

نگاشت ویژگی صریح دقیق متناهی البعد برای توابع هسته

سید کمال‌الدین غیائی شیرازی*

استادیار، دانشکده مهندسی کامپیوتر- دانشگاه فردوسی مشهد- مشهد- ایران

پست الکترونیکی: k.ghiasi@um.ac.ir

محمد رضا قرائی

دانشجوی دکتری، دانشکده مهندسی کامپیوتر- دانشگاه آلتو- اسپو - فنلاند

پست الکترونیکی: mohammadreza.mohammadniaqaraei@aalto.fi

چکیده

نسخه هسته‌ای یک الگوریتم را در فرم اولیه و بدون نیاز به حقه هسته و نمایش دوگان به دست آوریم. به عنوان اولین کاربرد، ما نشان می‌دهیم که چطور می‌توان، بدون توسل به نمایش دوگان، نسخه هسته‌ای الگوریتم‌های یادگیری ماشین را به دست آورد و این روش را به صورت مشخص در مورد PCA به کار می‌بریم. به عنوان کاربردی دیگر از روش پیشنهادی، ما بدون انجام هیچ تغییری در الگوریتم t-SNE، از کد آماده آن برای بصری‌سازی فضای ویژگی توابع هسته استفاده می‌کنیم.

واژه‌های کلیدی: روش‌های هسته، نگاشت ویژگی

صریح دقیق، فضای ویژگی متناهی البعد

۱-مقدمه

روش‌های هسته یکی از تکنیک‌های مهم یادگیری ماشین است که در کنار تکنیک‌های دیگری چون یادگیری بیزی، شبکه‌های عصبی و یادگیری عمیق و با ترکیب با آن‌ها نقش مهمی در شکل‌گیری یک راه حل مناسب برای مسائل یادگیری ماشین ایفا می‌کند. اساس روش‌های هسته جایگزینی تمام موارد ضرب داخلی در یک الگوریتم یادگیری ماشین با یک تابع هسته است که به آن حقه هسته

روش‌های هسته در یادگیری ماشین، از یک تابع هسته استفاده می‌کنند که به عنوان ورودی دو داده را دریافت می‌کند و حاصل ضرب داخلی آن‌ها پس از نگاشت به یک فضای هیلبرت را، به صورت ضمنی و بدون این که چنین نگاشتی به فضای هیلبرت واقعا محاسبه شود، برمی‌گرداند. برای بسیاری از توابع هسته، همانند هسته‌های گوسی و لاپلاسی، فضای ویژگی شناخته شده بی‌نهایت بعدی است و به همین دلیل گفته می‌شود که عملیات در فضای هیلبرت این توابع هسته تنها به صورت ضمنی ممکن است. ضمنی بودن این فضا باعث می‌شود که ما مجبور باشیم الگوریتم‌ها را در فضایی که تصویری از آن نداریم با استفاده از نمایش دوگان و حقه هسته بیان کنیم. ما در این مقاله برای هر تابع هسته دلخواه یک فضای ویژگی متناهی البعد و نگاشتی صریح و دقیق به این فضای ویژگی معرفی می‌کنیم که در حد عملیات مورد نیاز در الگوریتم‌های یادگیری ماشین هسته‌ای، هم در زمان آموزش و هم آزمون، حاصل ضرب داخلی داده‌ها در این فضای ویژگی صریح با مقدار تابع هسته برابر است. وجود این نگاشت صریح به فضای ویژگی به ما این امکان را می‌دهد که

