار و حاصل را مطابق شکل (۱) در نظر بگیرید.



(شکل-۱): مثالی برای جمع پراکنده
(figure-1): example of scatter_add

برای مثال در آرایهٔ اندیس، سه درایه با مقدار صفر وجود دارد. درایه‌های متناظر در آرایهٔ مقدار با هم جمع عددی شده و حاصل در درایهٔ صفرم آرایهٔ حاصل قرار می‌گیرد. به همین شکل تمام درایه‌های آرایهٔ حاصل پر می‌شود.

در ادامه به ارتباط جمع پراکنده و روش این مقاله می‌پردازیم.

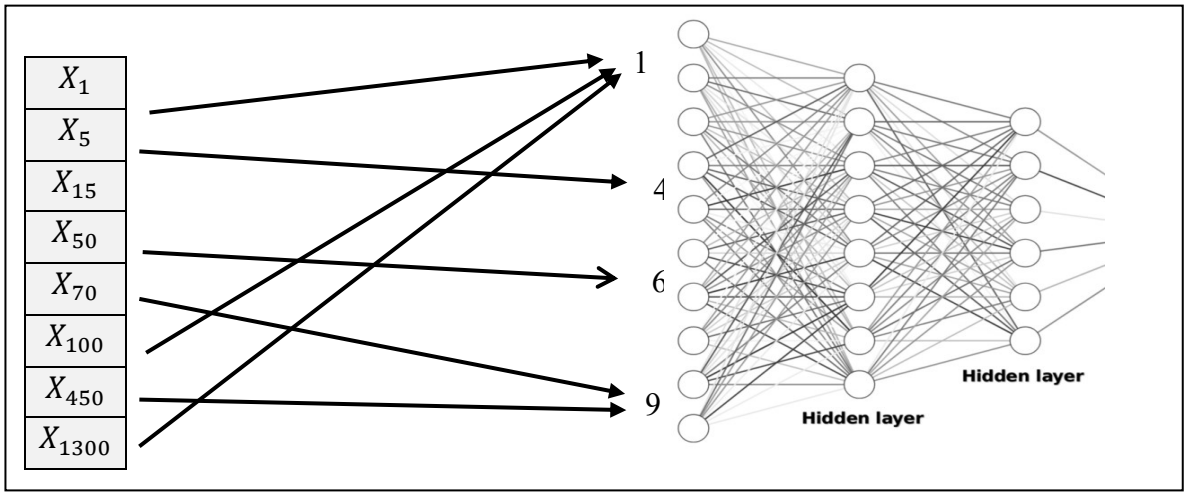
۳-۲-۲-کاهش بُعد لایهٔ ورودی شبکه

همان طور که در قسمت ۳-۲ گفته شد، ورودی شبکه شماره خوشه‌های صفت‌هاست (نه خود صفت‌ها)؛ پس باید تابعی طراحی شود که هر بار نمونهٔ جدیدی دریافت کرده و شماره خوشه‌هایی که صفت‌های نمونه در آن قرار دارند را در اختیار شبکهٔ عصبی قرار دهد. به شکل (۲) توجه فرمایید.

[1] scatter_add

همان‌طورکه در شکل (۲) پیداست، نمونهٔ جدید با صفت‌های خود وارد می‌شود. صفت‌هایی که در خوشهٔ یک قرار دارند، در شبکهٔ عصبی گره یک را فعال می‌کنند. صفت‌هایی که در خوشهٔ نُه قرار دارند، گره نُه را در شبکه فعال کرده‌اند.

از خوشهٔ نخست سه صفت، درصورتی‌که از خوشهٔ شش تنها یک صفت موجود است؛ پس منطقی است که وزن اولیهٔ خوشهٔ یک، سه برابر وزن خوشهٔ شش باشد. در این قسمت از جمع‌کردن پراکنده که در قسمت ۳-۳-۱ بیان شد، استفاده شده‌است.



(شکل-۲): شمارهٔ خوشه‌ها به‌عنوان ورودی شبکه
(figure-2): Number of clusters as network input

۳-۲-۳-مرحلهٔ نهایی برچسب‌زنی

بعد از مشخص‌شدن وزن خوشه‌های برچسب، وزن برچسب‌های نهایی تعیین می‌شود. باید برای هر یک از خوشه‌های برچسب، یک شبکهٔ عصبی جداگانه طراحی شود، به‌گونه‌ای‌که ورودی این شبکه‌ها همان ورودی شبکهٔ نخست شکل (۱) یعنی همان خوشه‌های صفت‌ها باشد. خروجی این شبکه‌ها نیز برچسب‌های خوشهٔ مورد نظر است؛ برای مثال اگر تعداد خوشه‌های برچسب ده باشد، باید ده شبکهٔ عصبی جداگانه طراحی شود که هر کدام از شبکه‌ها برچسب‌های نهایی را پیش‌بینی می‌کنند. لایهٔ ورودی این ده شبکه، یکسان بوده و همان لایهٔ ورودی شبکهٔ اصلی یعنی خوشه‌های صفت است.

