

## Multi-label feature selection based on competitive swarm optimization

Hamid Bayati<sup>1</sup>, Mohammad Bagher Dowlatshahi<sup>2\*</sup> and Mohssen Paniri<sup>3</sup>

1- Department of Computer Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran.

2- Department of Computer Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran.

3- Department of Computer Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran.

<sup>1</sup>hamidbayati74@gmail.com, <sup>2</sup>dowlatshahi.mb@lu.ac.ir, and <sup>3</sup>mohssen.p10@gmail.com

Corresponding author's address: Mohammad Bagher Dowlatshahi, Department of Computer Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran.

**Abstract-** Feature selection is one of the important preprocessing steps in data mining and machine learning, which is used to dimensionality reduction and selecting a subset of representative features. Feature selection can remove redundant and irrelevant features that can increase the accuracy of the machine learning tasks. In this paper, a novel embedded approach for the multi-label feature selection method using competitive Swarm Optimizer (CSO) is proposed. In this method, at first, of particles is generated, then the particles are divided into two equal groups and compete in pairs, the winners are moved to the next iteration and the losers learn from the winners, and at the end of each iteration, the objective function for all the particles is computed. To increase the convergence rate, half of the initial population is generated by the similarity between features and labels, and a local search method inspired by the gradient descent algorithm is applied to discover the local structure of data. Finally, based on the best particle, the feature selection is done. The performance of the proposed method is compared with six known and state-of-the-art multi-label feature selection methods. The experimental results on the image and text multi-label datasets show the efficiency and superiority of the proposed method in different multi-label evaluation measures criteria.

**Keywords-** Competitive swarm optimizer, Feature selection, Multi-label data, Reconstruction error, Sparse representation.

## انتخاب ویژگی چندبرچسبی با استفاده از الگوریتم بهینه‌ساز جمعیت رقابتی

حمید بیاتی<sup>۱</sup>، محمدباقر دولتشاهی<sup>۲</sup>، محسن پنییری<sup>۳</sup>

۱- دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه لرستان، خرم آباد، ایران.

۲- دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه لرستان، خرم آباد، ایران.

۳- دانشکده فنی و مهندسی، گروه مهندسی کامپیوتر، دانشگاه لرستان، خرم آباد، ایران.

<sup>1</sup>hamidbayati74@gmail.com, <sup>2</sup>dowlatshahi.mb@lu.ac.ir, <sup>3</sup>mohssen.p10@gmail.com

\* نشانی نویسنده مسئول: محمدباقر دولتشاهی، خرم آباد، کیلومتر ۵ جاده خرم آباد-تهران، دانشگاه لرستان، دانشکده فنی و مهندسی.

چکیده- انتخاب ویژگی یکی از اساسی‌ترین مراحل پیش پردازش داده‌ها در یادگیری ماشین است که با کاهش ابعاد در مجموعه داده‌ها باعث صرفه‌جویی در منابع و افزایش سرعت محاسبات می‌شود. همچنین، انتخاب ویژگی می‌تواند با حذف ویژگی‌های غیرمرتبط و افزونه باعث افزایش دقت و کارایی الگوریتم‌های یادگیری ماشین شود. در این مقاله، یک روش جدید تعبیه‌شده برای انتخاب ویژگی چندبرچسبی پیشنهاد شده است که در آن، مساله انتخاب ویژگی چندبرچسبی برای اولین بار با استفاده از الگوریتم بهینه‌ساز جمعیت رقابتی حل خواهد شد. در روش پیشنهادی، ابتدا جمعیتی از ذرات ساخته می‌شود، سپس ذرات به دو دسته مساوی تقسیم شده و به صورت جفت با هم رقابت می‌کنند، ذرات برنده به تکرار بعد منتقل شده و ذرات بازنده از برنده‌ها یاد می‌گیرند، و در انتهای هر تکرار تابع هدف برای همه ذرات محاسبه می‌شود. این فرایند تا پیدا شدن یک ذره نزدیک به بهینه (که در اینجا ذره‌ای است که کمترین مقدار ضرر را دارد)، ادامه پیدا می‌کند. در این روش برای افزایش سرعت همگرایی، نیمی از جمعیت اولیه با استفاده از یک معیار شباهت ساخته می‌شود و همچنین از یک جستجوگر محلی برای کشف خاصیت محلی داده‌ها استفاده می‌شود. در انتها، بر اساس بهترین ذره، فرآیند انتخاب ویژگی انجام می‌شود. نتایج پیاده‌سازی روش پیشنهادی و مقایسه آن با نتایج سایر الگوریتم‌های انتخاب ویژگی چند برچسبی نمایانگر کارایی مناسب الگوریتم پیشنهادی است.

واژه‌های کلیدی: الگوریتم بهینه‌ساز جمعیت رقابتی، انتخاب ویژگی، داده‌های چندبرچسبی، خطای باز تولید، بازنمایی تُنک.

### ۱- مقدمه

الگوریتم‌ها می‌شوند [۲]. یک روش برای اجتناب از اثر نحوست ابعاد<sup>۱</sup>، روش کاهش ابعاد است که اغلب قبل از اجرای الگوریتم‌های داده‌کاوی بر روی داده‌های با ابعاد بالا اجرا می‌شود. به طور کلی روش‌های کاهش ابعاد به سه دسته تقسیم می‌شوند که عبارتند از: (۱) انتخاب ویژگی؛ (۲) یادگیری زیرفضا؛ و (۳) روش‌های ترکیبی [۳]. انتخاب ویژگی<sup>۵</sup> یکی از مراحل پیش پردازش در داده کاوی، یادگیری ماشین و خوشه بندی [۴] است که به وسیله‌ی حذف کردن ویژگی‌های نامرتبط و افزونه، ابعاد

در سال‌های اخیر داده‌های با ابعاد بسیار بالا<sup>۱</sup> در زمینه‌های داده کاوی، یادگیری ماشین، پردازش تصویر و غیره به وجود آمده‌اند [۱]. داده‌های با ابعاد بالا دارای مشکلاتی از قبیل اطلاعات نامرتبط<sup>۲</sup>، افزونگی<sup>۳</sup> و اطلاعات نویزی هستند که این مشکلات منجر به پردازش اطلاعات بی-فایده، افزایش زمان پردازش اطلاعات و افزایش پیچیدگی محاسباتی

نتایج قابل قبولی نیز رسیده‌اند [۱۵] و [۱۶]. این الگوریتم‌ها به دو دسته الگوریتم‌های هوش جمعی مانند (الگوریتم کلونی مورچگان<sup>۱۵</sup> [۱۷] و الگوریتم جستجوی گرانشی<sup>۱۶</sup> [۱۸] و [۱۹]) و الگوریتم‌های تکاملی (مانند الگوریتم ژنتیک [۲۰]) تقسیم می‌شوند. الگوریتم‌های مبتنی بر جمعیت به دلیل توانایی جستجوی سراسری و انطباق با فضاهای مختلف جستجو، نتایج قابل قبولی را در فرآیند انتخاب ویژگی از خود نشان داده‌اند [۲۱].

در این مقاله یک روش تعبیه‌شده جدید برای انتخاب ویژگی نظارتی چند برچسبی بر اساس الگوریتم هوش جمعی بهینه‌ساز جمعیت رقابتی<sup>۱۷</sup> تک هدفه [۲۲] پیشنهاد شده است. در این الگوریتم، ابتدا تعدادی ذره<sup>۱۸</sup> ساخته می‌شوند، سپس ذرات به صورت تصادفی دو به دو با یکدیگر به رقابت پرداخته و نیمی از آنها برنده و نیم دیگر بازنده می‌شوند. در مرحله بعد بر اساس تابع هدف، ضرر هر کدام از ذرات محاسبه می‌شود. الگوریتم تا رسیدن به بهترین ذره به صورت تکراری ادامه پیدا می‌کند. در نهایت ذره‌ای که کمترین ضرر را دارد، برای فرآیند انتخاب ویژگی استفاده می‌شود. به منظور افزایش سرعت همگرایی نیمی از جمعیت با استفاده از شباهت بین ویژگی‌ها و برچسب کلاس‌ها تولید می‌شوند و برای کشف ویژگی محلی داده‌ها از یک جستجوگر محلی که از الگوریتم گرادیان کاهشی الهام گرفته است، استفاده می‌کنیم.

نوآوری‌های روش پیشنهادی به شرح زیر می‌باشند:

- تاجایی که نویسندگان اطلاع دارند، برای اولین بار است که به طور همزمان از بازنمایی تنک و یک الگوریتم هوش جمعی برای حل مسئله انتخاب ویژگی چندبرچسبی استفاده می‌شود. برای این منظور از الگوریتم بهینه‌ساز جمعیت رقابتی برای یافتن یک بازنمایی تنک استفاده شده است.
- علاوه بر جستجوی سراسری<sup>۱۹</sup> که توسط الگوریتم بهینه‌ساز جمعیت رقابتی انجام می‌شود، یک الگوریتم گرادیان کاهشی به عنوان جستجوگر محلی برای کشف خاصیت محلی داده‌های چندبرچسبی استفاده شده است. تاجایی که نویسندگان اطلاع دارند، برای اولین بار است که از الگوریتم گرادیان کاهشی به عنوان جستجوگر محلی در حل مسئله انتخاب ویژگی چندبرچسبی استفاده می‌شود.
- برای اثبات برتری و اثربخشی روش پیشنهادی نسبت به شش روش شناخته‌شده انتخاب ویژگی چندبرچسبی، آزمایشات گسترده‌ای بر روی شش مجموعه داده در حوزه متن و تصویر انجام شده است.

