انتخاب برخط ویژگی‌های جریانی با استفاده از سری هندسی ماتریس مجاورت ویژگی‌ها

صادق اسکندری

گروه علوم رایانه، دانشکده علوم ریاضی، دانشگاه گیلان، رشت، ایران

چکیده

انتخاب، ویژگی یکی از گام‌های پیش‌پردازش مهم در یادگیری ماشینی و داده‌کاوی است. تمامی الگوریتم‌های انتخاب ویژگی سنتی فرض می‌کنند که کل فضای ویژگی از ابتدای چرخهٔ انتخاب در دسترس است؛ با این وجود در بسیاری از کاربردهای دنیای واقعی با سناریوی ویژگی‌های جریانی مواجه هستیم. در این سناریو، تعداد ویژگی‌ها به‌مرور زمان افزایش می‌یابد. در این مقاله، مسأله انتخاب برخط ویژگی‌های جریانی از منظر سری‌های هندسی گراف ارتباط ویژگی‌ها مورد بررسی قرار گرفته و یک الگوریتم جدید به نام OSFS-GS پیشنهاد شده است. این الگوریتم با استفاده از مفهوم سری هندسی گراف مجاورت، ویژگی‌های افزونه را به شکل برخط حذف می‌کند؛ علاوه‌براین، الگوریتم پیشنهادی از یک سازوکار نگهداری ویژگی‌های افزونه بهره می‌بَرَد که امکان بررسی مجدد ویژگی‌های بسیار خوبی را که درقبل حذف شده‌اند، فراهم می‌آورد. الگوریتم پیشنهادی بر روی هشت مجموعه‌داده با ابعاد بزرگ اعمال شده و نتایج نشان‌دهنده دقت بالای این الگوریتم در نمونه‌های زمانی مختلف است.

واژگان کلیدی: ویژگی‌های جریانی، انتخاب ویژگی، سری هندسی.

Online Streaming Feature Selection Using Geometric Series of the Adjacency Matrix of Features

Sadegh Eskandari

Department of Computer Science, Faculty of Mathematical Sciences,
University of Guilan, Rasht, Iran

Abstract

Feature Selection (FS) is an important pre-processing step in machine learning and data mining. All the traditional feature selection methods assume that the entire feature space is available from the beginning. However, online streaming features (OSF) are an integral part of many real-world applications. In OSF, the number of training examples is fixed while the number of features grows with time as new features stream in. For instance, in the problem of semantic segmentation of images using texture-based features, the number of features can be infinitely growing.

In these dynamically growing scenarios, a rudimentary approach is waiting a long time for all features to become available and then carry out the feature selection methods. However, due to the importance of optimal decisions at every time step, a more rational approach is to design an online streaming feature selection (OSFS) method which selects a best feature subset from so-far-seen information and updates the subset on the fly when new features stream in. Any OSFS method must satisfy three critical conditions; first, it should not require any domain knowledge about feature space, because the full feature space is unknown or inaccessible. Second, it should allow efficient incremental updates in selected features. Third, it should be

* Corresponding author

*نویسندهٔ عهده‌دار مکاتبات

• تاریخ ارسال مقاله: ۱۳۹۷/۱۰/۰۹ • تاریخ پذیرش: ۱۳۹۸/۰۶/۱۱ • تاریخ انتشار: ۱۳۹۹/۱۲/۰۴ • نوع مطالعه: پژوهشی

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]

[DOI: 10.29252/jsdp.17.4.3]

as accurate as possible at each time instance to allow having reliable classification and learning tasks at that time instance.

In this paper, OSFS is considered from the geometric series of features adjacency matrix and, a new OSFS algorithm called OSFS-GS is proposed. This algorithm ranks features based on path integrals and the centrality concept on an online feature adjacency graph. The most appealing characteristics of the proposed algorithm are; 1) all possible subsets of features are considered in evaluating the rank of a given feature, 2) it is extremely efficient, as it converts the feature ranking problem to simply calculating the geometric series of an adjacency matrix and 3) beside selected features subset, it uses a redundant features subset that provides the reconsideration of good features at different time instances.

This algorithm is compared with three state-of-the-art OSFS algorithms, namely information-investing, fast-OSFS and OSFSMI. The information-investing algorithm is an embedded online feature selection method that considers the feature selection as a part of learning process. This algorithm selects a new incoming feature if it reduces the model entropy more than the cost of the feature coding. The fast-OSFS algorithm is a filter method that gradually generates a Markov-blanket of feature space using causality-based measures. For any new incoming feature, this algorithm executes two processes: an online relevance analysis and then an online redundancy analysis. OSFSMI is a similar algorithm to fast-OSFS, in which uses information theory for feature analysis.

The algorithms are extensively evaluated on eight high-dimensional datasets in terms of compactness, classification accuracy and run-time. In order to provide OSF scenario, features are considered one by one. Moreover, in order to strengthen the comparison, the results are averaged over 30 random streaming orders. Experimental results demonstrate that OSFS-GS algorithm achieves better accuracies than the three existing OSFS algorithms.

Keywords: **Streaming Features, Feature Selection, Geometric Series.**

۱- مقدمه

اگرچه یک مجموعه‌ویژگی بزرگ، اطلاعات بیشتری را درباره یک مفهوم در اختیار ما قرار می‌دهد، در رده‌بندی و یادگیری ماشین این مجموعه تا جایی که امکان دارد باید کوچک در نظر گرفته شود. دلیل اصلی این کار، مقابله با پدیده نفرین ابعاد[1] و افزایش قابلیت تعمیم مدل یادگیری است [1,2,3]. در مسائل کلاسیک انتخاب ویژگی، یک مجموعه‌داده با n بردار ویژگی mبُعدی و یک یا چند ویژگی خروجی (رده) داده شده و هدف انتخاب ویژگی، انتخاب یک زیرمجموعه کمینه از مهم‌ترین و تمییزدهنده‌ترین ویژگی‌ها است. در طول سه دهه اخیر، الگوریتم‌های مختلفی جهت انتخاب ویژگی معرفی شده‌اند [2,4,5,6]. تمامی الگوریتم‌های انتخاب ویژگی سنتی فرض می‌کنند که کل فضای ویژگی از همان ابتدای چرخهٔ انتخاب در دسترس است؛ اما، ویژگی‌های جریانی بخش جدایی‌ناپذیر بسیاری از کاربردهای دنیای واقعی هستند. در این سناریو، تعداد بردارهای ویژگی (n) ثابت بوده و اندازه مجموعه ویژگی (m) با گذشت زمان رشد می‌کند. می‌توان دو دلیل اصلی برای جریانی‌بودن ویژگی‌ها بیان کرد:

۱. فضای ویژگی نامعلوم و یا حتی بی‌نهایت است. به‌عنوان مثال، در مسائل یادگیری بیوانفورماتیک استخراج تمامی ویژگی‌ها برای تمامی نمونه‌ها گاهی به‌دلیل محدودیت‌های آزمایشگاهی بسیار زمان‌بر است [7].