در لایهٔ آخر شبکه باید برای هر گره یک تابع فعال‌ساز در نظر گرفت. در شکل (۱) خوشه‌های برچسب مشخص شده‌اند و همان‌طورکه پیدا است تعداد برچسب‌ها در هر خوشه کمتر از تعداد کل برچسب‌هاست. به‌طور دقیق‌تر اگر تعداد کل برچسب‌ها N و تعداد خوشه‌ها L باشد، تعداد عناصر هر خوشه به‌طور متوسط برابر V خواهد بود و رابطهٔ (۲) برقرار است:

$$V = \frac{N}{L} \qquad (۲)$$

این، نویدبخش آن است که شبکه‌های ثانویه که قرار است برچسب‌های نهایی را تعیین کنند، دارای گره‌های کمتری خواهند بود؛ و این باعث کمترشدن هزینهٔ زمانی و حافظه می‌شود.

درعمل ماتریسی که به‌عنوان لایهٔ آخر این شبکه‌ها قرار می‌گیرد، به تعداد کل برچسب‌ها درایه دارد؛ و برای هر درایه باید تابع فعال‌ساز جداگانه تعریف کرد؛ به همین جهت درعمل، کاهش هزینه‌ای که مدنظر بود، محقق نمی‌شود. برای رفع این مشکل از کاهش ابعاد به روش جمع پراکنده استفاده شده‌است و آرایه‌ای که به‌عنوان لایهٔ خروجی شبکه‌های ثانویه قرار می‌گیرد، به تعداد عناصر همان خوشهٔ (V) است و دیگر به تعداد عناصر کل برچسب‌ها (N) نیست.

۳-۲-۴-مراحل ساخت لایهٔ خروجی شبکه‌های ثانویه

همان‌طورکه در قسمت قبل بیان شد، آرایه‌هایی که به‌عنوان لایهٔ خروجی شبکه‌ها قرار می‌گیرند، به تعداد کل برچسب‌ها درایه دارند. در این بخش روش کاهش تعداد درایه‌های آرایه‌ها بیان می‌شود.

بعد از این که برچسب‌ها خوشه‌بندی شدند، در خروجی آرایه‌ای خواهیم داشت که نشان می‌دهد هر برچسب متعلق به کدام خوشه است؛ برای مثال اگر تعداد کل برچسب‌ها ده و تعداد خوشه‌ها سه باشد، آرایهٔ زیر را در نظر بگیرید:

$$[1\ \ 1\ 3\ 2\ \ 2\ \ 3\ \ 3\ \ 3\ 1\ \ 2\ ]_{1*10}$$

درایه‌های نخست، دوم و نهم مقدار یک دارند و این یعنی برچسب‌های نخست و دوم و نهم در خوشهٔ یک هستند. درایه‌های سوم، ششم، هفتم و هشتم مقدار صفر

دارند و در خوشهٔ سه قرار دارند؛ به همین ترتیب درایه‌های چهارم، پنجم و دهم در خوشهٔ دو قرار گرفته‌اند.

مرحلهٔ نخست:

در ابتدا خوشه‌ها جداسازی می‌شوند؛ برای مثال آرایهٔ بالا به سه آرایهٔ زیر تبدیل می‌شود:

$$[1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0]_{1*10}$$
$$[0\ \ 0\ \ 0\ \ 2\ \ 2\ \ 0\ \ 0\ \ 0\ \ 0\ \ 2]_{1*10}$$
$$[0\ \ 0\ \ 3\ \ 0\ \ 0\ \ 3\ \ 3\ \ 3\ \ 0\ \ 0]_{1*10}$$

آرایهٔ نخست بیان‌کنندهٔ خوشهٔ یک، آرایهٔ دوم بیان‌کنندهٔ خوشهٔ دو و آرایهٔ سوم بیان‌کنندهٔ خوشهٔ سه است. همان طور که پیداست برای مثال در آرایهٔ نخست عناصر غیر از یک همگی به صفر تبدیل شده‌اند و همین روند برای خوشه‌های دیگر نیز برقرار است.

مرحلهٔ دوم:

در مرحلهٔ بعد ماتریس دوبعدی برچسب‌ها در هر یک از خوشه‌های جداسازی‌شده ضرب می‌شود؛ برای مثال اگر سه نمونه وجود داشته باشد، ضرب ماتریس برچسب آن سه نمونه در خوشهٔ نخست به‌صورت زیر است:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} * [1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0]_{1*10} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

ضرب صورت گرفته، یک ضرب درایه به درایه و با ضرب متداول ماتریسی متفاوت است. ماتریس حاصل ضرب بیان‌کنندهٔ آن است که در هر نمونه کدام برچسب در خوشهٔ نخست قرار دارد.

مرحلهٔ سوم:

ماتریس خروجی مرحلهٔ دوم تعداد زیادی سطر دارد که تمام درایه‌های آن صفر است. در این مرحله تمام سطرهای صفر در ماتریس و همچنین سطرهای نظیر آن در ماتریس ورودی شبکه پاک می‌شوند.