* نویسنده مسئول

گفته می‌شود. توابع هسته دارای این خاصیت هستند که متناظر با آن‌ها حداقل یک فضای هیلبرت و یک نگاشت از فضای ورودی به آن فضای هیلبرت وجود دارد به نحوی که مقدار برگشتی توسط تابع هسته برابر حاصل ضرب داخلی نگاشت داده‌های ورودی در آن فضای هیلبرت می‌باشد. به‌طور مشخص، برای هر تابع هسته $k: X \times X \rightarrow \mathbb{R}$ که بر روی فضای ورودی X تعریف شده است، فضای هیلبرت \mathcal{H} و نگاشت $\phi: X \rightarrow \mathcal{H}$ وجود دارند به نحوی که برای هر $x, z \in X$ داریم: $k(x, z) = \langle \phi(x), \phi(z) \rangle_{\mathcal{H}}$. نکته مهم این است که تابع هسته معمولاً یک عبارت ریاضی ساده است که مقدار آن با حاصل ضرب داخلی پس از نگاشت به فضای هیلبرت برابر است و چنین نگاشتی معمولاً به صورت صریح انجام نمی‌شود. برای مثال تابع هسته گوسی که با عبارت ساده زیر بیان می‌شود، دارای یک فضای هیلبرت بی‌نهایت بعدی است.

$$k(x, z) = \exp\left(-\frac{1}{2\sigma^2} \|x - z\|^2\right)$$

به دلایل مختلف همواره فرض شده است که نگاشت صریح دادگان به فضای هیلبرت متناظر با یک تابع هسته در حالت کلی ممکن نیست. اولاً بسیاری از فضاهای هیلبرت شناخته شده متناظر با توابع هسته، همانند RKHS [۱] و فضای هیلبرت هسته‌های مرسر [۲ صفحه ۳۶] فضاهای ریاضی با اشیائی از نوع تابع و یا دنباله‌های نامتناهی هستند که برای یک نگاشت صریح محاسباتی مناسب نیستند. ثانیاً بجز در موارد خاص، فضاهای هیلبرت متناظر با توابع هسته معمولاً بی‌نهایت بعدی هستند که باعث می‌شود این تصور به وجود آید که امکان محاسبه صریح نگاشت از دیدگاه محاسباتی وجود ندارد.

با این وجود با فرض امکان محاسبه نگاشت ویژگی صریح، می‌توان درکی از فضای ویژگی توابع هسته به دست آورد که به انتخاب بهتر تابع هسته و در نهایت بهبود الگوریتم یادگیری ماشین کمک خواهد کرد. همچنین در سال‌های اخیر از نگاه کاهش محاسبات در روش‌های هسته، روش‌هایی برای به دست آوردن نگاشت ویژگی

صریح مستقل از تعداد دادگان البته به شکل تقریبی ارائه شده است [۳]. وجود فضای ویژگی صریح این امکان را فراهم می‌کند که الگوریتم یادگیری در فرم اولیه بیان شود و نیازی به فرم دوگان نباشد. در نتیجه می‌توان پارامترهای ماشین یادگیری را صراحتاً یاد گرفت و نیازی به بیان پاسخ بر حسب دادگان آموزشی نیست. از روش‌های مطرح در تقریب نگاشت ویژگی تابع هسته می‌توان به نیستروم اشاره کرد [۴]. این روش با تقریب زدن توابع ویژه تابع هسته و محدود کردن بسط تابع هسته بر حسب توابع ویژه به تعدادی متناهی جمله با بیشترین مقادیر ویژه، یک نگاشت ویژگی صریح تقریبی وابسته به دادگان معرفی می‌کند. روش مطرح دیگر در تقریب نگاشت توابع هسته، روش نگاشت ویژگی فوریه تصادفی^۱ است [۵]. این روش، نگاشت ویژگی توابع هسته ناوردان نسبت به انتقال را با نمونه برداری از تبدیل فوریه این توابع به دست می‌آورد. در مرجع [۶] روش‌های تقریب نگاشت هسته به خانواده هسته‌های جمعی و در مرجع [۷] به خانواده هسته‌های شباهت تک نمونه^۲ تعمیم داده شده‌اند. همچنین برای کاهش ابعاد فضای ویژگی در روش‌های تقریب نگاشت ویژگی، مرجع [۸] استفاده از تصویرسازی تصادفی بردارها در فضای ویژگی تقریبی برای نگاشت خطی آن‌ها به فضایی با ابعاد کمتر را پیشنهاد داده است. با وجود بهبود زمانی محاسبات الگوریتم‌های مبتنی بر هسته در روش‌های فوق‌الذکر، تمام این روش‌ها نگاشت ویژگی تابع هسته را به شکل تقریبی محاسبه می‌کنند که این ذهنیت را به وجود آورده که امکان محاسبه دقیق نگاشت ویژگی وجود ندارد.