به‌عنوان مثالی دیگر، در مسأله قطعه‌بندی مبتنی بر متن تصاویر[2]، تعداد فیلترهای متن می‌تواند بی‌نهایت باشد و درنتیجه ایجاد کل مجموعه‌ویژگی غیر ممکن خواهد بود [8,9,10].

۲. فضای ویژگی مشخص است؛ ولی پردازش آن‌ها به شکل جریانی مزایای بیشتری در مقایسه با پردازش دسته‌ای در اختیار ما قرار می‌دهد. این سناریو اغلب در مجموعه‌داده‌هایی که دارای فضای ویژگی بسیار بزرگی هستند، یک انتخاب مفید خواهد بود. به‌عنوان مثال، در مسائل رده‌بندی اسناد[3] یک مجموعه داده می‌تواند شامل صدها هزار ویژگی باشد [11] و درنتیجه، استفاده از الگوریتم‌های انتخاب ویژگی سنتی نیاز به فضا و زمان بسیار زیادی خواهد داشت.

شاید ابتدائی‌ترین ایده جهت انتخاب ویژگی در چنین سناریوهایی، منتظرماندن جهت تکمیل فضای ویژگی و سپس استفاده از الگوریتم‌های سنتی انتخاب ویژگی باشد؛ ولی، همان‌طور که گفته شد، در فضاهای ویژگی بسیار بزرگ و بی‌نهایت، این ایده عملی نخواهد بود. ایده کارآمدتر و منطقی‌تر، طراحی یک الگوریتم انتخاب ویژگی برخط است که بهترین زیرمجموعه را بر اساس ویژگی‌هایی که تابه‌حال مشاهده شده‌اند، انتخاب کرده و در زمان ورود هر ویژگی جدید، زیرمجموعه انتخابی را بروز کند. هر الگوریتم انتخاب

[1] Curse of Dimensionality

[2] Texture-Based Image Segmentation

[3] Document Classification

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]

[DOI: 10.29252/jsdp.17.4.3]

ویژگی برخط برای ویژگی‌های جریانی باید سه شرط اساسی را برآورده کند [1]: نخست، چنین الگوریتمی نباید نیاز به هیچ دانش اضافی درباره فضای ویژگی داشته باشد؛ زیرا فضای ویژگی نامشخص و دسترس ناپذیر است. دوم‌این‌که، تا حد امکان سریع بوده و به‌روزرسانی‌های کارآمدی را در ویژگی‌های انتخابی ممکن سازد، زیرا به‌طورمعمول در این سناریو، زمان‌های میان ورود ویژگی‌ها بسیار محدود است. سوم‌این‌که، تا جایی که ممکن است، در هر نمونهٔ زمانی دقیق باشد.

اگرچه چندین مطالعه در حوزه انتخاب برخط ویژگی‌های جریانی صورت گرفته است [1,3,9,8,11,12,13]، هیچ یک از آن‌ها تمامی نیازمندی‌های سه گانه بالا را برآورده نمی‌کنند. در این مقاله، مسأله انتخاب برخط ویژگی‌های جریانی از منظر سری‌های هندسی گراف ارتباط ویژگی‌ها مورد بررسی قرار گرفته است. انگیزه اصلی نگارنده برای چنین بررسی، امکان بیان وابستگی‌های بسیار پیچیده میان ویژگی‌ها با استفاده از مسیرهای با طول بی‌نهایت در گراف مجاورت است. در این مقاله، یک الگوریتم جدید به نام OSFS-GS پیشنهاد شده است. این الگوریتم با استفاده از مفهوم سری هندسی گراف مجاورت، ویژگی‌های افزونه را به‌شکل برخط حذف می‌کند. علاوه براین، الگوریتم پیشنهادی از یک سازوکار نگهداری ویژگی‌های افزونه بهره می‌برد که امکان بررسی مجدد ویژگی‌های بسیار خوبی را که درقبل حذف شده‌اند، فراهم می‌آورد. کارآیی و دقت الگوریتم پیشنهادی به‌وسیله نتایج تجربی متعدد به اثبات رسیده است.

ساختار مقاله بدین شکل است که در بخش دو کارهای مرتبط و سپس در بخش سه پیش‌نیازهای روش پیشنهادی بیان می‌شود. چهارچوب روش پیشنهادی در بخش چهار و نتایج تجربی در بخش پنج و در انتها نیز جمع‌بندی بیان می‌شود.

۲- پیشینهٔ پژوهش

هدف از انتخاب ویژگی، حذف ویژگی‌های نامرتبط[4] و افزونه[5] است. ویژگی‌های نامرتبط هیچ اطلاعات مفیدی درخصوص مسأله مورد نظر در اختیار ما قرار نمی‌دهند. ویژگی‌های افزونه حاوی اطلاعات مفیدی هستند؛ ولی این اطلاعات به‌وسیلهٔ ویژگی‌های بهتر دیگری نیز بیان شده‌اند. در طول سه دهه اخیر الگوریتم‌های انتخاب ویژگی فراوانی معرفی شده‌اند [2,4,5,6]. این الگوریتم‌ها را می‌توان در سه گروه دسته‌بندی کرد: روش‌های لفافه‌ای، روش‌های فیلتر و روش‌های جاسازی‌شده [14].

ساختار الگوریتم‌های انتخاب ویژگی سنتی به‌گونه‌ای است که در هر تکرار نیازمند دسترسی به کل فضای ویژگی، جهت انتخاب بهترین زیرمجموعه هستند؛ بنابراین، فرض اساسی ضمنی در این الگوریتم‌ها، مهیابودن تمامی ویژگی‌ها، قبل از شروع رویه انتخاب ویژگی است. استفاده از این الگوریتم‌ها در سناریوی ویژگی‌های جریانی امکان‌پذیر نیست؛ زیرا بزرگ‌ترین چالش در این سناریو، عدم دسترسی به کل فضای ویژگی است؛ بنابراین مطالعاتی درخصوص انتخاب ویژگی در سناریوی ویژگی‌های جریانی صورت گرفته است.

گرافتینگ برخط[6] [9] را می‌توان نخستین تلاش در حوزه انتخاب ویژگی‌های جریانی دانست. این الگوریتم نوعی روش انتخاب ویژگی جاسازی شده است که از مفاهیم الگوریتم سنتی انتخاب ویژگی گرافتینگ [15] استفاده می‌کند. در گرافتینگ برخط، ویژگی ورودی جدید درصورتی به ویژگی‌های انتخابی اضافه می‌شود که آن ویژگی دقت مدل را بیش از یک حد آستانه λ بهبود بخشد. در [10] از الگوریتم گرافتینگ برخط جهت انتخاب ویژگی‌های مناسب برای تشخیص لبه استفاده شده است. اگرچه الگوریتم گرافتینگ برخط قادر به مدیریت ویژگی‌های جریانی است، ولی استفاده از آن در سناریوهای واقعی با دو چالش اساسی مواجه است: ۱) انتخاب λ مناسب نیازمند دانش کافی درباره فضای ویژگی است و ۲) این الگوریتم از مسأله تأثیر لانه‌ای[7] [16] رنج می‌برد. بدین معنی که اگر یک ویژگی در زمان t به مجموعه ویژگی اضافه شود، این ویژگی هیچ زمان دیگری بعد از t از مجموعه حذف نخواهد شد، حتی اگر ویژگی مورد نظر افزونه شده باشد.