مرحلهٔ چهارم:

در این مرحله باید آرایهٔ جداسازی‌شده در مرحلهٔ نخست که بیان‌کنندهٔ خوشهٔ مورد نظر است، آمادهٔ جمع‌کردن پراکنده (که در قسمت ۲-۱ بیان شد) بشود. اگر به درایه‌های غیر صفر آرایه، به‌ترتیب مقدار یک اضافه شود، این امکان صورت خواهد گرفت؛ برای مثال خوشهٔ نخست به‌صورت زیر است:

$$[1\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0]_{1*10}$$

پس از اضافه‌شدن مقدار متوالی یک، آرایه به‌شکل زیر تغییر می‌کند:

$$[1\ \ 2\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 3\ \ 0]_{1*10}$$

مرحلهٔ پنجم:

ماتریس به‌دست‌آمده از مرحلهٔ سوم با آرایهٔ به‌دست‌آمده از مرحلهٔ چهارم جمع پراکنده (در قسمت ۲-۱) می‌شوند و ماتریس حاصل به‌عنوان لایهٔ خروجی شبکه‌ها قرار می‌گیرد و بدین شکل کاهش ابعاد انجام خواهد گرفت. شبه‌کد روش این مقاله در شکل (۳) ملاحظه می‌شود.

همان‌طورکه در الگوریتم (۱) مشخص است، ورودی الگوریتم نمونه‌ها که با ویژگی‌های خود بیان می‌شوند هستند و خروجی الگوریتم مجموعه برچسب پیش‌بینی شده‌است.

در خط نخست، ویژگی‌ها خوشه‌بندی می‌شوند. ورودی این خوشه‌بند عبارت است از:

-تعداد خوشهٔ (k)

- بیشینه برچسبی که در یک خوشه قرار می‌گیرد (max)

- کمینه برچسبی که در یک خوشه قرار می‌گیرد (min)

سه ورودی بالا را برنامه‌نویس تعیین می‌کند.

(الگوریتم-۱): روش پیشنهادی خوشه‌بندی
(Algorithm-1): Proposed Algorithm

Algoritm1
Input: samples with their features
Output: set of possible labels
1. $S_1, \ldots, S_k \leftarrow$ K-means-constraint(K ,max, min)
2.new-features $\leftarrow$ scatter-add(features, clusters)
3. $S_1, \ldots, S_{k'} \leftarrow$ K-means-constraint(K' ,max, min)
4.new-labels $\leftarrow$ scatter-add(labels, clusters)
5. $V_1, \ldots, V_{k'} \leftarrow$separating(clusters)
6. new-out $\leftarrow$multiplication(new-labels , labels)
7.training K' neural networks
8. concatenate possible of each neural networks
9.return final set of labels.

در خط دوم به‌وسیلهٔ جمع پراکندهٔ ویژگی‌های جدید ساخته می‌شود. در خط سوم و چهارم، الگوریتم مراحلی را که برای ویژگی‌ها انجام شد، برای برچسب‌ها تکرار می‌کند. خط پنجم و ششم جداسازی خوشه‌ها و ضرب نقطه‌به‌نقطه در برچسب‌های اولیه برای آماده‌شدن در آموزش شبکه‌های عصبی است. در خط هفتم، شبکه‌های عصبی که به تعداد خوشه‌ها هستند، آموزش می‌بینند. در خط هشتم و نهم، برچسب‌های پیش‌بینی‌شده با هم ادغام و درنهایت برچسب‌های نهایی برگردانده می‌شود.

۴- آزمایش‌ها و نتایج

برای پیاده‌سازی الگوریتم‌ها از زبان برنامه‌نویسی Python 3.8.8 در سیستم عامل Windows 10 x64 استفاده شد.

برای ارزیابی الگوریتم در آزمایش‌های از مجموعه‌داده‌های Stackex_cs،Stackex_cooking و delicious که از مجموعه‌داده‌های سایت دانشگاه کوردوبا[1] است، استفاده شده‌است؛ همچنین نتایج این مقاله با بهترین الگوریتم‌های موجود (از این جهت بهترین‌اند؛ زیرا مقالاتی که در سال اخیر در مجلات معتبر چاپ شده‌اند، نتایج را با این الگوریتم‌ها مقایسه کرده‌اند) که LIFT [18]،LLSF[19] ، LLSF-DL [20]، MLCIB [21]،JLCLS[23] ، [22] LSF-CI،[24]LFCMLL و CLML[17] هستند، مقایسه شده‌است. مجموعه‌داده‌های استفاده‌شده در مقالات از دو بخش ویژگی‌ها و برچسب‌های استخراج‌شده است. مشخصات مجموعه‌داده‌ها در جدول (۲) آورده شده‌است:

(جدول-۲): مشخصات مجموعه‌داده‌های استفاده‌شده
(Table-2): Multi-label datasets used in the experiment

مجموعه‌داده	تعداد نمونه	تعداد برچسب	تعداد ویژگی	کاردینالیتی[2]	C_{VIR}[3]
delicious	۱۶۱۰۵	۹۸۳	۵۰۰	۱۹.۰۲	۰.۷۴
Stackex_cs	۹۲۷۰	۲۷۴	۶۳۵	۲.۵۶	۰.۷۶
Stackex_cooking	۱۰۴۹۱	۴۰۰	۵۷۷	۲.۲۳	۰.۶۵

در جدول (۲) کاردینالیتی بیان‌کنندهٔ میانگین برچسب برای هر نمونه است؛ همچنین CVIR ضریب تغییرات نسبت عدم تعادل برای تمام برچسب‌های طبقه است.

delicious یک مجموعه‌دادهٔ وب‌محور است که از تارنمای Delicious (یک سامانهٔ نشانه‌گذاری اجتماعی) استخراج شده‌است؛ هر نمونه مربوط به یک صفحهٔ وب است و برچسب‌های آن عبارت‌اند از برچسب‌هایی که کاربران به آن صفحه داده‌اند. Stackex_cs زیرمجموعه‌ای از داده‌های StackExchange است و بهطور خاص شامل پست‌های مربوط به Computer Science می‌شود. Stackex_cooking مشابه Stackex_cs است، اما حوزهٔ آن مرتبط با آشپزی و غذا است. هر نمونه یک پست دربارهٔ آشپزی بوده و برچسب‌ها موضوعات آشپزی مانند مواد غذایی، تکنیک‌های آشپزی و ابزارهای آشپزی هستند.