ما در این مقاله روشی برای محاسبه دقیق نگاشت ویژگی صریح هر تابع هسته دلخواه در روش‌های یادگیری ماشین مبتنی بر هسته ارائه می‌کنیم. به بیان دقیق‌تر، ما برای هر الگوریتم یادگیری ماشین که بر روی دادگان آموزشی $x^{(1)}, \dots, x^{(N)}$ آموزش دیده است، یک فضای هیلبرت متناهی

1- Random Fourier feature map
2- One-shot similarity kernels

۲- نگاهت و ویژگی صریح، دقیق و متناهی البعد پیشنهادی

فرض کنید که تابع هسته k و مجموعه دادگان آموزشی x_1, \dots, x_N داده شده است. اگر \mathcal{H} یک فضای هیلبرت متناظر با این تابع هسته و $\psi: X \rightarrow \mathcal{H}$ یک نگاشت از فضای ورودی به این فضای هیلبرت باشد، آنگاه برای هر دو داده دلخواه $x, z \in X$ داریم:

$$k(x, z) = \langle \phi(x), \phi(z) \rangle_{\mathcal{H}}$$

ما به دنبال یافتن یک فضای ویژگی متناهی البعد F و یک نگاشت صریح به صورت $\phi: X \rightarrow F$ هستیم به نحوی که برای هر زوج (x_n, z) که x_n یک داده آموزشی و z یک داده دلخواه است (که می‌تواند یک داده آموزشی نیز باشد) داشته باشیم:

$$k(x_n, z) = \langle \phi(x_n), \phi(z) \rangle_F$$

شرط فوق تضمین می‌کند که عملکرد ماشین یادگیر غیر هسته‌ای در فضای متناهی البعد F هم در زمان آموزش و هم در زمان تست با عملکرد آن در فضای هیلبرت \mathcal{H} یکسان باشد.

قضیه: فرض کنید K ماتریس هسته بر روی N داده آموزشی باشد. نگاشت $\phi: X \rightarrow \mathbb{R}^N$ تعریف شده در زیر خاصیت مورد نظر را دارد.

$$\phi(z) = K^{-\frac{1}{2}} [k(x_1, z), \dots, k(x_N, z)]^T$$

یعنی برای هر زوج (x_n, z) که x_n یک داده آموزشی و z یک داده دلخواه است داریم:

$$\langle \phi(x_n), \phi(z) \rangle = k(x_n, z)$$

اثبات: داریم:

$$\langle \phi(x_n), \phi(z) \rangle = \phi(x_n)^T \phi(z) = [k(x_1, x_n), \dots, k(x_N, x_n)] K^{-1} [k(x_1, z), \dots, k(x_N, z)]^T$$

با توجه به این که $[k(x_1, x_n), \dots, k(x_N, x_n)]$ سطر n ام از ماتریس K است، حاصل ضرب آن در ماتریس معکوس برابر برداری می‌شود که تنها در درایه n ام مقدار یک دارد و در بقیه درایه‌ها مقدار آن صفر است. بنابراین:

$$\langle \phi(x_n), \phi(z) \rangle = [k(x_1, x_n), \dots, k(x_N, x_n)] K^{-1} [k(x_1, z), \dots, k(x_N, z)]^T = k(x_n, z)$$

که همان مقدار مطلوب است.