ادامه مقاله به این صورت سازماندهی شده است. در بخش ۲ برخی کارهای مرتبط را مرور می‌کنیم. در بخش ۳، مفاهیم پایه را شرح می‌دهیم. در بخش ۴، الگوریتم پیشنهادی برای انتخاب ویژگی نظارتی چند برچسبی شرح داده می‌شود. نتایج الگوریتم پیشنهادی همراه با مقایسات انجام شده با سایر الگوریتم‌های رقیب در بخش ۵ بررسی می‌شوند. در

داده‌ها را کاهش می‌دهد. به طور کلی، فرآیند انتخاب ویژگی باعث افزایش دقت، درک بهتر داده‌ها، کاهش پیچیدگی محاسباتی، و جلوگیری از بیش برآزش<sup>۱</sup> می‌شود. به طور خلاصه، هدف اصلی فرآیند انتخاب ویژگی ارائه زیر مجموعه‌ای از ویژگی‌های بهینه است که دقت الگوریتم‌های داده کاوی و یادگیری ماشین را افزایش داده و همچنین اثر داده‌های نامرتب و نویزی را کاهش دهد [۵].

روش‌های انتخاب ویژگی بر اساس وجود و یا عدم وجود برچسب کلاس به سه دسته‌ی غیرنظارتی<sup>۷</sup>، نیمه‌نظارتی<sup>۸</sup> و نظارتی<sup>۹</sup> تقسیم می‌شوند. در روش‌های غیرنظارتی به دلیل عدم وجود برچسب کلاس، اهمیت ویژگی‌ها با توجه به وجود خواص ذاتی ویژگی‌ها نسبت به یکدیگر مانند واریانس، همبستگی<sup>۱۰</sup> و دیگر معیارها مشخص می‌شود [۶]. در روش‌های نیمه‌نظارتی، تعداد زیادی از نمونه‌ها فاقد برچسب و تعدادی دارای برچسب هستند. این روش‌ها ابتدا ساختار ذاتی نمونه‌های برچسب زده شده را برای ساختن یک مدل پایه یاد می‌گیرند و سپس از نمونه‌های بدون برچسب برای بهبود مدل پایه استفاده می‌کنند [۷]. در روش‌های نظارتی به دلیل وجود برچسب کلاس، اغلب ویژگی‌ها بر اساس ارتباط بین ویژگی‌ها و برچسب کلاس (برای مثال با توجه به همبستگی بین ویژگی‌ها و برچسب کلاس‌ها) انتخاب می‌شوند [۸]. در روش‌های نظارتی تک برچسبی، هر نمونه دارای یک برچسب کلاس<sup>۱۱</sup> است، در حالیکه در روش‌های نظارتی چند برچسبی<sup>۱۲</sup> یک نمونه به طور همزمان می‌تواند بیش از یک برچسب کلاس داشته باشد. با افزایش تعداد مجموعه داده‌های چند برچسبی مانند مجموعه داده‌های بیوانفورماتیک، نوشتاری، تصویر و غیره، یادگیری چندبرچسبی به یک موضوع پژوهشی تازه تبدیل شده است [۹] و [۱۰].

در یک تقسیم‌بندی دیگر، روش‌های انتخاب ویژگی بر اساس نحوه تعامل‌شان با الگوریتم‌های یادگیری ماشین و داده کاوی، به سه دسته فیلتر، پوشه<sup>۱۳</sup> و تعبیه‌شده<sup>۱۴</sup> تقسیم می‌شوند [۱۱]. روش‌های فیلتر مستقل از الگوریتم‌های یادگیری ماشین و داده کاوی و بر اساس اندازه‌گیری معیارهایی مانند مرتبط بودن، افزونگی و غیره به هر ویژگی یک امتیاز اختصاص داده و در نهایت ویژگی‌هایی با امتیاز بالاتر انتخاب می‌شوند. روش‌های فیلتر به دلیل پیچیدگی محاسباتی پایین، اغلب در بسیاری از حوزه‌ها استفاده می‌شوند [۱۲]. در روش‌های پوشه، در هر تکرار زیرمجموعه‌ای از ویژگی‌ها انتخاب شده، سپس توسط الگوریتم‌های یادگیری ماشین تا رسیدن به یک عملکرد مطلوب یا معیار توقف، ارزیابی می‌شوند. این روش‌ها به دلیل استفاده از الگوریتم‌های یادگیری ماشین، دارای پیچیدگی محاسباتی زیادی بوده و بر روی مجموعه‌داده‌هایی با ابعاد بالا بسیار کند هستند [۱۳]. در روش‌های تعبیه‌شده، فرآیند انتخاب ویژگی و یادگیری مدل با یکدیگر ترکیب می‌شوند. در این روش‌ها، زیرمجموعه ویژگی‌ها حین فرآیند آموزش در الگوریتم‌های یادگیری ماشین انتخاب می‌شود [۱۴].

در سال‌های اخیر، الگوریتم‌های مبتنی بر جمعیت به طور گسترده‌ای برای حل مشکلات بهینه‌سازی ترکیباتی مورد استفاده قرار گرفته و به

نهایت، در بخش ۶ به نتیجه‌گیری پرداخته خواهد شد.

برچسب کلاس‌ها است) استفاده می‌شود. سرانجام، بر اساس دو معیار بالا بهترین مجموعه ویژگی‌ها انتخاب می‌شود [۲۸].

## ۲- کارهای مرتبط

در این بخش به تشریح برخی از مهم‌ترین کارهای انجام شده در زمینه انتخاب ویژگی نظارتی چندبرچسبی می‌پردازیم:

- الگوریتم PMU: در این روش، یک زیر مجموعه خوب از ویژگی‌ها به وسیله یک الگوریتم تکراری با بیشینه کردن اطلاعات متقابل چند متغیره بین ویژگی‌های انتخاب شده و برچسب کلاس‌ها انتخاب می‌شود [۲۳].

- الگوریتم‌های ReliefF-ML، PPT-ReliefF، و RRReliefF: این الگوریتم‌ها هر کدام یک نسخه گسترش یافته از روش انتخاب ویژگی تک برچسب ReliefF هستند. در این الگوریتم‌ها، داده‌های چند برچسبی به داده‌های تک برچسب تبدیل شده و سپس با استفاده از الگوریتم ReliefF وزن هر ویژگی مشخص می‌شود. این الگوریتم‌ها در نوع انتقال داده‌ها از فضای اصلی به فضای تک برچسب با یکدیگر تفاوت دارند [۲۴].

- در سال ۲۰۱۴، Ying Yu و همکاران یک الگوریتم انتخاب ویژگی نظارتی چند برچسبی ارائه دادند، این الگوریتم شامل دو مرحله است، ابتدا اهمیت هر ویژگی نسبت به برچسب کلاس‌ها توسط اطلاعات متقابل مشخص می‌شود، سپس با استفاده از الگوریتم ژنتیک زیرمجموعه ویژگی‌های بهینه انتخاب می‌شود [۲۵].

- الگوریتم MCLS<sup>۲۱</sup>: در این روش، ابتدا برچسب کلاس‌ها از فضای باینری به یک فضای اقلیدسی انتقال داده شده و به این ترتیب شباهت بین نمونه‌ها توسط برچسب‌های عددی محدود می‌شود. سپس، اهمیت هر ویژگی با در نظر گرفتن برچسب‌های عددی و خصوصیات محلی داده‌ها مشخص خواهد شد [۲۶].

- الگوریتم ELA<sup>۲۲</sup>: این الگوریتم با توجه به معیار آنتروپی، به برچسب کلاس‌ها وزن‌های مختلفی اختصاص می‌دهد. آنتروپی برای تعیین عدم قطعیت یک نمونه در یک مجموعه داده چند برچسبی استفاده می‌شود، بطوریکه هرچه عدم قطعیت یک نمونه بیشتر باشد، احتمال تعلق آن به یک کلاس خاص کمتر می‌شود [۲۷].

- الگوریتم LRFS<sup>۲۳</sup>: در این الگوریتم ابتدا برچسب کلاس‌ها به دو دسته مستقل و وابسته تقسیم می‌شوند، سپس برای محاسبه افزونگی برچسب کلاس‌ها، تفاوت بین این دو گروه بررسی می‌شود. از طرف دیگر برای محاسبه میزان مرتبط بودن ویژگی‌ها، از روش اطلاعات متقابل شرطی بین ویژگی-های نامزد و هر برچسب کلاس (که نماینده مجموعه ای از

## ۳- مفاهیم پایه

در این بخش، برای درک بهتر روش پیشنهادی مفاهیم پایه در رابطه با روش پیشنهادی بیان می‌شوند، به شکلی که در ابتدا یادگیری چند برچسبی را شرح می‌دهیم، سپس الگوریتم بهینه ساز جمعیت رقابتی و در نهایت بازنمایی تک را بررسی می‌کنیم.

### ۳-۱- یادگیری چندبرچسبی

با فرض اینکه  $X$  یک مجموعه داده شامل  $n$  نمونه و  $d$  ویژگی باشد، هر نمونه توسط یک بردار ویژگی  $x = \{x_1, x_2, \dots, x_d\} \in X$  و یک زیرمجموعه از ویژگی‌ها  $L \subset Y$  مشخص می‌شود، که  $Y = \{y_1, y_2, \dots, y_l\}$  یک مجموعه از  $l$  برچسب کلاس است. شکل ۱ نمایشی کلی از یک مجموعه داده‌ی چند برچسبی است. در این سناریو، دسته-بندی‌کننده چند برچسبی یک مجموعه از برچسب‌ها را  $L \subset Y$  برای نمونه‌های مشاهده نشده  $x \in X$  پیش‌بینی می‌کند [۵] و [۲۹].