برای ارزیابی میزان کارایی نتایج طبقه‌بندی، در آزمایش‌های از معیارهای Accuracy، precision و F-measure استفاده شده‌است. در این معیارها هر چه عدد به‌دست‌آمده بالاتر باشد، به معنای کارایی بهتر روش طبقه‌بندی است. بیشینهٔ مقدار این معیارها برابر با یک و نشان‌دهندهٔ بهترین کارایی است؛ همچنین از معیار hamming loss نیز بهره برده شد. برخلاف سه معیار دیگر در این معیارها هر چه عدد به‌دست‌آمده پایین‌تر باشد، به معنای کارایی بهتر روش طبقه‌بندی است.

معیار Accuracy (در جداول به‌اختصار Acc است) در رابطهٔ (۳) بیان شده‌است. برای به‌دست‌آوردن Accuracy، ابتدا نسبت برچسب‌های پیش‌بینی‌شدهٔ درست را به کل برچسب‌های فعال برای هر یک از نمونه‌ها حساب می‌کنیم. میانگین این نسبت‌ها برابر Accuracy است. (Z_i برچسبی است که مدل پیش‌بینی کرده‌است و Y_i برچسب درست است.)

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \qquad (۳)$$

برای معیار Precision (در جداول به‌اختصار prec) رابطهٔ (۴) برقرار است. برای به‌دست‌آوردن این معیار ابتدا برای هر نمونه نسبت برچسب‌های درست پیش‌بینی‌شده به کل برچسب‌های پیش‌بینی‌شده حساب می‌شود؛ سپس میانگین این اعداد بیان‌کنندهٔ Precision است.

$$\text{Precision} = \frac{1}{n}\sum_{i-1}^{n}\frac{|Y_i \cap Z_i|}{Z_i} \qquad (۴)$$

معیار دیگری که در این مقاله استفاده شده‌است، F-measure (در جداول به اختصار F1) است. برای به‌دست‌آوردن این معیار ابتدا لازم است معیاری به نام recall را معرفی کنیم. برای به‌دست‌آوردن این معیار ابتدا برای هر نمونه نسبت برچسب‌های درست پیش‌بینی‌شده به کل برچسب‌های درست حساب می‌شود؛ سپس میانگین این اعداد بیان‌کنندهٔ recall است که در رابطهٔ (۵) آمده‌است:

$$\text{recall} = \frac{1}{n}\sum_{i-1}^{n}\frac{|Y_i \cap Z_i|}{Y_i} \qquad (۵)$$

حال معیار F-measure از رابطهٔ (۶) به دست می‌آید:

$$\text{F} - \text{measure} = 2 * \frac{\text{Precision} * \text{recall}}{\text{Precision} + \text{recall}} \qquad (۶)$$

Hamming loss (در جداول به‌اختصارH-Loss) نیز از معیارهای دیگر است که از رابطهٔ (۷) به‌دست می‌آید. این معیار، تفاوت برچسب‌های درست و برچسب‌های پیش‌بینی‌شده را حساب می‌کند و درنهایت نسبت به تعداد برچسب (k) و تعداد نمونه (n) نرمال‌سازی می‌شود.

$$\text{Hamming loss} = \frac{1}{n}\frac{1}{k}\sum|Y_i \Delta Z_i| \qquad (۷)$$

[1] www.uco.es/kdis/mllresources/
[2] Cardinality
[3] coefficient variation of the imbalance ratio

(جدول-۳): نتایج مقایسهٔ الگوریتم‌ها بر اساس چهار معیار روی مجموعه‌دادهٔ delicious

(Table-3): Experimental Results of All Comparing Algorithms on delicious Data Set in Terms of Four Evaluation Metrics

الگوریتم	Prec	Acc	H-Loss	F1
LLSF	۰.۳۸۳	۰.۱۹۶	۰.۰۲۲	۰.۲۹۹
LLSF-DL	۰.۴۹۹	۰.۱۸۳	۰.۰۱۹	۰.۲۸۱
LIFT	۰.۴۹۷	۰.۱۲۵	۰.۰۱۸	۰.۱۹۳
MLCIB	۰.۳۱۷	۰.۱۸۰	۰.۰۲۸	۰.۲۸۱
LSFCI	۰.۳۸۲	۰.۱۹۶	۰.۰۲۲	۰.۳۰۰
LSFMLL	۰.۰۹۲	۰.۰۲۳	۰.۰۱۹	۰.۰۳۴
JLCLS	۰.۳۱۶	۰.۰۵۲	۰.۰۱۹	۰.۰۸۸
CLML-L	۰.۳۲۴	۰.۲۰۴	۰.۰۲۶	۰.۳۱۴
CLML-C	۰.۳۱۷	۰.۲۰۱	۰.۰۲۷	۰.۳۱۱
CLML	۰.۳۱۷	۰.۲۰۱	۰.۰۲۷	۰.۳۱۱
الگوریتم پیشنهادی	۰.۴۰۱	۰.۲۰۶	۰.۰۲۶	۰.۳۱۰

جداول (۴ و۶ و ۸) مناسب‌ترین حالت مدل (از نظر تعداد شبکهٔ عصبی و لایه‌های میانی) برای مجموعه‌داده‌های مختلف‌اند که بهترین نتایج را به‌دست آورده‌اند.