البعد و یک نگاهت صریح معرفی می‌کنیم به نحوی که مقدار تابع هسته بین هر ورودی دلخواه و یکی از دادگان آموزشی با ضرب داخلی بین آن‌ها در آن فضای ویژگی برابر باشد. لازم به ذکر است که در یک الگوریتم یادگیری ماشین همواره یکی از ورودی‌های تابع هسته یکی از دادگان آموزشی است. در حقیقت در مرحله آموزش هر دو ورودی تابع هسته از دادگان آموزشی هستند و در مرحله آزمون یکی از ورودی‌ها از دادگان آموزشی است و ورودی دیگر داده دلخواه آزمون است. بنابراین روش ارائه شده که در آن فرض شده است یکی از ورودی‌ها حتماً از داده‌های آموزش است، برای محاسبه صریح و دقیق نگاهت ویژگی توابع هسته در الگوریتم‌های یادگیری ماشین کفایت می‌کند. لازم به ذکر است که در مقاله [۹] نیز ادعای یک نگاهت ویژگی صریح و دقیق مطرح شده است. اما کاری که واقعا در این مقاله انجام شده است این است که یک نگاهت صریح معرفی شده است و براساس ضرب داخلی در آن فضا یک تابع هسته خاص تعریف می‌شود. بدیهی است که نگاهت ویژگی معرفی شده برای تابع هسته تعریف شده یک نگاهت صریح است اما صرفاً برای تابع هسته معرفی شده قابل استفاده است و قابل تعمیم به توابع هسته دیگر نیست.

در ادامه مقاله، در بخش ۲ ما نگاهت و ویژگی صریح، دقیق و متناهی البعد پیشنهادی را معرفی می‌کنیم. در بخش ۳ ما پاسخ نسخه هسته‌ای الگوریتم PCA را در فرم اولیه و دوگان با استفاده از نگاهت ویژگی صریح معرفی شده به دست می‌آوریم و بعد با ترکیب پاسخ‌های فرم اولیه و دوگان به یک پاسخ جدید که از نظر محاسباتی مناسب‌تر است می‌رسیم. در بخش ۴ ما با استفاده از ابزار t-SNE فضای ویژگی توابع هسته را بر روی دادگان MNIST بصری‌سازی می‌کنیم و نشان می‌دهیم که مناسب بودن/نبودن تابع هسته برای یک مجموعه دادگان در این بصری‌سازی مشهود است.

۳- تحلیل PCA هسته‌ای با استفاده از نگاشت صریح و دقیق معرفی شده

روش کلاسیک برای به‌دست آوردن نسخه هسته‌ای الگوریتم PCA در مرجع [۱۰] شرح داده شده است که ما برای کامل بودن مطلب خلاصه آن را در ضمیمه الف آورده‌ایم. در این بخش ما با استفاده از نگاشت صریح و دقیق معرفی شده در این مقاله ابتدا نسخه هسته‌ای PCA را در فرم اولیه حل می‌کنیم. سپس نشان می‌دهیم که نگاشت صریح و دقیق معرفی شده در این مقاله ما را از حقه هسته بی‌نیاز می‌کند و می‌توانیم پاسخ دوگان را نیز به شیوه‌ای جدید به‌دست آوریم. در نهایت با ترکیب پاسخ‌های فرم اولیه و فرم دوگان به یک پاسخ جدید می‌رسیم که از نظر محاسباتی ساده‌تر است.

۳-۱- انجام تحلیل PCA در فضای ویژگی صریح و دقیق تابع هسته به‌صورت مستقیم

برای انجام تحلیل PCA در فضای ویژگی لازم است که ابتدا میانگین داده‌های آموزشی را در فضای ویژگی صفر کنیم. با توجه به این‌که ما دادگان را در فضای ویژگی صراحتاً داریم، می‌توانیم میانگین آن‌ها را چنین محاسبه کنیم

$$M = \frac{1}{N} \sum_{n=1}^N \phi(x_n) = \frac{1}{N} \sum_{n=1}^N K^{-\frac{1}{2}} [k(x_1, x_n), \dots, k(x_N, x_n)]^T$$

$$= \frac{1}{N} K^{-\frac{1}{2}} \sum_{n=1}^N [k(x_1, x_n), \dots, k(x_N, x_n)]^T$$

$$= K^{-\frac{1}{2}} Ke = K^{\frac{1}{2}} e$$

که e یک بردار N بعدی است که تمام درایه‌های آن $1/N$ است. بنابراین اگر ψ نگاشت جدیدی باشد که علاوه بر بردن دادگان به فضای ویژگی میانگین آن‌ها را هم صفر می‌کند، می‌توان ψ را چنین بیان کرد:

$$\psi(z) = \phi(z) - M = K^{-\frac{1}{2}} ([k(x_1, z), \dots, k(x_N, z)]^T - Ke)$$