X				Y			
$x_1$	$x_2$	...	$x_d$	$y_1$	$y_2$	...	$y_l$
$x_{11}$	$x_{12}$	...	$x_{1d}$	0	0	1	0
$x_{21}$	$x_{22}$	...	$x_{2d}$	1	1	1	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{n1}$	$x_{n2}$	...	$x_{nd}$	1	1	0	0

شکل ۱: نمایش داده‌های چند برچسبی

### ۳-۲- الگوریتم بهینه‌ساز جمعیت رقابتی

الگوریتم بهینه‌ساز جمعیت ذرات<sup>۲۴</sup> با الهام گرفتن از رفتار اجتماعی دسته‌های پرندگان، مدل شده است [۳۰]. این الگوریتم یک روش محاسباتی است که به صورت تکراری سعی در بهبود یک راه‌حل کاندیدا (یک ذره) با توجه به یک معیار مشخص از کیفیت را دارد. در این روش، یک جمعیت از راه‌حل‌های کاندیدا (جمعیت ذرات) وجود دارد که این ذرات با توجه به فرمول ریاضی بر اساس موقعیت<sup>۲۵</sup> و سرعت<sup>۲۶</sup> ذرات در فضای جستجو حرکت می‌کنند. حرکت هر ذره تحت تاثیر بهترین موقعیت محلی شناخته شده خود ذره است، با این حال به سمت بهترین موقعیت‌های شناخته شده در فضای جستجو هم هدایت می‌شود که با یافتن موقعیت‌های بهتر توسط سایر ذرات، به روز رسانی می‌شود [۱۴]. یکی از مشکلات بزرگ این الگوریتم کار کردن ضعیف آن با داده‌های با ابعاد زیاد است. برای حل این مشکل، الگوریتم بهینه‌ساز جمعیت رقابتی ارائه شده است [۲۲]. این الگوریتم بسیار شبیه به الگوریتم بهینه‌ساز جمعیت ذرات است با این تفاوت که در هر تکرار به طور تصادفی ذرات به دو دسته تقسیم می‌شوند و به صورت دو به دو با یکدیگر به رقابت

عناصر پایه است که از یک فرهنگ لغت کامل انتخاب می‌شوند. با این فرض که  $\gamma$  سیگنالی است که انتظار می‌رود ساخته شود و  $A$  فرهنگ لغت باشد، مفهوم بازنمایی  $\text{تُنک}$  به شکل زیر مدل می‌شود:

$$\min_w \|y - Aw\|_F^2 + \lambda \|w\|_{1,1} \quad (5)$$

که در آن،  $w$  یک بردار  $\text{تُنک}$  است و  $\lambda$  پارامتری برای ایجاد تعادل بین خطای بازسازی  $\|y - Aw\|_F^2$  و محاسبه  $\text{تُنکی}$   $\|w\|_{1,1}$  است.

#### ۴- روش پیشنهادی

در این بخش، روش انتخاب چند برچسبی با استفاده از الگوریتم جمعیت رقابتی صورت کامل شرح داده می‌شود. در ابتدا توضیح می‌دهیم که چگونه جمعیت اولیه تولید می‌شود، سپس تابع برازندگی با توجه به کمینه کردن خطای بازسازی و بازنمایی  $\text{تُنک}$  طراحی شده و در نهایت الگوریتم بهینه ساز محلی گرادبان کاهش می‌دهد برای داده‌های چندبرچسبی معرفی می‌شود.

#### ۴-۱- مفاهیم اولیه

در این بخش به توضیح مفاهیم اولیه مقاله می‌پردازیم. در این مقاله مجموعه داده را  $X \in \mathbb{R}^{n \times d}$  نمایش می‌دهیم که در آن تعداد نمونه‌ها و تعداد ویژگی‌ها به ترتیب با  $n$  و  $d$  نمایش داده می‌شوند. برچسب کلاس‌ها با  $Y \in \mathbb{R}^{n \times l}$  نمایش داده می‌شود که  $l$  نشان‌دهنده تعداد برچسب کلاس‌ها می‌باشد. هر عضو از جمعیت (یک ذره) توسط  $W \in \mathbb{R}^{d \times l}$  نمایش داده می‌شود.

#### ۴-۲- تولید جمعیت اولیه

در این مقاله برای یک مجموعه داده با  $d$  ویژگی و  $l$  برچسب کلاس، هر ذره توسط  $W \in \mathbb{R}^{d \times l}$  نمایش داده می‌شود. به منظور افزایش سرعت همگرایی و بهبود دقت الگوریتم پیشنهادی، نیمی از جمعیت اولیه را با استفاده از معیار شباهت کوسینوسی  $\gamma$  تولید می‌کنیم. همچنین به منظور جلوگیری از همگرایی زودرس و پوشش کل فضای جستجو، نیم دیگر جمعیت اولیه به صورت تصادفی تولید می‌شود. معیار شباهت کوسینوسی یکی از محبوب‌ترین معیارهای شباهت بین دو بردار است. فرض کنید  $x_1$  اولین بردار ویژگی از  $X$  و  $y_1$  اولین بردار برچسب از  $Y$  باشد، بنابراین میزان شباهت کوسینوسی بین دو بردار  $x_1$  و  $y_1$  به شکل زیر محاسبه می‌شود:

$$\text{cosine}(x_1, y_1) = \frac{x_1 \cdot y_1}{\|x_1\| \|y_1\|} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}, \quad (6)$$

که  $x_1$  و  $y_1$  دو بردار با تعداد درایه  $n$  (تعداد نمونه‌ها) هستند (شکل ۱ را

می‌پردازند. بعد از هر رقابت، ذره برنده مستقیماً به تکرار بعد منتقل می‌شود، در حالی که ذره بازنده موقعیت و سرعت خود را با یادگیری از ذره برنده به روز رسانی می‌کند.

در الگوریتم بهینه‌ساز جمعیت رقابتی، هر ذره در یک فضای جستجوی  $d$  بعدی با  $P_i^t = (\bar{X}_i(t), \bar{V}_i(t))$  نشان داده می‌شود که  $\bar{X}_i(t)$  و  $\bar{V}_i(t)$  به ترتیب بردار موقعیت و سرعت ذره  $i$  در تکرار  $t$  از الگوریتم بوده و به شکل زیر نمایش داده می‌شوند:

$$\bar{X}_i(t) = (x_i^1(t), x_i^2(t), \dots, x_i^q(t), \dots, x_i^d(t)), \quad (1)$$

$$\bar{V}_i(t) = (v_i^1(t), v_i^2(t), \dots, v_i^q(t), \dots, v_i^d(t)), \quad (2)$$

که در آن  $i$  از ۱ تا  $N$  بوده جایی که  $N$  نمایش دهنده تعداد اعضای جمعیت است. در این الگوریتم، ذره برنده با نماد  $P_W^t = (\bar{X}_W(t), \bar{V}_W(t))$  و ذره بازنده با نماد  $P_L^t = (\bar{X}_L(t), \bar{V}_L(t))$  نشان داده می‌شوند و سرعت هر ذره بازنده در  $q$  بعد به شکل زیر به روز رسانی می‌شود:

$$v_L^q(t+1) = r_1 \times v_L^q(t) + r_2 \times (x_W^q(t) - x_L^q(t)) + \varphi \times r_3 \times (x_M^q(t) - x_L^q(t)), \quad (3)$$

که در آن  $v_L^q(t)$  و  $v_L^q(t+1)$  به ترتیب سرعت بعدی و سرعت فعلی ذره بازنده در بعد  $d$ ،  $r_1, r_2, r_3$  سه عدد تصادفی در بازه  $[0,1]$ ،  $x_W^q(t)$  به ترتیب نمایش دهنده موقعیت فعلی ذره برنده و ذره بازنده،  $x_M^q(t)$  نشان دهنده میانگین موقعیت جمعیت در تکرار  $t$  و  $\varphi$  یک پارامتر برای تعیین تاثیر  $x_M^q(t)$  در به روز رسانی بردار سرعت هستند.

موقعیت بعدی ذره (که با نماد  $x_L^q(t)$  نشان داده می‌شود) به شکل زیر به روز رسانی می‌شود:

$$x_L^q(t+1) = x_L^q(t) + v_L^q(t+1), \quad (4)$$

که در آن  $x_L^q(t)$  و  $v_L^q(t+1)$  به ترتیب نمایش دهنده موقعیت فعلی و سرعت بعدی ذره هستند [۵].

#### ۳-۳- بازنمایی $\text{تُنک}$

در دهه گذشته، مفهوم بازنمایی  $\text{تُنک}$   $\gamma$  تاثیر زیادی بر روی یادگیری ماشین و تشخیص الگو گذاشته است و یکی از ابزارهای قدرتمند کار کردن با داده‌های ابعاد بالا است. این روش برای اولین بار در زمینه بازسازی سیگنال‌ها استفاده شد. سیگنال ورودی یک ترکیب خطی از

کاهش به عنوان جستجوگر محلی در روش پیشنهادی استفاده شده است.

در هر تکرار از الگوریتم پیشنهادی، گرادیان کاهش تنها یک مرحله و بر روی نیمی از ذرات به صورت تصادفی، به شکل زیر اعمال می‌شود:

$$W_{new} = W - \alpha \nabla f(W), \quad (9)$$

که در آن  $\alpha$  نرخ یادگیری،  $W \in \mathbb{R}^{d \times l}$  ماتریس انتقال،  $W_{new} \in \mathbb{R}^{d \times l}$  ماتریس انتقال جدید (که توسط گرادیان کاهش به روز رسانی شده است)، و  $f(W)$  قسمت مشتق پذیر تابع هزینه (یعنی تابع خطای بازسازی) است که به شکل زیر نوشته می‌شود:

$$f(W) = \|Y - XW\|_F^2 = \sum_{i=1}^n \sum_{j=1}^l (K_{ij} - y_{ij})^2, \quad (10)$$

که در آن

$$K_{ij} = \sum_{k=1}^d x_{ik} W_{kj}.$$

همچنین گرادیان کاهش ماتریس  $f(W)$  یعنی  $\nabla f(W)$  توسط فرمول زیر به دست می‌آید:

$$\nabla f(W) = \begin{bmatrix} \frac{\partial f(W)}{\partial W_{11}} & \dots & \frac{\partial f(W)}{\partial W_{1l}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(W)}{\partial W_{d1}} & \dots & \frac{\partial f(W)}{\partial W_{dl}} \end{bmatrix}, \quad (11)$$

$$\begin{aligned} \frac{\partial f(W)}{\partial W_{kj}} &= 2 \sum_{i=1}^n (K_{ij} - y_{ij}) \frac{\partial K_{ij}}{\partial W_{kj}} \\ &= 2 \sum_{i=1}^n (K_{ij} - y_{ij}) (x_{ik} W_{kj}). \end{aligned} \quad (12)$$

#### ۴-۵- انتخاب ویژگی

بعد از پیدا کردن بهترین ذره توسط الگوریتم، با بیشینه گرفتن از هر سطر این ذره، بزرگترین مقدار هر سطر که نشان دهنده بالاترین اهمیت آن ویژگی برای یکی از برچسب کلاس‌ها است، مشخص می‌شود. در نهایت، یک بردار با  $d$  درایه در اختیار داریم که اهمیت و وزن هر ویژگی را نمایش می‌دهد، و بعد از مرتب سازی نزولی این بردار، ویژگی‌ها بر اساس اهمیت‌شان مرتب خواهند شد.