همان‌طورکه در جدول (۳) مشاهده می‌شود، accuracy الگوریتم پیشنهادی از تمامی الگوریتم‌های برگزیده بالاتر است. F1 با فاصلهٔ خیلی کمی از بهترین الگوریتم قرار دارد. Precision نیز مقام سوم را دارد.

تعداد و مشخصات مناسب‌ترین مدل و شبکه‌های عصبی در جدول (۴) آورده شده‌است:

(جدول-۴): مشخصات شبکهٔ عصبی استفاده‌شده برای مجموعه‌دادهٔ delicious

(Table-4): Description of neural network used for delicious dataset

تعداد شبکه	تعداد لایهٔ میانی	epoch	Learning rate	Time(h)
۱۰	۲	۱۵	۰.۰۱	۰.۳

بر اساس داده‌های جدول (۵) Precision, accuracy و F1الگوریتم پیشنهادی از تمامی الگوریتم‌های برگزیده با اختلاف زیاد بالاتر است. فاصلهٔ hamming در فاصلهٔ بسیار کمی از بهترین الگوریتم قرار دارد.

(جدول-۵): نتایج مقایسه الگوریتم‌ها بر اساس چهار معیار روی مجموعه‌دادهٔ stackex_cooking

(Table-5): Experimental Results of All Comparing Data Set in Terms of Four Algorithms on stackex cooking Evaluation Metrics

الگوریتم	Prec	Acc	H-Loss	F1
LLSF	۰.۴۸۰	۰.۳۰۹	۰.۰۰۵	۰.۳۸۹
LLSF-DL	۰.۳۰۱	۰.۲۴۵	۰.۰۱۰	۰.۳۵۰
LIFT	۰.۳۶۱	۰.۱۹۷	۰.۰۰۴	۰.۲۴۷
MLCIB	۰.۴۰۱	۰.۲۸۲	۰.۰۰۷	۰.۳۷۱
LSFCI	۰.۴۸۰	۰.۳۰۰	۰.۰۰۵	۰.۳۷۶
LSFMLL	۰.۱۴۴	۰.۰۷۹	۰.۰۰۵	۰.۰۹۹
JLCLS	۰.۰۱۷	۰.۰۰۹	۰.۰۰۶	۰.۰۱۱
CLML-L	۰.۴۶۰	۰.۳۲۷	۰.۰۰۶	۰.۴۲۲
CLML-C	۰.۴۶۹	۰.۳۲۶	۰.۰۰۶	۰.۴۱۸
CLML	۰.۴۶۸	۰.۳۲۶	۰.۰۰۶	۰.۴۱۹
الگوریتم پیشنهادی	۰.۷۴۶	۰.۴۳۶	۰.۰۰۵	۰.۵۷۳

تعداد و مشخصات شبکه‌های عصبی در جدول (۶) آورده شده‌است:

(جدول-۶): مشخصات شبکهٔ عصبی استفاده‌شده برای مجموعه‌دادهٔ stackex_cooking

(Table-6): Description of neural network used for stackex_cooking dataset

تعداد شبکه	تعداد لایهٔ میانی	epoch	Learning rate	Time(h)
۴	۱	۱۲۰۰	۰.۰۱	۲.۵

با توجه به نتایج جدول (۷) Precision, accuracy و f1 الگوریتم پیشنهادی از تمامی الگوریتم‌های برگزیده با اختلاف زیاد بالاتر است. فاصلهٔ hamming در فاصلهٔ بسیار کمی از بهترین الگوریتم قرار دارد.

(جدول-۷): نتایج مقایسهٔ الگوریتم‌ها بر اساس چهار معیار روی مجموعه‌دادهٔ stackex_cs

(Table-7): Experimental Results of All Comparing Data Set in Terms of Four Algorithms on stackex_cs Evaluation Metrics

الگوریتم	Prec	Acc	H- Loss	F1
LLSF	۰.۳۷۰	۰.۲۷۷	۰.۰۱۳	۰.۳۸۱
LLSF-DL	۰.۳۲۴	۰.۲۵۵	۰.۰۱۶	۰.۳۶۲
LIFT	۰.۳۱۵	۰.۱۶۲	۰.۰۰۸	۰.۲۱۱
MLCIB	۰.۳۵۴	۰.۲۵۹	۰.۰۱۴	۰.۳۵۵
LSFCI	۰.۴۰۱	۰.۲۸۵	۰.۰۱۱	۰.۳۸۳
LSFMLL	۰.۱۲۴	۰.۰۶۶	۰.۰۰۹	۰.۰۸۷
JLCLS	۰.۰۴۲	۰.۰۱۷	۰.۰۰۹	۰.۰۲۳
CLML-L	۰.۴۲۶	۰.۳۱۲	۰.۰۱۱	۰.۴۲۰
CLML-C	۰.۴۲۲	۰.۳۱۵	۰.۰۱۱	۰.۴۲۳
CLML	۰.۴۲۶	۰.۳۱۷	۰.۰۱۱	۰.۴۲۶
الگوریتم پیشنهادی	۰.۶۹	۰.۴۱	۰.۰۰۹	۰.۵۵