در [17]، یک الگوریتم جاسازی‌شده مبتنی بر رگرسیون خطی به نام سرمایه‌گذاری اطلاعات معرفی شده است. در این الگوریتم، ویژگی جدید در صورتی به مدل اضافه می‌شود که کاهش آنتروپی مدل بیش از هزینه کدگذاری ویژگی باشد. از نظر ساختاری این الگوریتم بسیار شبیه به گرافتینگ برخط است با این تفاوت که مقدار حد آستانه در طول اجرای الگوریتم به شکلی پویا تغییر می‌کند. در [18] نیز یک الگوریتم بسیار مشابه با سرمایه‌گذاری

[4] Irrelevant
[5] Redundant
[6] Online Grafting
[7] Nesting Effect

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]
[DOI: 10.29252/jsdp.17.4.3]

اطلاعات، تحت عنوان سرمایه‌گذاری آلفا معرفی شده است. این الگوریتم به جای آنتروپی از مقدار p آماری به‌عنوان معیار انتخاب ویژگی استفاده می‌کند. در [11] نشان داده شده است که دو الگوریتم سرمایه‌گذاری اطلاعات و سرمایه‌گذاری آلفا معادل هستند. اگرچه این الگوریتم‌ها قادر به مدیریت فضای ویژگی نامعلوم هستند، دو مشکل اساسی دارند: ۱) فرض اولیه در این الگوریتم‌ها آن است که تمامی ویژگی‌ها به‌وسیله منبعی با توزیع یکسان (مانند گوسین) تولید می‌شوند. این فرض ضمنی با ماهیت سناریوی ویژگی‌های جریانی در تناقض است. ۲) این الگوریتم‌ها همانند گرافتینگ برخط از مشکل تأثیر لانه‌ای رنج می‌برند. در [19] توسعه‌ای از الگوریتم‌های سرمایه‌گذاری جهت مرتفع‌ساختن مشکل نخست معرفی شده است.

در [8]، مسأله انتخاب ویژگی‌های جریانی با استفاده از تئوری ارتباط فرموله‌سازی و یک الگوریتم انتخاب برخط به نام الگوریتم سریع معرفی شده است. این الگوریتم از دو گام اساسی تشکیل می‌شود: ۱) آنالیز ارتباط برخط جهت حذف ویژگی‌های نامرتبط و ۲) آنالیز افزونگی برخط جهت حذف ویژگی‌های افزونه. الگوریتم سریع یک الگوریتم انتخاب ویژگی فیلتر است و درنتیجه بسیار سریع‌تر از الگوریتم‌های گرافتینگ برخط و سرمایه‌گذاری عمل می‌کند؛ علاوه‌براین، برخلاف الگورتیم‌های پیشین، این الگوریتم مشکل تأثیر لانه‌ای ندارد. مشکل اصلی این الگوریتم، آزمون‌های (وابستگی و استقلال) شرطی هستند. جهت تولید نتایج قابل اعتماد به‌وسیلهٔ این آزمون‌ها، نیاز به داده‌های بسیار زیادی است. بنابراین، این الگوریتم از تعداد ویژگی‌های محدودی (حداکثر ۳) جهت انجام این آزمون‌ها استفاده می‌کند. درنتیجه، افزونگی های پیچیده‌تر نادیده گرفته می‌شوند.

در [1,3,12]، مسأله انتخاب ویژگی‌های جریانی از منظر مجموعه‌های ناهموار مورد بررسی قرار گرفته است. مزیت اصلی مجموعه‌های ناهموار عدم نیاز عملگرهای آن به دانش اضافی درباره داده است. این ویژگی، مجموعه‌های ناهموار را به یک ابزار ایده‌آل برای سناریوی ویژگی‌های جریانی تبدیل می‌کند. مشکل اصلی استفاده از مجموعه‌های ناهموار، پیچیدگی محاسباتی بالای عملگرهای آن است؛ بنابراین، در تمامی الگوریتم‌های ارائه‌شده در [1,3,12]، فرض بر این است که تعداد ویژگی‌های انتخابی بسیار کوچک است. درصورتی‌که تعداد ویژگی‌های غیرافزونه و مرتبط زیاد باشد، استفاده از مجموعه‌های ناهموار بسیار ناکارآمد خواهد بود.

در [13]، یک الگوریتم فیلتر مبتنی بر تئوری اطلاعات معرفی شده است. ساختار کلی این الگوریتم بسیار شبیه به الگوریتم سریع [8] است. مشکل اصلی این الگوریتم نیز نبود درنظرگرفتن تمامی زیرمجموعه‌های مجموعه انتخابی در گام حذف ویژگی‌های افزونه است.

۳- تحلیل ویژگی به‌وسیله سری هندسی

در این مقاله، از ایده تحلیل ویژگی‌ها در [20] و [14] استفاده شده است. بنابراین، در این بخش به بررسی و معرفی این ایده می‌پردازیم. فرض کنید یک مجموعه‌داده با m ویژگی $F=\{f_1,\dots,f_m\}$ داده شده است. می‌توان گراف غیرجهت‌دار، کامل و وزن‌دار $G=(V,E,e)$ را برای این مجموعه‌داده ایجاد کرد؛ به‌گونه‌ای که $V=\{f_i|\, f_i\in F\}$ و $E=\{\{f_i,f_j\}|f_i,f_j\in F\wedge i\neq j\}$ به‌ترتیب نشان‌دهنده رئوس و یال‌ها بوده و $e:E\to\mathbb{R}$ تابعی است که انرژی دوبه‌دوی ویژگی‌ها را محاسبه می‌کند. گراف G را می‌توان با استفاده از ماتریس مجاورت A نشان داد؛ به‌گونه‌ای که $a_{ij}=e(\{f_if_j\})$. حال فرض کنید P_{ij}^l نشان‌دهنده مجموعه تمامی مسیرهای با طول l میان رئوس i و j باشد. دقت داشته باشید که این مجموعه می‌تواند شامل مسیرهایی که حاوی دور هستند نیز باشد.