شبه کد الگوریتم پیشنهادی در الگوریتم ۱ آمده است. همچنین چهارچوب کلی روش پیشنهادی را در شکل ۲ می‌توانید مشاهده کنید.

مشاهده کنید). مقدار حاصل از شباهت کوسینوسی بین ۱- و ۱ است، که مقدار ۱ به معنای دقیقاً یکسان بودن دو بردار، مقدار ۱- به معنای دقیقاً برعکس بودن دو بردار، و مقدار صفر نشان دهنده‌ی متعامد بودن دو بردار است.

#### ۴-۳- تابع هزینه

برای اینکه روش‌های فرا ابتکاری بتوانند راه‌حل‌های خوبی را در فضای جستجو پیدا کنند، می‌بایست توانایی ارزیابی راه‌حل‌ها را داشته باشند. برای این منظور روش‌های فرا ابتکاری با استفاده از یک تابع برازندگی، عملکرد هر یک از راه‌حل‌ها در فضای جستجو را ارزیابی می‌کنند و کیفیت هر یک از راه‌حل‌ها با استفاده از یک مقدار عددی مشخص می‌شود. تابع برازندگی باید توانایی تشخیص نزدیک‌ترین راه‌حل به راه‌حل بهینه را داشته باشد. بر اساس فرمول بازنمایی تُنک در معادله ۵، تابع هزینه (برازندگی) برای روش پیشنهادی به شکل زیر مدل می‌شود:

$$\min_W \|Y - XW\|_F^2 + \lambda \|W\|_{1,1}, \quad (7)$$

که در آن  $Y$  برچسب کلاس‌ها،  $X$  مجموعه داده،  $W$  یک عضو از جمعیت (ذره) و  $\lambda$  ضریب تنظیم است. تابع برازندگی فوق شامل دو قسمت است. هدف از قسمت اول که به خطای بازسازی<sup>۲۶</sup> معروف است، یادگیری یک ماتریس انتقال برای بازسازی داده‌های خروجی توسط داده‌های ورودی و برعکس، با کمترین خطای ممکن است. قسمت دوم یک تابع تنظیم است که تُنک بودن ماتریس انتقال  $W$  را اندازه‌گیری می‌کند که به شکل زیر تعریف می‌شود:

$$\|W\|_{1,1} = \sum_{i,j} |w_{i,j}|. \quad (8)$$

پس از یادگیری مدل بالا، بهینه‌ترین ماتریس انتقال  $W$  پیدا شده و سپس فرآیند انتخاب ویژگی بر اساس این ماتریس قابل انجام است. با توجه به اینکه در روش پیشنهادی ویژگی‌ها به عنوان داده‌ی ورودی و برچسب کلاس‌ها به عنوان داده‌ی خروجی در نظر گرفته می‌شوند، هدف پیدا کردن بهینه‌ترین ماتریس انتقال  $W$  برای نگاشت ویژگی‌ها به برچسب کلاس‌ها با کمترین خطای ممکن است. از طرفی به دلیل اینکه ماتریس انتقال  $W$  یک ماتریس  $d \times l$  است، هر ردیف از ماتریس انتقال وزن ویژگی  $q$  برای برچسب کلاس  $p$  را نشان می‌دهد. در نهایت بیشترین وزن هر ویژگی (بیشترین مقدار هر ردیف)، وزن آن ویژگی محسوب می‌شود زیرا ویژگی  $q$  برچسب  $p$  را از بقیه برچسب کلاس‌ها متمایز می‌کند.

#### ۴-۴- بهینه ساز محلی

یکی از مزایای روش پیشنهادی نسبت به سایر روش‌های هوش جمعی و تکاملی برای انتخاب ویژگی، کشف خاصیت محلی داده‌ها با استفاده از یک الگوریتم جستجوگر محلی است. در این مقاله، از الگوریتم گرادیان

## ۵- نتایج آزمایشات

این مجموعه داده‌ها حاوی پست‌های متنی بوده که در آن سطرها نمایش دهنده پست‌ها و هر ستون فرکانس کلمه را نشان می‌دهد. مشخصات این مجموعه داده‌ها در جدول ۱ آمده است.

جدول ۱: خصوصیات مجموعه داده‌های استفاده شده

مجموعه داده	تعداد نمونه‌ها	تعداد ویژگی‌ها	تعداد برچسب کلاس‌ها	دامنه
Corel5k	۵۰۰۰	۴۹۹	۳۷۴	تصویر
Scene	۲۴۰۷	۲۹۴	۶	تصویر
Chemistry	۶۹۶۱	۷۱۵	۱۷۵	نوشتاری
Image	۲۰۰۰	۲۹۴	۵	تصویر
CS	۹۲۷۰	۹۰۹	۲۷۴	نوشتاری
Cooking	۱۰۴۹۱	۹۷۷	۴۰۰	نوشتاری

## ۵-۲- دسته‌بند و معیارهای ارزیابی

ML-KNN یک نسخه چندبرچسبی از KNN است. در ML-KNN، برای هر داده تست ابتدا همسایگانش شناسایی می‌شود و سپس مجموعه برچسب کلاس‌ها براساس اطلاعات نزدیکترین همسایه‌ها، تعداد همسایگان متعلق به هر کلاس ممکن و روش حداکثر خلفی<sup>۳۰</sup> پیش بینی می‌شود [۳۲].

برای اندازه‌گیری کارایی زیر مجموعه ویژگی‌های انتخاب شده توسط روش پیشنهادی و سایر روش‌ها، از سه معیار متداول ارزیابی چند برچسبی، دقت، یک-خطا<sup>۳۱</sup> و میانگین صحت<sup>۳۲</sup> استفاده شده است.

فرض کنید  $T = \{(x_i, Y_i) : i = 1, \dots, n\}$  داده‌های تست باشد،  $h$  نمایش دهنده دسته‌بند باشد، که در آن  $Y_i \subseteq L$  زیر مجموعه برچسب‌های صحیح و  $L = \{y_q : q = 1, \dots, l\}$  مجموعه همه برچسب‌های کلاس‌ها است. برای هر نمونه  $x_i$ ، مجموعه برچسب‌های پیش‌بینی شده توسط دسته‌بند با  $Z_i$  نمایش داده می‌شود، و  $f(\cdot; \cdot)$  به‌عنوان یک تابع پیوسته است که توسط اغلب سیستم‌های یادگیری چندبرچسبی بازگردانده می‌شود. هر یک از معیارهای ارزیابی به شکل زیر محاسبه می‌شوند:

معیار دقت، درصد برچسب‌های پیش‌بینی شده درست را نسبت به کل برچسب‌ها اندازه‌گیری می‌کند و به شکل زیر تعریف می‌شود:

$$Accuracy(h, T) = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (13)$$

معیار یک-خطا، تعداد دفعاتی که برچسب رتبه برتر به مجموعه برچسب‌های نمونه مربوط نیست را ارزیابی می‌کند. معیار یک-خطا به شکل زیر تعریف می‌شود:

$$1 - Error = \frac{1}{n} \sum_{i=1}^n \left[ \left[ \arg \max_{y \in Y_i} \{f(x_i, y)\} \right] \notin Y_i \right] \quad (14)$$

در این بخش عملکرد روش پیشنهادی خود را در مقایسه با شش روش انتخاب ویژگی نظارتی چند برچسبی شناخته شده ELA\_CHI، PPT\_MI، PPT\_Relief، LRFS، MCLS و بررسی می‌کنیم. این آزمایشات بر روی چند مجموعه داده چند برچسبی شناخته شده برای حذف ویژگی‌های زائد و افزونه انجام می‌گیرد.

الگوریتم ۱: الگوریتم انتخاب ویژگی نظارتی چند برچسبی با بهینه‌ساز جمعیت رقابتی

**ورودی:** مجموعه داده  $X \in \mathbb{R}^{n \times d}$ ، ماتریس برچسب کلاس‌ها  $Y \in \mathbb{R}^{n \times l}$ ، تعداد اعضای جمعیت  $m$ ، حداکثر تعداد ارزیابی تابع برازندگی  $maxfe$  و ضریب  $\lambda$ .