تعداد و مشخصات شبکه‌های عصبی در جدول (۸) آورده شده‌است:

(جدول-۸): مشخصات شبکهٔ عصبی استفاده‌شده برای مجموعه‌دادهٔ stackex_cs

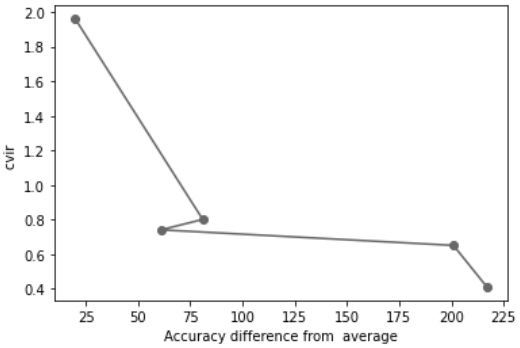
(Table-8): Description of neural network used for stackex_cs dataset

تعداد شبکه	تعداد لایهٔ میانی	epoch	Learning rate	Time(h)
۳	۱	۱۵۰۰	۰.۰۱	۲.۴

۴-۱-مشخصات طبقه‌بند شبکهٔ عصبی

در این مقاله از شبکهٔ عصبی چندلایهٔ پرسپترون استفاده شده‌است. از دلایل استفادهٔ این نوع شبکهٔ عصبی به‌عنوان طبقه‌بند پایه این است که به‌صورت بومی، مسئلهٔ چندبرچسبی را پشتیبانی می‌کند و نیاز به تغییر ساختار آن نیست؛ همچنین مقالات متعددی با استفاده از این نوع شبکه به نتایج بسیار خوبی از نظر دقت و سرعت رسیده‌اند. شبکهٔ عصبی چندلایهٔ پرسپترون دست‌کم از سه لایهٔ ورودی، خروجی و میانی[1] (مخفی) تشکیل شده‌است. لایه‌های میانی را می‌توان افزایش داد. در بیشتر موارد وجود یک لایهٔ میانی بهترین نتایج را دربرداشت. انواع توابع فعال‌ساز[2] برای مجموعه‌داده‌های متفاوت در نظر گرفته و بهترین نتایج با توابع فعال‌ساز relu در لایه‌های میانی و فعال‌ساز sigmoid در لایهٔ پایانی حاصل شد. تابع sigmoid احتمال عضویت در طبقه را برای برچسب پیش‌بینی می‌کند و چون این مقدار احتمال بین صفر و یک است، برای مقدار نهایی که یا صفر یا یک است (یک بیان‌کنندهٔ پیش‌بینی برچسب و صفر عدم پیش‌بینی آن برچسب به‌وسیلهٔ مدل را بیان می‌کند) نیاز به حد آستانه است؛ همان‌طورکه بیان شد، با نتایج به‌دست‌آمده، درمجموع حد آستانهٔ ۰.۳ نتایج بهتری را حاصل می‌کند.

در این مقاله با استفاده از خوشه‌بندی مطرح‌شده، مشکل برچسب‌هایی که نسبت پراکندگی بالایی داشتند، تا حدودی رفع شد. نتایج نشان می‌دهد، به‌طورکلی در مجموعه‌داده‌هایی که واریانس کمتر بود، نتایج بهتر است. در نمودار (۱) می‌توان این نسبت را مشاهده کرد.



(شکل-۳): ارتباط دقت و cvir
(figure-3): relation between accuracy and cvir

محور افقی در نمودار بیان‌کنندهٔ اختلاف دقت الگوریتم پیشنهادی از میانگین دقت سایر الگوریتم‌هاست و محور عمودی cvir واریانس طبقه‌های موجود را در مجموعه‌داده بیان می‌کند. ضریب تغییر نسبت عدم تعادل (CVIR) یک معیار آماری است که برای سنجش میزان عدم تعادل در یک مجموعه‌دادهٔ طبقه‌بندی استفاده می‌شود.

روند کلی نمودار، بیان‌کنندهٔ آن است که با کاهش cvir اختلاف از میانگین بیشتر شده و الگوریتم پیشنهادی عملکرد بهتری دارد.