یک معیار اولیه جهت رتبه بندی ویژگی‌ها آن است که یک l مناسب را انتخاب کرده و مقدار $s_e(i)$ را برای هر ویژگی f_i محاسبه کنیم:

$$s_e(i)=\sum_{j\in V}\sum_{p\in P_{i,j}^l}\prod_{k=0}^{l-1}a_{v_k,v_{k+1}}=\sum_{j\in V}A^l(i,j) \qquad (۱)$$

این معیار دارای دو مشکل کلی است: نخست، دورها می‌توانند تأثیر معناداری در مقادیر نهایی داشته باشند. دوم، محاسبه A^l از مرتبه $O(n^4)$ است که برای تعداد زیاد ویژگی بسیار ناکارآمد است. در [20] پیشنهاد شده است تا به جای استفاده از یک طول مسیر خاص، مجموع معیارها با مقادیر متفاوت l (حتی مقادیر بی‌نهایت) مورد استفاده قرار گیرد. این کار موجب یک‌نواخت‌شدن احتمال عضویت ویژگی‌ها در دورها شده و تأثیر دورها را از بین می‌برد؛ بنابراین، معیار رتبه‌بندی جدید برای هر ویژگی f_i به‌صورت زیر قابل بیان است:

$$s(i)=\left[\left(\left(\sum_{l=0}^{\infty}A^l\right)-I\right)\vec{\mathbf{1}}\right]_i \qquad (۲)$$

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]

[DOI: 10.29252/jsdp.17.4.3]

که در آن I یک ماتریس همانی و $\overline{\mathbf{1}}$ بردار ستونی از یک‌ها می باشند.

در جبر ماتریس‌ها، $\sum_{l=0}^{\infty} X^l$ را سری هندسی ماتریس X می‌نامند. این سری به $(I-X)^{-1}$ همگرا است اگر و تنها اگر $\rho(X) < 1$ که در آن $\rho(X)$ بزرگترین مقدار ویژه X است. می‌توان نشان داد که برای هر ماتریس X، $\rho(rX) < 1$ اگر و تنها اگر $0 < r < \frac{1}{\rho(X)}$. با استفاده از این ویژگی، معیار منظم‌شده برای هر ویژگی به‌صورت زیر قابل تعریف است:

$$s'(i) = \left[\left(\left(\sum_{l=0}^{\infty} r^l A^l\right) - I\right)\overline{\mathbf{1}}\right]_i = \left[((I-rA)^{-1} - I)\overline{\mathbf{1}}\right]_i \quad (۳)$$

پیچیدگی محاسبه $(I-rA)^{-1} - I$ برابر $O(n^{2.37})$ است که در مقایسه با $O(n^4)$ برای فرمول (۱)، بسیار کارآمدتر است.

یکی از مسائل مهم در راهبرد بالا، تعیین تابع e است. در [14] نشان داده شده است که برای سناریوهای نظارت‌شده، تابع زیر بهترین دقت را نتیجه می‌دهد:

$$e(\{f_i f_j\}) = \alpha\left(max\left(I(f_i,Y), I(f_j,Y)\right)\right) + (1-\alpha)(1 - |SPR(f_i,f_j)|) \quad (۴)$$

که در آن $I(.,.)$ و $SPR(.,.)$ به‌ترتیب نشان‌دهنده اطلاعات متقابل و ضریب همبستگی رتبه‌ای اسپیرمن هستند. علاوه‌براین در [14] نشان داده شده است که $\alpha \in [0.8,1]$ بازه مناسبی برای تعیین ضرایب تأثیر هر یک از معیارهای بالا است.

۴- روش پیشنهادی

فرض کنید $D_t = (X_t, F_t, C)$ مجموعه‌داده در زمان t باشد که در آن $X_t = \{X_{1_t}, X_{2_t}, \dots, X_{N_t}\}$ مجموعه‌بردارهای ویژگی، $F_t = \{f_{1_t}, f_{2_t}, \dots, f_{M_t}\}$ مجموعه ویژگی و C نیز صفت رده باشد. در سناریوی ویژگی‌های جریانی، ویژگی‌های جدید به‌مرور زمان اضافه می‌شوند؛ درحالی‌که تعداد بردارهای ویژگی ثابت است. به عبارت دیگر، در این سناریو برای هر $t' > t$، $N_{t'} = N_t$ و $M_{t'} \geq M_t$. علاوه‌براین، فرض کنید $S_t \subseteq \{S_{t-1} \cup f_t\}$ زیرمجموعه انتخاب‌شده تا زمان t و f_t ویژگی ورودی در زمان t باشند. هدف از انتخاب برخط ویژگی‌های جریانی، انتخاب یک زیرمجموعه کمینه $S_t \subseteq \{S_{t-1} \cup f_t\}$ است؛ به‌گونه‌ای که برخی معیارهای وابستگی به C را بیشینه کند.

الگوریتم یک، رویه پیشنهادی جهت انتخاب برخط ویژگی‌های جریانی را نشان می‌دهد. این الگوریتم سعی در انتخاب بهترین ویژگی‌ها با استفاده از سری هندسی گراف ارتباط ویژگی‌ها دارد. این الگوریتم با یک مجموعه انتخابی خالی S کار خود را آغاز می‌کند؛ سپس منتظر یک ویژگی ورودی جدید می‌شود (خط ۴). زمانی که ویژگی جدید f_t مهیا می‌شود، این ویژگی به‌همراه تعداد K ویژگی خوبی که در زمان های قبلی حذف شده‌اند (این ویژگی‌ها در مجموعه R نگهداری می‌شوند) به مجموعه انتخابی فعلی (S) اضافه می‌شوند (خط ۵). در گام بعد، ماتریس مجاورت ویژگی‌های موجود در S بر اساس فرمول (۴) ایجاد شده (خط ۶) و سری هندسی منظم‌شدهٔ آن بر اساس فرمول (۳) محاسبه می‌شود (خطوط ۷ و ۸). درنهایت تمامی ویژگی‌هایی که معیار منظم‌شده آنها کمتر از ویژگی f_t باشد از S حذف شده و به ویژگی‌های افزونه (R) منتقل می‌شوند. گام‌های بالا تا زمان برآورده‌شدن برخی معیارهای توقف تکرار می‌شوند. بسته به نوع مسأله، از معیارهای توقف مختلفی می‌توان بهره جست. به‌عنوان مثال، اگر فضای ویژگی نامعلوم و یا بی‌نهایت باشد، معیار توقف می‌تواند رسیدن به یک حد مناسب از دقت رده‌بندی باشد. از طرفی دیگر، اگر فضای ویژگی معلوم باشد، می‌توان معیار توقف را مشاهده آخرین ویژگی در نظر گرفت.

$OSFS-GS(C,K)$
C: The class attribute
K: Number of relevant features for reconsideration

1: $S \leftarrow \{\}$ //the selected feature subset
2: $R \leftarrow \{\}$ //the redundant feature subset
3: do
4: $f_t \leftarrow$ Get_New_Feature() //wait for a new feature
5: $S \leftarrow S \cup f_t \cup R[1:K]$
6: $A[i,j] \leftarrow \alpha\left(max\left(I(f_i,Y), I(f_j,Y)\right)\right) + (1-\alpha)(1-|SPR(f_i,f_j)|), \forall f_i, f_j \in S$
7: $G \leftarrow (I-rA)^{-1} - I$
8: $\gamma_{f_i} \leftarrow [G\overline{1}]_i, \forall f_i \in S$
9: $S \leftarrow S - \{f_i\}, Add\left(\left\{R, f_i, \beta_{f_i}\right\}\right), \forall f_i \in S \text{ and } \gamma_{f_i} < \gamma_f$
10: until (stopping criterion is met)
11:

(الگوریتم-۱): رویه پیشنهادی جهت انتخاب برخط ویژگی‌های جریانی

(Algorithm-1): Proposed online streaming feature selection method

۴-۱- فهرست ویژگی‌های افزونه

در الگوریتم (۱) از یک فهرست مرتب R جهت نگهداری ویژگی‌های افزونه (ویژگی‌هایی که تابه‌حال از ویژگی‌های انتخابی حذف شده‌اند) استفاده شده است. دلیل نگهداری

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]
[DOI: 10.29252/jsdp.17.4.3]

چنین فهرستی آن است که ممکن است، ویژگی‌ای که در زمان t به‌دلیل حضور ویژگی f_t افزونه شده باشد، در یک زمان $t' > t$ به‌دلیل حذف ویژگی f_t افزونه نباشد. یک مشکل عمده در نگهداری چنین فهرستی، افزایش اندازه آن با گذشت زمان است. این افزایش باعث کاهش سرعت جستجو در یافتن بهترین ویژگی‌ها (خط ۵ در الگوریتم ۱) می‌شود. جهت حل این مشکل، فهرست R همواره بر اساس معیار زیر مرتب نگهداری می‌شود:

$$\beta(f) = I(f,C) - \frac{T_f}{\sum_{j\in R} T_{f_j}} \quad (۵)$$

که در آن T_f نشان‌دهنده تعداد دفعاتی است که ویژگی f از فهرست R به فهرست S منتقل شده ولی دوباره حذف شده است؛ بنابراین، اگر یک ویژگی چندین بار به فهرست S منتقل شده و دوباره حذف شود، دچار جریمه شده و از ابتدای فهرست R فاصله می گیرد.

۲-۴- پیچیدگی زمانی روش پیشنهادی

فرض کنید در زمان t، ویژگی جدید f_t وارد شود و نیز فرض کنید S_t و R_t به‌ترتیب نشان‌دهندهٔ مجموعه ویژگی‌های انتخابی و افزونه در زمان t باشند. نخستین گام محاسباتی پس از مشاهده f_t، محاسبه ماتریس مجاورت برای مجموعه $S_t \cup f_t \cup R[1..K]$ است. پیچیدگی زمانی این گام برابر $\Theta((|S_t|+K+1)^2)$ است. در گام بعد، محاسبه معیار منظم‌شده اتفاق می‌افتد. پیچیدگی زمانی این گام برابر $\Theta((|S_t|+K+1)^{2.37})$ است؛ درنهایت عمل Add اجرا می‌شود که دارای پیچیدگی زمانی $O(\log_2|R_t|)$ است؛ بنابراین، پیچیدگی زمانی الگوریتم پیشنهادی جهت پردازش هر ویژگی جدید ورودی به‌صورت زیر خواهد بود:

$$O((|S_t|+K+1)^2 + (|S_t|+K+1)^{2.37} + \log_2|R_t|) \quad (۶)$$

۵- نتایج تجربی

نتایج تجربی را می‌توان به دو دسته تقسیم‌بندی کرد. دسته نخست شامل آزمایش‌های اولیه جهت بررسی تأثیر مقادیر مختلف پارامتر K بر کارآیی روش پیشنهادی است. دسته دوم شامل آزمایش‌هایی جهت مقایسه الگوریتم پیشنهادی با سه الگوریتم مطرح در حوزه انتخاب برخط ویژگی‌های جریانی، شامل الگوریتم‌های سرمایه‌گذاری آلفا [18]، سریع [8] و مبتنی بر تئوری اطلاعات [13]، است.

در این مقاله از هشت مجموعه‌داده با ابعاد بزرگ استفاده شده است. این مجموعه‌داده‌ها به همراه منبع، اندازه و نوع آنها در جدول (۱) فهرست شده است. رده‌بندی داده‌ها با استفاده از SVM خطی انجام گرفته و جهت مقداردهی پارامتر C در این رده‌بند، از اعتبارسنجی پنج‌تایی استفاده شده است.

با توجه به عدم دسترسی به فضای کلی ویژگی‌ها، ترتیب ورود ویژگی‌ها در نتایج نهایی تأثیرگذار هستند. بنابراین، جهت افزایش دقت مقایسه‌ها، تعداد ۳۰ ترتیب ورود تصادفی برای هر مجموعه‌داده ایجاد و از یک آزمون فرض t جهت مقایسه نتایج بر روی این ترتیب‌ها استفاده شده است. فرض کنید d_A و d_B به‌ترتیب نمونه‌هایی از جوامع A و B باشند. آزمون‌های فرض t را به‌صورت زیر تعریف می‌کنیم:

$$t_1: \begin{cases} H_0: \mu_{d_A} = \mu_{d_B} \\ H_1: \mu_{d_A} > \mu_{d_B} \end{cases} \quad (۷)$$

$$t_1: \begin{cases} H_0: \mu_{d_A} = \mu_{d_B} \\ H_1: \mu_{d_B} > \mu_{d_A} \end{cases} \quad (۸)$$

که در آن μ_D میانگین جمعیت D است. حال بر اساس نتایج این دو آزمون، متغیر t را به‌صورت زیر تعریف می‌کنیم:

$$t = \begin{cases} \uparrow & if\ H_0\ in\ t_1 is\ rejected \\ \downarrow & if\ H_0\ in\ t_2 is\ rejected \\ = & otherwise \end{cases} \quad (۹)$$

(جدول-۱): فهرست مجموعه‌داده‌های مورد استفاده در نتایج تجربی

(Table-1): The benchmark datasets used in experimental results

مجموعه‌داده	تعداد ویژگی‌ها	تعداد نمونه ها	حوزه
][21 dorothea	100000	1600	ژنتیک
][21 arcene	10000	800	تشخیص سرطان
][21 dexter	20000	2300	رده بندی متن
][21 madelon	500	3800	نامعلوم
][15 tm1	100	1000	مصنوعی
][15 tm2	100	1000	مصنوعی
][22 VOC 2007	6096	9963	رده بندی تصویر
][23 VOC 2012	6096	22531	رده بندی تصویر

۱-۵- تأثیر پارامتر K بر کارآیی الگوریتم پیشنهادی

مهم‌ترین پارمتر در الگوریتم یک، تعداد ویژگی‌هایی است که از مجموعه R به مجموعه S اضافه می‌شوند. این انتقال به ویژگی‌های خوبی که حذف شده‌اند، شانس بررسی مجدد را می‌دهد. جداول (۲ و ۳) نتیجه تأثیر این پارامتر بر دقت

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]

[DOI: 10.29252/jsdp.17.4.3]

رده‌بند SVM خطی برای نمونه‌های زمانی (نسبت ویژگی‌های مشاهده شده) مختلف را نشان می‌دهد. همان‌طورکه از این دو جدول قابل مشاهده است، بررسی مجدد ویژگی‌های افزونه ($K > 0$) باعث بهبود چشم‌گیر دقت رده‌بند نسبت به حالتی که ویژگی‌های افزونه به‌طورکامل حذف می‌شوند ($K = 0$)، شده است. از میان مقادیر مختلف این پارامتر، $K = 1$ بیشترین افزایش دقت را نتیجه داده است. به نظر می‌رسد با افزایش بیشتر مقدار K، مجموعه انتخابی حاوی تعداد بیشتری ویژگی افزونه می‌شود که خود باعث افزایش اندازه مجموعه ویژگی انتخابی و درنتیجه کاهش دقت رده‌بندی می‌شود.