**شروع:**

تولید جمعیت اولیه با توجه به بخش ۲-۳

محاسبه تابع برازندگی برای اعضای جمعیت توسط رابطه (۵)

**حلقه اصلی:**

$$FES = 0$$

**تا زمانی که**  $FES < maxfe$

تولید جفت‌های تصادفی

اجرای رقابت بین جفت‌ها

یادگیری بازنده‌ها از برنده‌ها با توجه به عبارت (۲)

اجرای جستجوگر محلی با استفاده از عبارت (۱۰)

محاسبه تابع برازندگی برای اعضای جمعیت توسط رابطه (۵)

$$FES = FES + \text{ceil}(m/2) \text{ با } FES$$

به روز رسانی  $FES$  با استفاده از بهترین عضو جمعیت

**پایان**

**خروجی:** تعداد ویژگی مورد نیاز

## ۵-۱- مجموعه داده‌ها

در این مقاله از شش مجموعه داده چند برچسبی شناخته شده برای نشان دادن عملکرد روش پیشنهادی نسبت به سایر روش‌ها استفاده شده است. این مجموعه داده‌ها شامل chemistry، image، corel5k، cs، scene و cooking هستند [۳۱]. مجموعه داده corel5k شامل ۵۰۰۰ تصویر corel است که در کل ۳۷۴ کلمه در دایره واژگان (برچسب کلاس) آن قرار دارد و هر تصویر دارای ۴-۵ کلمه کلیدی است. در این مجموعه داده هر تصویر با استفاده از ۴۹۹ حساب (ویژگی) توصیف می‌شود. مجموعه داده‌های image و scene متعلق به حوزه تصویر هستند. در این مجموعه داده‌ها هر تصویر می‌تواند شامل چندین اشیا مانند کوه، خورشید و ... باشد. مجموعه داده‌های cs، chess و chemistry متعلق به پست‌های انجمن Stack Exchange هستند.

برای معیارهای دقت و میانگین صحت، مقدار بزرگتر نشان دهنده عملکرد بهتر است، در صورتیکه برای یک معیار دیگر مقدار کوچکتر نشان دهنده عملکرد بهتر است.

### ۳-۵- تنظیم پارامترها

همانند سایر الگوریتم‌های هوش جمعی، الگوریتم پیشنهادی نیز دارای پارامترهایی است که باید تنظیم شوند. در آزمایشات، ۶۷ درصد از نمونه‌های مجموعه داده به صورت تصادفی به عنوان داده‌ی آموزشی و ۳۳ درصد باقی‌مانده به عنوان داده‌های تست در نظر گرفته شده‌اند. تعداد ذرات جمعیت ۵۰ ( $m=50$ ) و تعداد تکرارهای الگوریتم ۱۰۰ در نظر گرفته شده‌اند. به عنوان ضریب  $\text{norm}\ell_{1,1}$ - برای ماتریس انتقال  $W$  (ذره)،  $\lambda$  تأثیری بسزایی بر روی تَنکی  $W$  می‌گذارد که بر اساس آزمایشات، مقدار  $\lambda = 10^{-2}$  انتخاب شده است. همچنین، پارامتر  $k$  (تعداد نزدیکترین همسایه‌ها) در ML-KNN برابر ۱۰ تنظیم شده است.

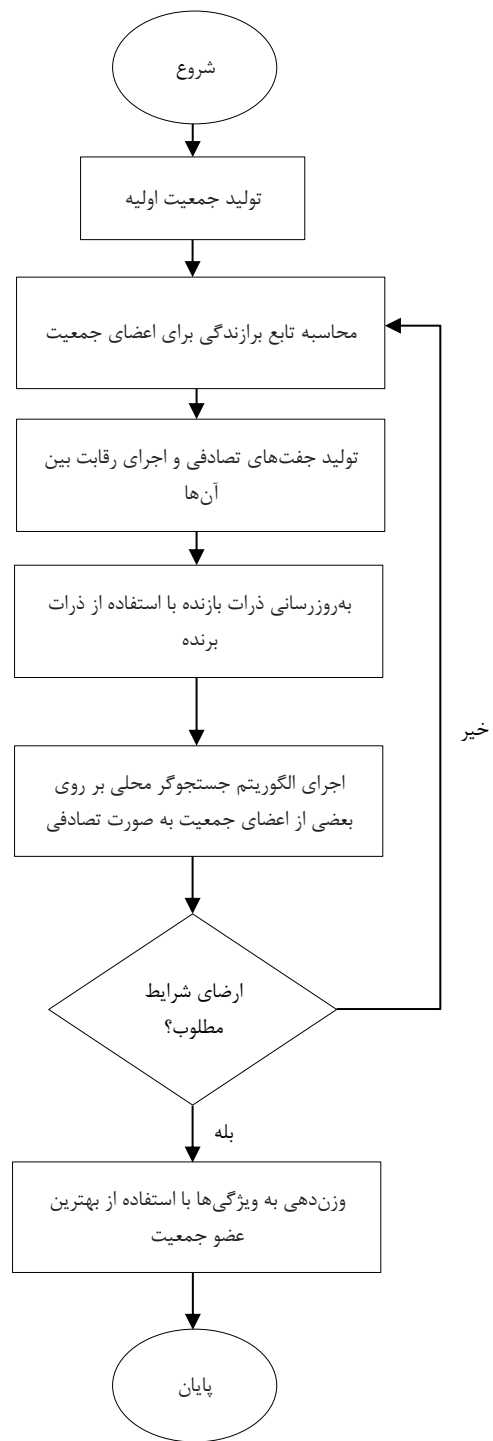
### ۴-۵- نتایج آزمایش‌ها

برای تست و پیاده‌سازی روش پیشنهادی و مقایسه با روش‌های دیگر از زبان Matlab\_R2018a بر روی سیستم عامل ویندوز ۸ و پردازنده اینتل i5 نسل ۴ با فرکانس هر هسته ۲.۵ گیگا هرتز استفاده شده است. هر الگوریتم ۲۰ بار انجام می‌شود و در نهایت از نتایج به دست آمده میانگین گرفته می‌شود. برای ارزیابی کارایی هر یک از روش‌های انتخاب ویژگی تست شده در این مقاله، تعداد ۷ ویژگی برتر برای هر الگوریتم انتخاب شده است که داریم  $v \in \{10,20,30,40,50,60,70,80,90,100\}$  و سپس دسته‌بندی را بر اساس زیرمجموعه ویژگی‌های انتخاب شده انجام می‌دهیم. نتایج سایر روش‌های انتخاب ویژگی، با استفاده از اجرای کد-های ارائه شده توسط نویسندگان هر روش به دست آمده است.

در جدول ۲، تعداد ۱۰ ویژگی برتر انتخاب شده توسط الگوریتم پیشنهادی برای مجموعه داده‌ها ارائه شده است. در جدول‌های ۳ تا ۵ میانگین معیارهای ارزیابی چندبرچسبی بر روی زیرمجموعه ویژگی‌های نهایی انتخاب شده، نمایش داده شده است، و بهترین نتیجه برای هر الگوریتم و هر مجموعه داده برجسته شده است.

از جدول‌های ۳ تا ۵ می‌توان نتیجه گرفت که روش پیشنهادی می‌تواند عملکرد طبقه‌بندی بهتری نسبت به سایر روش‌های انتخاب ویژگی دیگر به دست آورد. به طور خاص، جداول ۳ و ۴ نشان می‌دهند که روش پیشنهادی به طور قابل توجهی بالاترین دقت و کمترین خطای یک خطا را در کلیه مجموعه داده‌ها نسبت به همه روش‌ها به دست می‌آورد. همچنین جدول ۵ نشان می‌دهد که روش پیشنهادی از نظر معیار میانگین صحت به جز یک مورد نتایج بهتری نسبت به سایر روش‌ها به دست آورده است.

با توجه به نتایج به دست آمده از معیار دقت، می‌توان نتیجه گرفت که روش پیشنهادی در همه مجموعه داده‌ها نسبت به سایر روش‌ها عملکرد



شکل ۲: چهار چوب کلی روش پیشنهادی

معیار میانگین صحت: میانگین درصد برچسب کلاس‌های رتبه‌بندی شده مرتبط که بزرگتر از برچسب واقعی  $y \in Y_i$  هستند را محاسبه می‌کند:

$$Avg - Pre(f) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|Y_i|}$$

$$\sum_{y \in Y_i} \frac{\{y' \mid rank_f(x, y') \leq rank_f(x_i, y), y' \in Y_i\}}{rank_f(x_i, y)} \quad (15)$$



برای مقایسه آماری عملکرد الگوریتم‌های انتخاب ویژگی از تست آماری ویلکاکسن استفاده شده است. تست ویلکاکسن یک تست آماری بدون پارامتر است که برای مقایسه دو الگوریتم‌ها استفاده می‌شود. این تست تفاوت بین هر دو الگوریتم را محاسبه کرده و این تفاوت‌ها را تحلیل می‌کند [۳۳]. در جدول‌های ۳ تا ۵، سطر آخر هر جدول نتیجه تست ویلکاکسن بین روش پیشنهادی و سایر روش‌ها را نمایش می‌دهد، به شکلی که علامت (+) برتری آماری روش پیشنهادی نسبت به سایر روش‌ها، و علامت (=) برابری آماری دو روش انتخاب ویژگی را نمایش می‌دهند.

با توجه به نکات گفته شده و با در نظر گرفتن سطر آخر جداول ۳ تا ۵ می‌توان نتیجه گرفت که روش پیشنهادی در دو معیار دقت و یک خطا در تمام موارد نسبت به سایر روش‌ها برتری آماری دارد، و همچنین با توجه به سطر آخر جدول ۵ می‌توان مشاهده کرد که در معیار میانگین صحت، روش پیشنهادی به جز روش ela\_chi تمام روش‌های دیگر را از نظر آماری شکست می‌دهد.

با توجه به نتایج به دست آمده مشخص است که در اکثر آزمایشات، بهترین نتایج، از زیر مجموعه ویژگی‌های انتخاب شده توسط روش پیشنهادی به دست می‌آیند و عملکرد دسته‌بند در روش پیشنهادی نسبت به سایر روش‌ها بهتر است، و همچنین نتایج تست ویلکاکسن گواهی بر این فرضیه است که روش پیشنهادی از نظر آماری نیز نسبت به سایر روش‌ها برتری دارد.

بهتری را از خود نشان داده است و به طور میانگین نسبت به قوی‌ترین رقیب خود یعنی روش ppt\_chi بر روی مجموعه داده‌های Cs، Scene، Chemistry، Cooking، Corel5k، و Image به ترتیب ۰.۰۲۷۳، ۰.۰۰۶۴، ۰.۰۰۶۵، ۰.۰۱۸، ۰.۱۷۴ و ۰.۲۸۶ درصد بهبود داشته است. جزئیات نتایج برای معیارهای یک خطا و میانگین صحت در جدول‌های ۴ و ۵ ارائه شده است.