۵- نتایج

این مقاله روش جدیدی را با هدف بهبود دقت وظایف طبقه‌بندی چندبرچسبی معرفی می‌کند. رویکرد پیشنهادی دو نوآوری کلیدی را به‌ویژه در زمینه‌های خوشه‌بندی برچسب و کاهش ابعاد لایهٔ شبکهٔ عصبی، ادغام می‌کند؛ این نوآوری‌ها برای رفع چالش‌های رایج در طبقه‌بندی چندبرچسبی طراحی شده‌اند که منجر به افزایش عملکرد و کاهش پیچیدگی محاسباتی می‌شود. نخستین نوآوری شامل طراحی یک سازوکار خوشه‌بندی برابر برای مدیریت مسئلهٔ برچسب‌های دنباله (برچسب‌های کمیاب)، نوآوری دوم حول محور کاهش ابعاد در لایه‌های شبکهٔ عصبی است. با توجه به پیچیدگی آموزش طبقه‌بندی‌کننده‌های متعدد در یک تنظیمات چندبرچسبی، کاهش ابعاد لایه‌ها برای بهینه‌سازی زمان آموزش بسیار مهم است. این روش از scatter_add برای این منظور استفاده می‌کند که به طور مؤثر تعداد پارامترها را بدون کاهش دقت، کاهش می‌دهد و روند آموزش را سرعت قابل‌توجهی می‌بخشد؛ علاوه‌بر این، راه حل ارائه‌شده در این مقاله می‌تواند به‌عنوان یک رویکرد جدید در نظر گرفته شود که فضا را برای بهینه‌سازی بیشتر فراهم می‌کند. به‌طورخاص، نوع و تعداد طبقه‌بندی‌کننده‌ها را می‌توان برای به‌دست‌آوردن نتایج بهتر تنظیم کرد. در این مطالعه، یک طبقه‌بندی‌کنندهٔ پرسپترون چندلایه (MLP) به‌کار گرفته شد، اما در حال حاضر از چندین الگوریتم پیشرفته در انواع مجموعه‌داده‌ها و معیارهای عملکرد بهتر عمل کرده‌است. این نشان می‌دهد که با طبقه‌بندی‌کننده‌های پیشرفته‌تر، روش پیشنهادی می‌تواند به دقت بالاتر و کاهش بیشتر در زمان آموزش دست یابد، که آن را از جهتی امیدوارکننده برای پژوهش‌های آینده در طبقه‌بندی چندبرچسبی تبدیل می‌کند. این مقاله از دو روش خوشه‌بندی استفاده می‌کند؛ بااین‌حال، هیچ معیار دقیق یا علمی برای تعیین زمان استفاده از هر نوع خوشه‌بندی ارائه نشده‌است. این موضوع حتی می‌تواند به‌عنوان یک مشکل پژوهشی جدید در نظر گرفته شود؛ علاوه‌براین، پارامترهای دیگری مانند تعداد خوشه‌ها و نوع طبقه‌بندی‌کننده نیز مشابه این موضوع است و می‌تواند هم به‌عنوان محدودیت‌های این مقاله و هم به‌عنوان فرصت‌های جدید برای سایر پژوهش‌گران در نظر گرفته شود.

۶-جمع‌بندی

در این مقاله سه نوآوری جدید در بخش‌های خوشه‌بندی برچسب‌ها و کاهش بُعد لایه‌های شبکهٔ عصبی ارائه شد. با توجه به این‌که از چندین طبقه‌بند برای پیش‌بینی برچسب‌ها

[1] Hidden layer
[2] Activation function

استفاده شده‌است، کاهش بُعد به سبک جمع پراکنده برای کاهش زمان آموزش در نظر گرفته شد. راه‌کار این مقاله را می‌توان یک رویکرد جدید در نظر گرفت و با تغییر نوع و تعداد طبقه‌بندها به نتایج بهتری دست یافت؛ چراکه در این مقاله از طبقه‌بند ساده mlp استفاده شده‌است؛ بااین‌حال در مجموعه‌داده‌های مختلف در بسیاری از پارامترها از بهترین الگوریتم‌های موجود، بهتر عمل کرده‌است.

7-References ۷-مراجع

[1]Han, Jiawei, Jian Pei, and Micheline Kamber, *Data mining: concepts and techniques*, Elsevier, 2011.

[2]Zhang, Min-Ling, and Zhi-Hua Zhou. "A review on multi-label learning algorithms", *IEEE transactions on knowledge and data engineering*, 26.8, 1819-1837, 2013.

[3]Chalkidis, Ilias, et al. "Large-scale multi-label text classification on EU legislation", *arXiv preprint arXiv*, 1906.02192, 2019.

[4]Spyromitros-Xioufis, Eleftherios, et al. "Multi-target regression via input space expansion: treating targets as inputs", *Machine Learning*, 104, 55-98, 2016.

[5]Yang, Qi, et al. "Amnn: Attention-based multimodal neural network model for hashtag recommendation", *IEEE Transactions on Computational Social Systems*, 7.3, 768-779, 2020.

[6]Lee, Jaesung, et al. "Compact feature subset-based multi-label music categorization for mobile devices", *Multimedia Tools and Applications*, 78, 4869-4883, 2019.

[7]Wang, Jiang, et al. "Cnn-rnn: A unified framework for multi-label image classification", *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[8]Khandagale, Sujay, Han Xiao, and Rohit Babbar, "Bonsai: diverse and shallow trees for extreme multi-label classification", *Machine Learning* 109 (11), 2099-2119, 2020.

[9]Tanaka, Erica Akemi, et al. "A multi-label approach using binary relevance and decision trees applied to functional genomics", *Journal of biomedical informatics*, 54, 85-95, 2015.

[10]Prajapati, Purvi, Thakkar, Amit, "Performance improvement of extreme multi-label classification using K-way tree construction with parallel clustering algorithm", *Journal of King Saud University-Computer and Information Sciences*, 34(8), 6354-6364, 2021.