(جدول-۲): تأثیر پارامتر K بر دقت رده‌بند SVM در نمونه‌های زمانی مختلف برای مجموعه‌داده Arcene

(Table-2): The effect(s) of parameter K on the classification accuracy of SVM at different time instances on arcene dataset

Arcene					
٪ ویژگی‌های مشاهده شده					
100	75	50	25	10	K
62.1	65.7	59.7	61.4	59.8	**0**
71.2	**68.8**	**71.4**	71.9	**64.2**	**1**
67.2	67.3	67.1	72.0	63.1	**2**
67.1	66.9	65.3	65.5	58.9	**3**
59.8	59.0	60.7	60.6	58.4	**4**
60.2	57.1	57.4	60.3	57.4	**5**

(جدول-۳): تأثیر پارامتر K بر دقت رده‌بند SVM در نمونه‌های زمانی مختلف برای مجموعه‌داده Madelon

(Table-3): The effect(s) of parameter K on the classification accuracy of SVM at different time instances on madelon dataset

Madelon					
٪ ویژگی‌های مشاهده شده					
100	75	50	25	10	K
51.0	53.2	54.2	49.5	49.3	**0**
72.4	**67.6**	**67.8**	**66.9**	**58.7**	**1**
62.7	57.4	67.1	59.0	53.1	**2**
58.7	57.9	59.7	60.3	54.7	**3**
53.7	50.0	54.6	54.2	49.4	**4**
42.7	41.7	47.5	42.8	48.1	**5**

۲-۵- مقایسه با روش‌های پیشین

در این بخش، الگوریتم OSFS-GS را با الگوریتم‌های سرمایه‌گذاری آلفا [18]، fast-OSFS [8] و OSFSMI [13] مقایسه می‌کنیم. در الگوریتم OSFS-GS از $K = 1$ استفاده شده است. برای الگوریتم سرمایه‌گذاری آلفا، مقادیر پارامترها برابر با مقادیر پیش فرض آنها قرار داده شده است. در الگوریتم fast_OSFS، از آزمون‌های استقلال G^2 برای مجموعه داده‌هایی کاملاً گسسته (حاوی هیچ ویژگی پیوسته‌ای نباشد) و آزمون‌های z فیشر برای مجموعه‌داده‌هایی که حاوی ویژگی‌های پیوسته هستند، استفاده شده است. برای هر یک از این آزمون‌ها، سطح معناداری آماری (α) برابر ۰/۰۵ انتخاب شده است. در الگوریتم OSFSMI نیز مقدار مقدار پارامتر β برابر ۰/۱ قرار داده شده است.

۱-۲-۵- مقایسه از نظر تعداد ویژگی‌های انتخابی

جدول (۴)، تعداد ویژگی‌های انتخابی به‌وسیلهٔ الگوریتم‌های مورد مقایسه را فهرست کرده است. از نتایج این جدول مشخص است که الگوریتم سرمایه‌گذاری آلفا کوچک‌ترین زیرمجموعه‌ها را انتخاب می‌کند (به‌جز برای مجموعه داده‌های tm1 و tm2). نکته قابل ملاحظه در این جدول، تفاوت معنادار اندازه‌های زیرمجموعه‌های انتخابی توسط الگوریتم OSFS-GS برای دو مجموعه‌داده VOC2007 و VOC2012 است. این دو مجموعه‌داده از طریق ترکیب لایه‌های پایانی سه شبکه عمیق GoogleNet (هزار ویژگی)، ResNet (هزار ویژگی) و VGG-VD (۴۰۹۶ ویژگی) به‌دست آمده‌اند؛ در نتیجه، تعداد ویژگی‌های افزونه و نامرتبط در آنها به‌نسبه کم است؛ بنابراین، یک الگوریتم ایده‌آل باید بخش زیادی از این ویژگی‌ها را داشته باشد. اگرچه الگوریتم OSFS-GS زیرمجموعه‌هایی را انتخاب کرده که اندازه آنها بسیار کوچک‌تر از مجموعه اصلی است، ولی در مقایسه با سه الگوریتم دیگر، بسیار بهتر عمل کرده است. این رفتار را می‌توان با بررسی مجدد ویژگی‌های افزونه به‌وسیلهٔ الگوریتم پیشنهادی مرتبط دانست.

(جدول-۴): میانگین زیرمجموعه انتخابی در سی ترتیب ورود مختلف توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS. در این جدول، t نتیجه آزمون فرض رابطه (۹) را نشان می‌دهد و در آن A نتایج روش OSFS-GS و B نتایج یکی از سایر الگوریتم‌ها است.

(Table-4): Average sizes of the selected subsets for 30 different streaming orders using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS. In this table, t indicates the result of the hypothesis test defined in eq (9). In the tests, A is OSFS-GS and B is one of the other algorithms.

مجموعه داده	سرمایه‌گذاری آلفا		Fast-OSFS		OSFSMI		OSFS-GS
	میانگین	t	میانگین	t	میانگین	t	میانگین
dorothea	2.4	↑	8.5	=	15.7	↓	9.3
arcene	6.7	↑	8.6	↑	10.5	=	10.9
dexter	7.0	↑	12.8	↓	8.9	↑	10.7
madelon	3.7	↑	5.1	↑	15.2	↓	12.9
tm1	12.4	↓	23.4	↓	14.8	↓	9.6
tm2	11.9	↓	34.6	↓	17.3	↓	7.8
VOC 2007	5.9	↑	12.0	↑	19.5	↑	81.7
VOC 2012	8.1	↑	11.2	↑	21.8	↑	91.8

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]
[DOI: 10.29252/jsdp.17.4.3]

(جدول-۵): میانگین زمان‌های اجرا در سی ترتیب ورود مختلف توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS

(Table-5): Average running times for 30 different streaming orders alpha-investing, fast-OSFS, OSFSMI and OSFS-GS.