جدول ۲: ویژگی برتر انتخاب شده توسط روش پیشنهادی

مجموعه داده	۱۰ ویژگی برتر انتخاب شده
Cs	۴۴۵.۲۰۹.۲۲۹.۴۶۷.۲۴۲.۴۶۲.۲۹۸.۳۹۳.۲۳۷.۳۷۶
Corel5k	۲۳۰.۳۸۰.۳۶۲.۶۲.۴۹۶.۱۱۲.۲۸۵.۳۴۷.۴۵۹.۱۴۱
Chemistry	۲۲۹.۱۰۵.۵۲.۱۶۱.۱۸۵.۴۲۳.۱۶۷.۴۶۰.۲۳۱.۳۶۰
Cooking	۸۳.۵۳۸.۷۶.۲۶.۳۴۹.۵۰۷.۴۷۸.۵۳۸.۷۰.۱۹۰
Image	۱.۱۲۱.۷.۴.۸۷.۱۴۹.۱۱۴.۲۰۱.۱۰۹.۳۹
Scene	۳۰.۱۲۳.۳۴.۴۴.۱۴۳.۵۹.۲۴۵.۲۸۴.۱۴۰.۱۲۳

جدول ۳: میانگین دقت ML-KNN بر روی ۱۰ زیر مجموعه نهایی ویژگی‌ها

	MLCSO	LRFS	ela_chi	ppt_chi	ppt_mi	mcls	ppt_reliefF
Cs	0.1055	0.0305	0.0603	0.0782	0.0679	0.0209	0.0242
Corel5k	0.0113	0.0029	0.0051	0.0049	0.0048	0.0035	0.0025
Chemistry	0.0406	0.0160	0.0287	0.0341	0.0339	0.0201	0.0142
Cooking	0.1420	0.0249	0.1222	0.1241	0.1132	0.0389	0.0529
Image	0.3917	0.3781	0.2079	0.2177	0.3372	0.2363	0.3280
Scene	0.5469	0.3804	0.2523	0.2609	0.3526	0.3644	0.3677
Wilcoxon		+	+	+	+	+	+

جدول ۴: میانگین معیار یک خطا ML-KNN بر روی ۱۰ زیر مجموعه نهایی ویژگی‌ها

	MLCSO	LRFS	ela_chi	ppt_chi	ppt_mi	mcls	ppt_reliefF
Cs	<b>0.5941</b>	0.7043	0.6491	0.6264	0.6414	0.7137	0.7331
Corel5k	<b>0.7497</b>	0.7766	0.7594	0.7721	0.7679	0.7720	0.7783
Chemistry	<b>0.6976</b>	0.7635	0.7140	0.7035	0.7122	0.7526	0.7581
Cooking	<b>0.5865</b>	0.8249	0.6165	0.6158	0.6372	0.7692	0.7649
Image	<b>0.3805</b>	0.3914	0.5096	0.5038	0.4172	0.4861	0.4214
Scene	<b>0.3091</b>	0.4122	0.5041	0.4947	0.4317	0.4325	0.43715
Wilcoxon		+	+	+	+	+	+

جدول ۵: میانگین معیار میانگین صحت، ML-KNN بر روی ۱۰ زیر مجموعه نهایی ویژگی‌ها

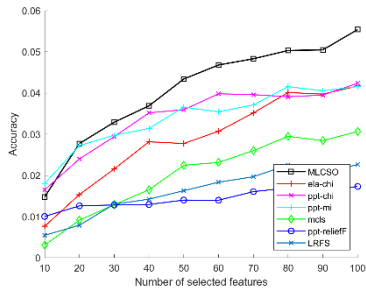
	MLCSO	LRFS	ela_chi	ppt_chi	ppt_mi	mcls	ppt_reliefF
Cs	<b>0.3568</b>	0.2697	0.3191	0.3340	0.3201	0.2476	0.2443
Corel5k	0.2283	0.2150	<b>0.2284</b>	0.2191	0.2192	0.2152	0.2121
Chemistry	<b>0.3223</b>	0.2709	0.3088	0.3120	0.3079	0.2750	0.2698
Cooking	<b>0.3237</b>	0.1661	0.3072	0.2993	0.2842	0.2049	0.2042
Image	<b>0.7529</b>	0.7471	0.6666	0.6714	0.7295	0.6828	0.7256
Scene	<b>0.8128</b>	0.7520	0.6739	0.6819	0.7360	0.7259	0.7215
Wilcoxon		+	=	+	+	+	+

میانگین ۲۰ اجرا مستقل برای هر روش هستند.

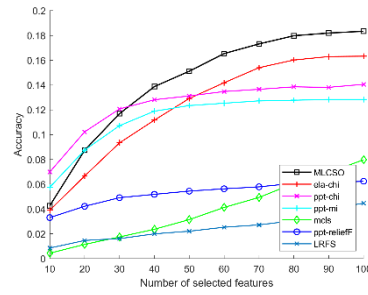
در ادامه برای سنجش میزان تاثیر جستجوگر محلی بر روی عملکرد روش پیشنهادی، نتایج به دست آمده از روش پیشنهادی با استفاده از جستجوگر محلی و بدون استفاده از جستجوگر محلی را برای دو معیار دقت و یک خطا و بر روی مجموعه داده‌های Cs، Corel5k، Chemistry و Cooking با یکدیگر مقایسه می‌نماییم. شکل ۶ نتایج به دست آمده از روش پیشنهادی با جستجوگر محلی و بدون جستجوگر محلی را بر روی معیار دقت نشان می‌دهد و همانطور که مشخص است روش پیشنهادی با جستجوگر محلی عملکرد بهتری را در مقایسه با روش پیشنهادی بدون جستجوگر محلی از خود نشان می‌دهد. نتایج به دست آمده از معیار یک خطا برای روش پیشنهادی بدون استفاده از جستجوگر محلی و با استفاده از جستجوگر محلی در شکل ۷ ارائه شده است. همانطور که از شکل مشخص است، روش پیشنهادی با جستجوگر محلی خطای کمتری را نسبت به روش پیشنهادی بدون جستجوگر محلی از خود ثبت کرده است.

با توجه به شکل‌های ۳ تا ۵ می‌توان نتیجه گرفت که در اکثر زیرمجموعه ویژگی‌های انتخاب شده از مجموعه داده‌ها، روش پیشنهادی عملکرد بهتری را از خود نشان داده است. برای مثال روش پیشنهادی در مجموعه داده‌های cs، corel5k و scene قابل توجهی نسبت به سایر روش‌ها دارد. همانطور که در شکل ۳ مشاهده می‌کنید روش پیشنهادی بر روی مجموعه ویژگی corel5k برتری چشم‌گیری نسبت به سایر روش‌ها دارد و تنها با انتخاب ۳۰ ویژگی بیشترین دقت را از خود نشان داده است و همچنین بر روی مجموعه داده‌های cs و scene برتری کاملی نسبت به سایر روش‌ها دارد.

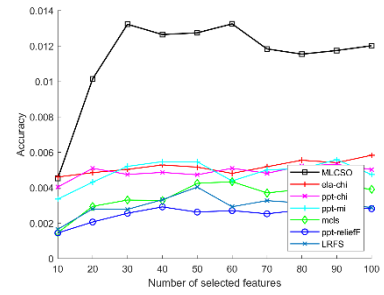
با توجه به شکل ۴ که مربوط به معیار یک خطا است با افزایش تعداد ویژگی‌های انتخاب شده خطای به دست آمده نیز کمتر می‌شود، برای مثال روش پیشنهادی بر روی مجموعه داده‌های cs و scene نسبت به سایر روش‌ها کاملاً برتر است. به طور کلی، در کلیه مجموعه داده‌ها، روش پیشنهادی با انتخاب تعداد کمتری از ویژگی‌ها، عملکرد بهتری نسبت به سایر روش‌های انتخاب ویژگی از خود نشان داده است. در تمام شکل‌ها، محور افقی تعداد ویژگی‌های انتخاب شده و محور عمودی مقدار معیارهای ارزیابی چند برچسبی را نشان می‌دهند، و نتایج به دست آمده



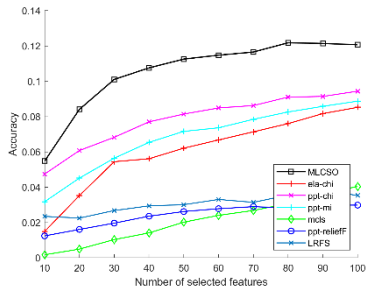
(a) chemistry



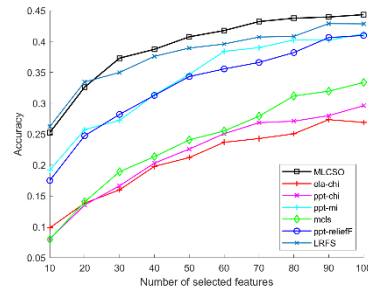
(b) cooking



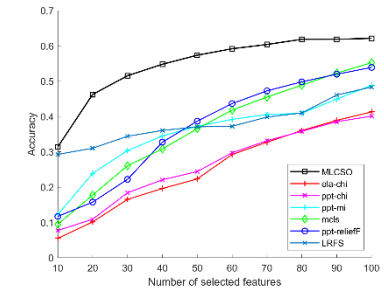
(c) Core15k



(d) cs

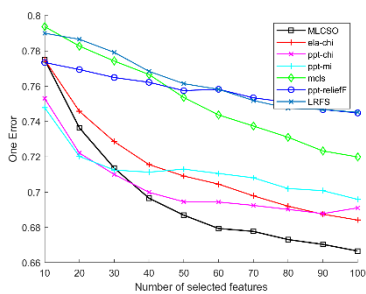


(e) image

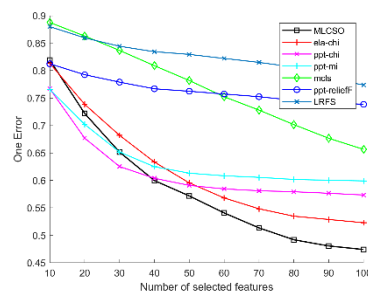


(f) scene

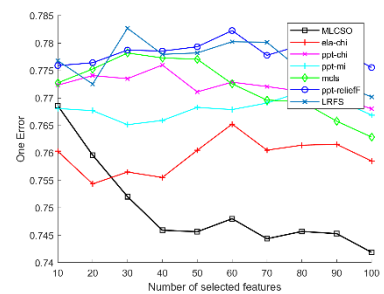
شکل ۳: مقایسه عملکرد دسته‌بند ML-KNN از لحاظ معیار دقت برای روش‌های مختلف بر روی شش مجموعه داده با توجه به ویژگی‌های انتخاب شده