[11]Prabhu, Yashoteja, et al. "Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising", *Proceedings of the 2018 World Wide Web Conference*, 2018, 993-1002.

[12]Zhang, Min-Ling, et al. "Binary relevance for multi-label learning: an overview", *Frontiers of Computer Science*, 12(2), 191-202, (2018).

[13]Jun, Xie, et al. "Conditional entropy based classifier chains for multi-label classification", *Neurocomputing*, 335, 185-194, 2019.

[14]Wang, Ran, et al. "Active k-labelsets ensemble for multi-label classification", *Pattern Recognition*, 109, 107583, (2021).

[15]Moyano, Jose M., et al. "Combining multi-label classifiers based on projections of the output space using Evolutionary algorithms", *Knowledge-Based Systems*, 196, 105770, 2020.

[16]Cerri, Ricardo, Rodrigo C. Barros, and André CPLF De Carvalho, "Hierarchical multi-label classification using local neural networks", *Journal of Computer and System Sciences*, 80.1, 39-56, 2014.

[17]Li, Junlong, et al. "Learning common and label-specific features for multi-Label classification with correlation information", *Pattern Recognition* , 121, 108259, 2022.

[18]Zhu, Xiaoyan, et al. "Dynamic ensemble learning for multi-label classification", *Information Sciences*, 623, 94-111, 2023.

[19]J. Huang, G. Li, Q. Huang, X. Wu, "Learning label specific features for multi-label classification", *IEEE ICDM 2015*, pp. 181–190, 2015.

[20]J. Huang, G. Li, Q. Huang, X. Wu, "Learning label-specific features and class dependent labels for multi-label classification", *IEEE Trans*. *Knowl. Data Eng*, 28 (12), 3309–3323, 2016.

[21]A. Braytee, W. Liu, A. Anaissi, P.J. Kennedy, "Correlated multi-label classification with incomplete label space and class imbalance", *ACM Trans. Intell. Syst. Technol*. 10 (5), 56:1–56:26, 2019.

[22]Y. Wang, W. Zheng, Y. Cheng, D. Zhao, "Joint label completion and label-specific features for multi-label learning algorithm", *Soft Comput*, 24 (11), 6553–6569, 2020.

[23]H. Han, M. Huang, Y. Zhang, X. Yang, W. Feng, "Multi-label learning with label specific features using correlation information", *IEEE Access 7*, 11474– 11484, 2017.

[24]X. Jia, S. Zhu, W. Li, "Joint label-specific features and correlation information for multi-label learning", *J. Comput. Sci. Technol*. 35 (2) (2020) 247–258

[۲۵] صامت عمرانی، مسلم، صنیعی آباده، محمد، مقدم چرکری، نصراله، «تشخیص شایعه در شبکه اجتماعی توییتر با استفاده از ویژگی‌های توییت و کاربر»، *فصلنامهٔ پردازش علائم و داده‌ها*، دورهٔ ۲۱، شمارهٔ ۲، صص ۱۵-۲۸، ۱۴۰۳.

[25] Moslem Samet Omrani, Mohammad Saniee Abadeh, Nasrollah Moghaddam Charkari, "Rumor Detection on Twitter using tweet and user features", *Signal and Data Processing*, 21(2), 15-28. 2024.

[۲۶] پروین نیا، الهام، صفری، محمد، خیامی، سید علیرضا، «تشخیص حالت غیر نرمال ماشین های دوار با داده کاوی در پارامترهای حفاظتی»، *فصلنامهٔ پردازش علائم و داده‌ها*، دورهٔ ۲۱، شمارهٔ ۱، صص ۲۷-۳۸، ۱۴۰۳.

[26] Elham Parvinnia, Mohammad Safari, Seyed Alireza Khayami, "Exploring on rotating machines abnormal state with data mining in protective parameters", *Signal and Data Processing*, 21(1), 27-38, 2024.

نگین دانشپور مدرک کارشناسی خود را در سال ۱۳۷۷ از دانشگاه شهید بهشتی و درجهٔ کارشناسی‌ارشد و دکترای خود را در رشتهٔ مهندسی کامپیوتر - نرم‌افزار از دانشگاه صنعتی امیرکبیر در سال‌های ۱۳۸۰ و ۱۳۸۹ دریافت کرد. در حال حاضر دانشیار دانشکدهٔ مهندسی کامپیوتر دانشگاه تربیت دبیر شهید رجایی است. زمینه‌های پژوهشی مورد علاقهٔ ایشان عبارت‌اند از: داده‌کاوی و پیش‌پردازش، مدیریت داده‌ها، پایگاه دادهٔ تحلیلی و سیستم‌های تصمیم یار.
نشانی رایانامهٔ ایشان عبارت است از:
ndaneshpour@sru.ac.ir

محسن نصیری کارشناسی را در دانشگاه شهید رجایی در رشتهٔ مهندسی نرم‌افزار گذرانده است. در حال حاضر دانشجوی مقطع کارشناسی‌ارشد رشتهٔ مهندسی نرم‌افزار دانشگاه شهید رجایی است. زمینهٔ مطالعاتی ایشان داده‌کاوی و پایگاه داده است.
نشانی رایانامهٔ ایشان عبارت است از:
Program.nasiri97@gmail.com