مجموعه داده	سرمایه‌گذاری آلفا		Fast-OSFS		OSFSMI		OSFS-GS
	میانگین	t	میانگین	t	میانگین	t	میانگین
dorothea	6.5	↑	8.6	↑	22.9	↑	33.8
arcene	1.8	↑	1.7	↑	12.8	↑	25.9
dexter	7.8	↑	5.9	↑	20.8	↑	32.1
madelon	0.8	↑	0.9	↑	3.8	↑	5.4
tm1	4.2	↑	0.8	↑	4.8	↑	7.5
tm2	3.7	↑	0.8	↑	4.9	↑	7.3
VOC 2007	2.0	↑	1.9	↑	11.8	↑	53.9
VOC 2012	2.3	↑	1.9	↑	13.5	↑	61.3

۲-۲-۵- مقایسه از نظر زمان‌های اجرا

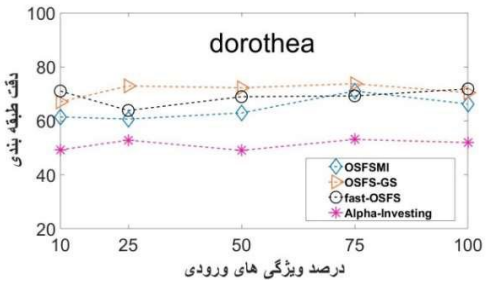
جدول (۵) میانگین زمان‌های اجرای چهار الگوریتم مورد مقایسه را نشان می‌دهد. چنان‌که قابل ملاحظه است، الگوریتم پیشنهادی بیشترین زمان اجرا را در میان این الگوریتم‌ها دارد. جهت توصیف این رخداد، می‌توان دو دلیل زیر را بیان کرد:

۱- الگوریتم پیشنهادی تنها الگوریتمی است که ویژگی‌های افزونه را نگهداری و مدیریت می‌کند. این عمل بار زمانی قابل توجهی را به حلقه‌های الگوریتم تحمیل می‌کند؛ به‌ویژه این که با گذشت زمان اندازه مجموعه ویژگی‌های افزونه بزرگ‌تر شده و نگهداری، به‌روزرسانی و جستجو در آن نیاز به زمان بیشتری دارد. چنان‌که درقبل گفته شد، این هزینه بابت بررسی مجدد ویژگی‌های مفیدی که حذف شده‌اند، اجتناب‌ناپذیر است؛ با این وجود، می‌توان با تعریف محدودیت بر روی بیشینه تعداد ویژگی‌های قابل نگهداری، بخش قابل توجهی از این بار زمانی را تقلیل داد؛ علاوه‌براین، می‌توان با ثابت‌کردن پارامتر K در این الگوریتم، نیاز به مرتب‌سازی فهرست ویژگی‌های افزونه را مرتفع ساخت.

۲- برای دو مجموعه داده VOC2007 و VOC2012، علاوه‌بر مدیریت ویژگی‌های افزونه، اندازه بزرگ مجموعه‌های انتخابی در طول جریان ویژگی‌ها، تأثیر بیشتری بر روی زمان اجرای الگوریتم پیشنهادی داشته است. به عبارت دیگر، با توجه به این که این دو مجموعه‌داده از وزن‌های شبکه عصبی عمیق استخراج شده‌اند، بخش بزرگی از ویژگی‌ها، مرتبط و غیرافزونه هستند؛ به همین دلیل، مجموعه ویژگی‌های انتخابی در طول زمان برای این دو مجموعه‌داده بسیار بزرگ است. درنتیجه محاسبه سری‌های هندسی برای این زیرمجموعه‌ها نیز بسیار زمان‌بر است. البته این مورد نشان‌دهنده ضعف الگوریتم‌های دیگر در تشخیص ویژگی‌های مناسب است.

۳-۲-۵- مقایسه از نظر دقت رده‌بندی

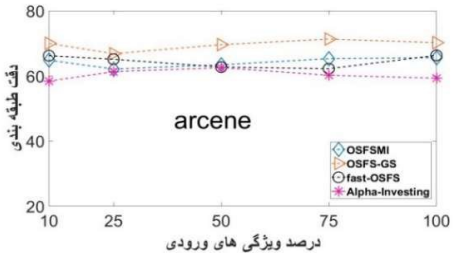
شکل‌های (۱ تا ۸)، دقت رده‌بند SVM خطی برای نمونه‌های زمانی مختلف را نشان می‌دهند. همان‌طور که از این شکل‌ها قابل ملاحظه است، الگوریتم پیشنهادی از نظر دقت رده‌بندی بسیار کارآمد عمل کرده است. این کارآمدی به‌ویژه در مجموعه داده‌های VOC2007 و VOC2012 محسوس است. برای مجموعه داده‌های dorothea، arcene، dexter، tm1 و tm2 نیز الگوریتم پیشنهادی بیش‌تر نمونه‌های زمانی بهتر از الگوریتم‌های دیگر عمل کرده است؛ علاوه‌براین، برای سایر مجموعه‌داده‌ها، این الگوریتم در زمره الگوریتم‌های با دقت بالا قرار گرفته است. درمجموع برای نمونه‌های زمانی ثبت‌شده در هشت مجموعه‌داده، روش پیشنهادی در ۲۶ نمونه زمانی از ۴۰ نمونه (۶۵ درصد نمونه‌های زمانی) بهتر از الگوریتم‌های دیگر عمل کرده است. علاوه‌براین، این الگوریتم تنها در هفت مورد (۱۷,۵ درصد نمونه‌های زمانی) قادر به تولید بهترین نتیجه نبوده است. در میان الگوریتم‌های مورد مقایسه، الگوریتم سرمایه‌گذاری آلفا دقت پایین‌تری دارد. دلیل چنین عملکردی را می‌توان به تفاوت مدل یادگیری آن با مدل SVM خطی مرتبط دانست (سرمایه‌گذاری آلفا یک روش انتخاب ویژگی جاسازی شده است).



(شکل-۱): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده dorothea

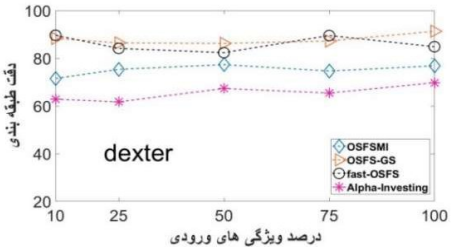
(Figure-1): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on dorothea

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]
[DOI: 10.29252/jsdp.17.4.3]



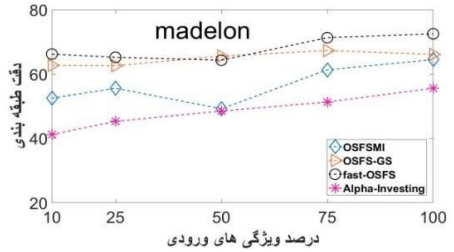
(شکل-۲): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده arcene

(Figure-2): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on arcene



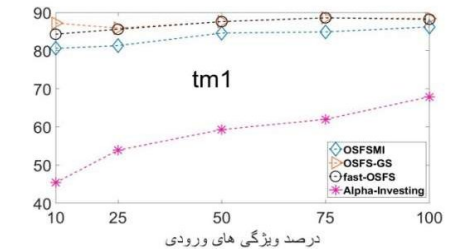
(شکل-۳): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده dexter

(Figure-3): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on dexter



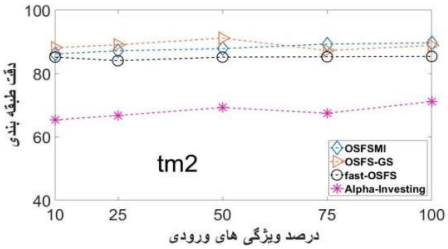
(شکل-۴): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده madelon