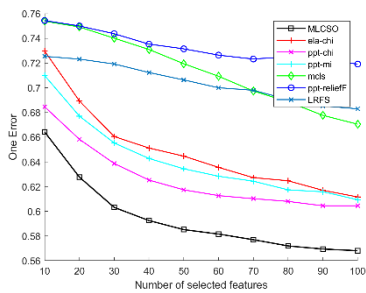
(a) chemistry



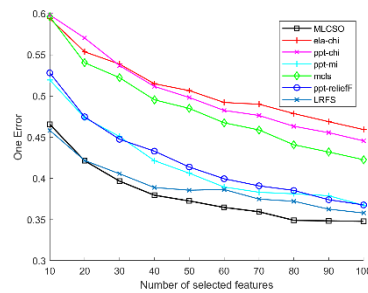
(b) cooking



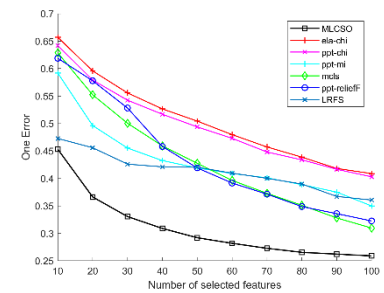
(c) Core15k



(d) cs

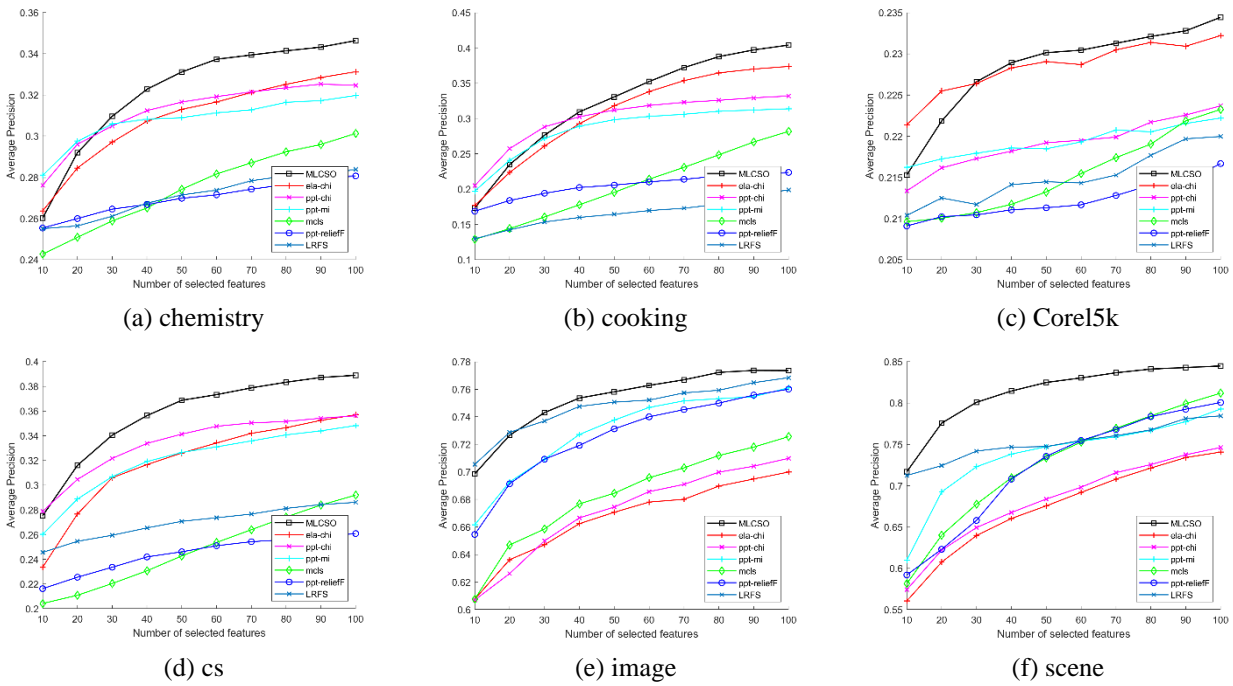


(e) image

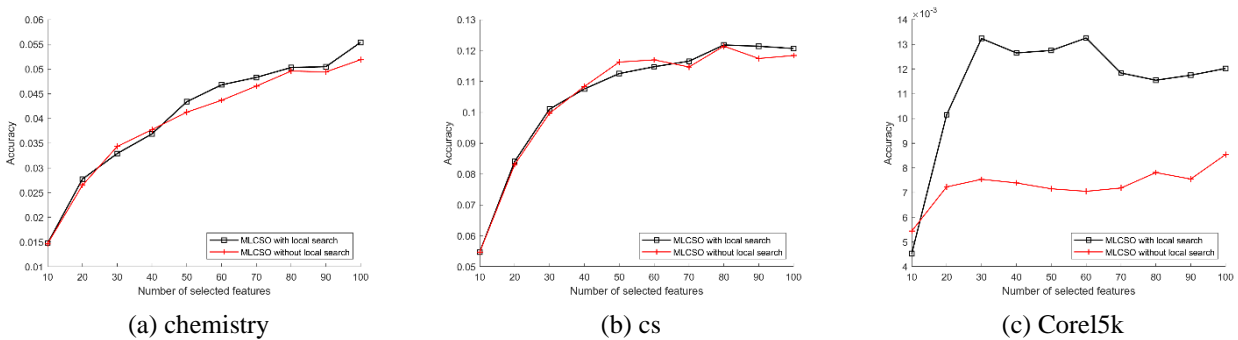


(f) scene

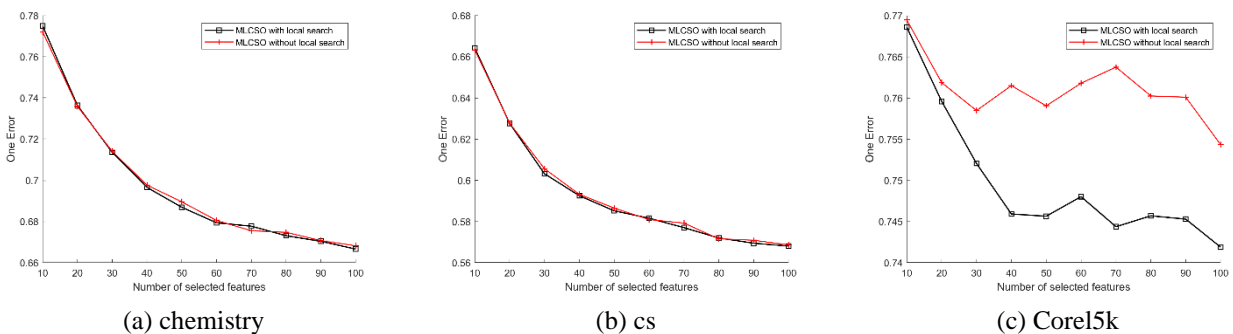
شکل ۴: مقایسه عملکرد دسته‌بند ML-KNN از لحاظ معیار یک خطا برای روش‌های مختلف بر روی شش مجموعه داده با توجه به ویژگی‌های انتخاب شده



شکل ۵: مقایسه عملکرد دسته‌بند ML-KNN از لحاظ معیار میانگین صحت برای روش‌های مختلف بر روی شش مجموعه داده با توجه به ویژگی‌های انتخاب شده



شکل ۶: مقایسه عملکرد دسته‌بند ML-KNN از لحاظ معیار دقت برای روش پیشنهادی همراه با جستجوگر محلی و بدون جستجوگر محلی بر روی سه مجموعه داده



شکل ۷: مقایسه عملکرد دسته‌بند ML-KNN از لحاظ معیار یک خطا برای روش پیشنهادی همراه با جستجوگر محلی و بدون جستجوگر محلی بر روی سه مجموعه داده

- Neurocomputing*, vol. 312, pp. 310–323, Oct. 2018.
- [2] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [3] N. Zhou, Y. Xu, H. Cheng, J. Fang, and W. Pedrycz, “Global and local structure preserving sparse subspace learning: An iterative approach to unsupervised feature selection,” *Pattern Recognit.*, vol. 53, pp. 87–101, May 2016.
- [4] M. B. Dowlatshahi and H. Nezamabadi-Pour, “GGSA: A Grouping Gravitational Search Algorithm for data clustering,” *Eng. Appl. Artif. Intell.*, vol. 36, pp. 114–121, Nov. 2014.
- [5] S. Kashef, H. Nezamabadi-pour, and B. Nikpour, “Multilabel feature selection: A comprehensive review and guiding experiments,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 2, p. e1240, Mar. 2018.
- [6] L. Du and Y. D. Shen, “Unsupervised feature selection with adaptive structure learning,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, vol. 2015-August, pp. 209–218.
- [7] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, “A Survey on semi-supervised feature selection methods,” *Pattern Recognit.*, vol. 64, pp. 141–158, Apr. 2017.
- [8] J. Luo, L. Jiao, F. Liu, S. Yang, and W. Ma, “A Pareto-Based Sparse Subspace Learning Framework,” *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3859–3872, Nov. 2019.
- [9] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, “A comparison of multi-label feature selection methods using the problem transformation approach,” *Electron. Notes Theor. Comput. Sci.*, vol. 292, pp. 135–151, Mar. 2013.
- [10] A. Hashemi and M. B. Dowlatshahi, “MLCR: A Fast Multi-label Feature Selection Method Based on K-means and L2-norm,” in *2020 25th International Computer Conference, Computer Society of Iran, CSICC 2020*, 2020.
- [11] M. B. Dowlatshahi, V. Derhami, and H. Nezamabadi-pour, “A Novel Three-Stage Filter-Wrapper Framework for miRNA Subset Selection in Cancer Classification,” *Informatics*, vol. 5, no. 1, p. 13, Mar. 2018.
- [12] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, “MLACO: A multi-label feature selection algorithm based on ant colony optimization,” *Knowledge-Based Syst.*, p. 105285, Dec. 2019.
- [13] S. Tabakhi, P. Moradi, and F. Akhlaghian, “An unsupervised feature selection algorithm based on