(Figure-4): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on madelon



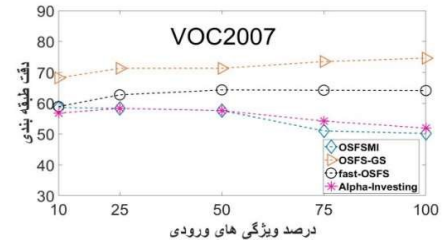
(شکل-۵): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده tm1

(Figure-5): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on tm1



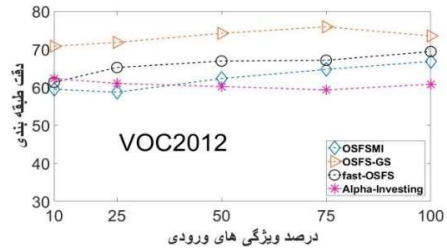
(شکل-۶): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده tm2

(Figure-6): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on tm2



(شکل-۷): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده VOC2007

(Figure-7): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on VOC2007



(شکل-۸): میانگین دقت‌های رده‌بندی در نمونه‌های زمانی متفاوت توسط الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS، OSFSMI و OSFS-GS برای مجموعه‌داده VOC2012

(Figure-8): The average classification accuracy at different time instances using alpha-investing, fast-OSFS, OSFSMI and OSFS-GS on VOC2012

۶- جمع‌بندی

در این مقاله، مسأله انتخاب برخط ویژگی‌های جریانی از منظر سری های هندسی گراف ارتباط ویژگی‌ها مورد بررسی قرار گرفت و یک الگوریتم جدید به نام OSFS-GS پیشنهاد شد. این الگوریتم با استفاده از مفهوم سری هندسی گراف مجاورت، ویژگی‌های افزونه را به‌شکل برخط حذف می‌کند؛ علاوه‌براین، الگوریتم پیشنهادی از یک سازوکار نگهداری ویژگی‌های افزونه بهره می‌برد که امکان بررسی مجدد ویژگی‌های بسیار خوبی را که درقبل حذف شده‌اند، فراهم

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]
[DOI: 10.29252/jsdp.17.4.3]

می‌آورد. الگوریتم پیشنهادی بر روی هشت مجموعه‌داده با ابعاد بزرگ اعمال شد و نتایج آن با الگوریتم‌های سرمایه‌گذاری آلفا، fast-OSFS و OSFSMI مورد مقایسه قرار گرفت. نتایج، نشان‌دهنده دقت بالای الگوریتم پیشنهادی در نمونه‌های زمانی مختلف بود.

۷- مراجع 7- References

[1] S. Eskandari and M. Javidi, "Online streaming feature selection using rough sets," *International Journal of Approximate Reasoning*, vol. 69, pp. 35-57, 2016.

[2] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, pp. 1157-1182, 2003.

[3] S. Eskandari and M. Javidi, "Streamwise feature selection: a rough set method," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 4, pp. 667-676, 2016.

[4] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997.

[5] Y. Saeys, I. Iñaki , and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507-2517, 2007.

[6] V. Bolón-Canedo, S. Noelia , and A. Amparo, "A review of feature selection methods on synthetic data," *Knowledge and information systems*, vol. 34, no. 3, pp. 483-519, 2013.

[7] J.Wang, Zh. Peilin , H. CH Steven , and J. Rong , "Online feature selection and its applications," *IEEE Transactions on Knowledge and Data Engineering* , vol. 29, no. 3, pp. 698-710, 2014.

[8] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 1178-1192, 2013.

[9] S. Perkins and J. Theiler, "Online feature selection using grafting," *in Proceedings of the 20th International Conference on Machine Learning*, 2003, pp. 592-599.

[10] K. Glocer, D. Eads, and J. Theiler, "Online feature selection for pixel classification," *in Proceedings of the 22nd international conference on Machine learning*, pp. 249-256.

[11] J. Zhou, D. P. Foster, R. A Stine, and L. H Ungar, "Streamwise feature selection," *Journal of Machine Learning Research*, vol. 7, pp. 1861-1885, 2006.

[12] M. Javidi and S. Eskandari, "Online streaming feature selection: a minimum redundancy, maximum significance approach," *Pattern Analysis and Applications*, pp. 1-15, 2018.

[13] M. Rahmaninia and P. Moradi, "OSFSMI: Online stream feature selection method based on mutual information," *Applied Soft Computing*, vol. 68, pp. 733-746, 2018.

[14] S. Eskandari and E. Akbas, "Supervised Infinite Feature Selection," CoRR arXiv, 2017.

[15] S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *Journal of machine learning research*, vol. 3, pp. 1333-1356, 2003.

[16] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern recognition letters*, vol. 15, no. 11, pp. 1119-1125, 1994.

[17] L. H Ungar, J. Zhou, D. P Foster, and B. A Stine, "Streaming Feature Selection using IIC," in Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, 2005.

[18] J. Zhou, D. Foster, R. Stine, and L. Ungar, "Streaming feature selection using alpha-investing," *in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 384-393.

[19] P. S Dhillon, D. Foster, and L. Ungar, "Feature selection using multiple streams," in Proceedings of International Workshop on Artificial Intelligence and Statistics, 2010.

[20] G. Roffo, S. Melzi, and M. Cristani, "Infinite feature selection," *in Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4202-4210.

[21] Isabelle Guyon. (2003) http://clopinet.com/isabelle/Projects/NIPS2003/.

[22] M. Everingham, L. Van Gool, Ch. Williams, J. Winn, and A. Zisserman, The pascal visual object classes (voc) challenge, 2007.

[23] M. Everingham, L. Van Gool, Ch. K. Williams, J. Winn, and A. Zisserman, The pascal visual object classes (voc) challenge, 2012.

[24] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Information Sciences*, vol. 179, no. 13, pp. 2208-2217, 2009.

[25] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537-550, 1994.

[26] G. Brown, A. Pocock, M. Zhao, and M. Luján, "Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection," *The Journal of Machine Learning Research*, vol. 13, pp. 27-66, 2012.

[27] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artificial intelligence*, vol. 151, no. 1-2, pp. 155-176, 2003.

[28] P. Zhao and B. Yu, "On model selection consistency of Lasso," *Journal of Machine learning research*, no. 7, pp. 2541-2563, 2006.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]

[DOI: 10.29252/jsdp.17.4.3]



صادق اسکندری مدرک دکترای خود را در رشته ریاضی کاربردی و مدرک کارشناسی ارشد در رشته علوم رایانه از دانشگاه شهید باهنر کرمان دریافت کرده و ایشان هم‌اکنون استادیار گروه علوم رایانه دانشگاه گیلان است. زمینه‌های پژوهشی ایشان شامل یادگیری ماشین، یادگیری عمیق و انتخاب ویژگی هستند.
نشانی رایانامه ایشان عبارت است از:
eskandari@guilan.ac.ir

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]

[DOI: 10.29252/jsdp.17.4.3]

[DOI: 10.29252/jsdp.17.4.3] [Downloaded from jsdp.rcisp.ac.ir on 2026-11-14]