در ادامه به بررسی و تشریح اینکه چرا عملکرد الگوریتم MLCSSO نسبت به سایر روش‌های انتخاب ویژگی چندبرچسبی بهتر است خواهیم پرداخت:

- رفتار جمعی و رقابت دو به دو ذرات در CSO باعث می‌شود بخش بزرگی از فضای ویژگی مورد بررسی قرار گیرد. همچنین نیمی از ذرات اولیه بصورت تصادفی تولید و نیم دیگر با استفاده از معیار شباهت کسینوسی تولید می‌شوند، که قابلیت اکتشاف و بهره‌گیری الگوریتم را تنظیم می‌کند.
- تابع هزینه روش پیشنهادی از نمایش تَنک استفاده می‌کند که هدف آن پیدا کردن یک ماتریس تَنک است که داده‌های ورودی را با کمترین خطای ممکن به برچسب کلاس‌ها نگاشت می‌کند. ماتریس تَنک بهینه، وزن کمتری به ویژگی‌های نامربوط و زائد می‌دهد، بنابراین ویژگی‌هایی که وزن بیشتری به دست می‌آورند به عنوان زیر مجموعه ویژگی نهایی انتخاب می‌شوند.
- در الگوریتم پیشنهادی، یک روش دو مرحله‌ای برای یافتن راه‌حل بهینه تابع هزینه ارائه شده است. مرحله اول، جستجوی سراسری توسط رقابت ذرات با یکدیگر است، و مرحله دوم جستجوی محلی (گرادیان کاهشی) پیرامون بهترین راه‌حل موجود در مرحله اول است. به عبارت دیگر، یک بخش مهم از الگوریتم پیشنهادی، مرحله جستجوی محلی است که در آن گرادیان کاهشی برای بهبود بیشتر هر ذره با استفاده از همسایه‌هایش استفاده می‌شود.

## ۶- نتیجه‌گیری

در این مقاله، یک الگوریتم انتخاب ویژگی نظارتی چند برچسبی مبتنی بر الگوریتم هوش جمعی بهینه‌ساز جمعیت رقابتی ارائه شده است. در این الگوریتم با ساخت هدفمند جمعیت اولیه، استفاده از تابع هدف مناسب برای کاهش هزینه و معرفی یک جستجوگر محلی، بهترین ویژگی‌ها با کمترین افزونگی و بالاترین ارتباط با برچسب‌های کلاس انتخاب می‌شوند. عملکرد روش پیشنهادی با روش‌های انتخاب ویژگی نظارتی چند برچسبی شناخته شده از جمله PPT\_MI, PPT\_Relief, PPT\_CH, ELA\_CHI, MCLS بر روی شش مجموعه داده مقایسه شد. برای تست دقت الگوریتم‌ها از ML-KNN به عنوان دسته‌بند استفاده و نتایج حاصل از سه معیار ارزیابی مرسوم گزارش شد. آزمایشات نشان می‌دهند که روش ارائه شده دارای عملکرد دسته‌بندی بهتری در مقایسه با سایر روش‌های انتخاب ویژگی است.

## مراجع

- [1] S. Feng and M. F. Duarte, “Graph autoencoder-based unsupervised feature selection with broad and local data structure preservation,”

- (including subseries *Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), 2014, vol. 8818, pp. 454–463.
- [26] R. Huang, W. Jiang, and G. Sun, “Manifold-based constraint Laplacian score for multi-label feature selection,” *Pattern Recognit. Lett.*, vol. 112, pp. 346–352, Sep. 2018.
- [27] W. Chen, J. Yan, B. Zhang, Z. Chen, and Q. Yang, “Document transformation for multi-label feature selection in text categorization,” in *Proceedings - IEEE International Conference on Data Mining, ICDM, 2007*, pp. 451–456.
- [28] P. Zhang, G. Liu, and W. Gao, “Distinguishing two types of labels for multi-label feature selection,” *Pattern Recognit.*, vol. 95, pp. 72–82, Nov. 2019.
- [29] A. Hashemi, M. B. Dowlatshahi, and H. Nezamabadi-pour, “MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality,” *Expert Syst. Appl.*, vol. 142, p. 113024, Mar. 2020.
- [30] J. Kennedy and R. Eberhart, “Particle Swarm Optimization,” pp. 1942–1948, 1995.
- [31] F. Charte and D. Charte, “Working with multilabel datasets in R: The mlr package,” *R J.*, vol. 7, no. 2, pp. 149–162, 2015.
- [32] M. L. Zhang and Z. H. Zhou, “ML-KNN: A lazy learning approach to multi-label learning,” *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.
- [33] D. Sheskin, “Handbook of parametric and nonparametric statistical procedures, 2003.
- ant colony optimization,” *Eng. Appl. Artif. Intell.*, vol. 32, pp. 112–123, 2014.
- [14] S. Wang, J. Tang, and H. Liu, “Embedded unsupervised feature selection,” *Proc. Natl. Conf. Artif. Intell.*, vol. 1, pp. 470–476, 2015.
- [15] M. B. Dowlatshahi, V. Derhami, and H. Nezamabadi-pour, “Ensemble of Filter-Based Rankers to Guide an Epsilon-Greedy Swarm Optimizer for High-Dimensional Feature Subset Selection,” *Information*, vol. 8, no. 4, p. 152, Nov. 2017.
- [16] M. B. Dowlatshahi, V. Derhami, and H. Nezamabadi-Pour, “Fuzzy particle swarm optimization with nearest-better neighborhood for multimodal optimization,” *Iran. J. Fuzzy Syst.*, vol. 17, no. 4, pp. 7–24, Jul. 2020.
- [17] M. B. Dowlatshahi and V. Derhami, “Winner Determination in Combinatorial Auctions using Hybrid Ant Colony Optimization and Multi-Neighborhood Local Search,” *J. AI Data Min.*, vol. 5, no. 2, pp. 169–181, Jul. 2017.
- [18] M. B. Dowlatshahi and M. Rezaeian, “Training spiking neurons with gravitational search algorithm for data classification,” in *1st Conference on Swarm Intelligence and Evolutionary Computation, CSIEC 2016 - Proceedings*, 2016, pp. 53–58.
- [19] Dowlatshahi, M., Derhami, V., Nezamabadi-pour, H. “Gravitational Locally Informed Particle Swarm Algorithm for solving Multimodal Optimization Problems,” *Tabriz Journal Of Electrical Engineering*, 48(3), 2018, pp.1131-1140.
- [20] E. G. Talbi, *Metaheuristics: From Design to Implementation*. 2009.
- [21] H. Bayati, M. B. Dowlatshahi, and M. Paniri, “MLPSO: A Filter Multi-label Feature Selection Based on Particle Swarm Optimization,” in *2020 25th International Computer Conference, Computer Society of Iran, CSICC 2020*, 2020.
- [22] R. Cheng and Y. Jin, “A competitive swarm optimizer for large scale optimization,” *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.
- [23] J. Lee and D. W. Kim, “Feature selection for multi-label classification using multivariate mutual information,” *Pattern Recognit. Lett.*, vol. 34, no. 3, pp. 349–357, Feb. 2013.
- [24] O. Reyes, C. Morell, and S. Ventura, “Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context,” *Neurocomputing*, vol. 161, pp. 168–182, Aug. 2015.
- [25] Y. Yu, Y. Yu, and Y. L. Wang, “Feature selection for multi-label learning using mutual information and GA,” in *Lecture Notes in Computer Science*

#### پاورقی‌ها:

<sup>1</sup> High-dimensional

<sup>2</sup> Irrelevant

<sup>3</sup> Redundant

<sup>4</sup> Curse of dimensionality

<sup>5</sup> Feature selection

<sup>6</sup> Overfitting

<sup>7</sup> Unsupervised

<sup>8</sup> Semi-supervised

<sup>9</sup> Supervised

<sup>10</sup> Correlation

<sup>11</sup> Single-label

<sup>12</sup> Multi-label

<sup>13</sup> Wrapper

<sup>14</sup> Embedded

<sup>15</sup> Ant Colony Optimization (ACO)

<sup>16</sup> Gravitational Search Algorithm (GSA)

<sup>17</sup> Competitive Swarm Optimizer (CSO)

<sup>18</sup> Particle

- <sup>19</sup> Global search
- <sup>20</sup> Pruned problem transformation
- <sup>21</sup> Manifold-based constraint Laplacian score
- <sup>22</sup> Entropy-based label assignment
- <sup>23</sup> Label redundancy feature selection
- <sup>24</sup> Particle Swarm Optimization (PSO)
- <sup>25</sup> Position
- <sup>26</sup> Velocity
- <sup>27</sup> Sparse representation
- <sup>28</sup> Cosine
- <sup>29</sup> Reconstruction error
- <sup>30</sup> Maximum a posteriori
- <sup>31</sup> One error
- <sup>32</sup> Average Precision