دقت کنید که برخلاف شیوه متداول در روش‌های هسته که ابتدا محاسبات برحسب نگاشت ناشناخته ϕ

نوشته می‌شود و سپس ضرب داخلی در فضای هیلبرت با محاسبه تابع هسته جایگزین می‌شود، در این روش جدید نگاشت $\phi(z)$ صراحتاً مشخص است و از ابتدا تمام محاسبات مبتنی بر دادگان آموزشی و آزمون در فضای ویژگی است که خود بر حسب تابع هسته بیان شده‌اند.

به‌طور خاص، برای هر داده آموزشی x_n داریم:

$$\psi(x_n) = K^{-\frac{1}{2}} ([k(x_1, x_n), \dots, k(x_N, x_n)]^T - Ke)$$

$$= K^{-\frac{1}{2}} (K_{*n} - Ke) = K^{\frac{1}{2}} (e_n - e)$$

که K_{*n} ستون n ام ماتریس K می‌باشد و e_n برداری N بعدی است که دقیقاً درایه n ام آن یک و بقیه درایه‌های آن صفر هستند.

برای انجام تحلیل PCA ابتدا ماتریس کواریانس را در فضای ویژگی محاسبه می‌کنیم:

$$C = \frac{1}{N} \sum_{n=1}^N \psi(x_n) \psi(x_n)^T = \frac{1}{N} \sum_{n=1}^N K^{\frac{1}{2}} (e_n - e) (e_n - e)^T K^{\frac{1}{2}}$$

$$= \frac{1}{N} K^{\frac{1}{2}} \left(\sum_{n=1}^N (e_n - e) (e_n - e)^T \right) K^{\frac{1}{2}}$$

$$= \frac{1}{N} K^{\frac{1}{2}} \left(\sum_{n=1}^N e_n e_n^T - e_n e^T - e e_n^T + e e^T \right)$$

$$= \frac{1}{N} K^{\frac{1}{2}} (I - N e e^T) K^{\frac{1}{2}}$$

حال معادله $Cv = \lambda v$ در تحلیل PCA را در فضای ویژگی صریح می‌نویسیم:

$$K^{\frac{1}{2}} (I - N e e^T) K^{\frac{1}{2}} v = N \lambda v \quad (1)$$

با ضرب دو طرف رابطه فوق از سمت چپ در $K^{\frac{1}{2}}$ داریم:

$$K (I - N e e^T) K^{\frac{1}{2}} v = N \lambda K^{\frac{1}{2}} v$$

که اگر قرار دهیم $u = K^{\frac{1}{2}} v$ به رابطه زیر می‌رسیم:

$$K (I - N e e^T) u = N \lambda u$$

با تعریف $\bar{K} = K (I - N e e^T)$ به مسئله زیر می‌رسیم:

$$\bar{K} u = N \lambda u$$

باید توجه کنیم که در تحلیل PCA بردارهای v نرمال هستند، نه بردارهای u . بنابراین با این‌که u در سیستم ویژه فوق صدق می‌کند، اندازه آن یک نیست و از شرط $v^T v = 1$ نتیجه می‌شود که

$$v^T v = \left(K^{-\frac{1}{2}} u \right)^T \left(K^{-\frac{1}{2}} u \right) = u^T K^{-1} u = 1$$

بنابراین اندازه بردار u باید به نحوی انتخاب شود که رابطه بالا برقرار باشد.

برای داده آزمون z با توجه به این که ما نگاهت صریح به فضای ویژگی را داریم، ویژگی متناظر با مؤلفه اصلی v برابر است با

$$\begin{aligned} \langle \psi(z), v \rangle &= \left(K^{-\frac{1}{2}}([k(x_1, z), \dots, k(x_N, z)]^T - Ke), v \right) = \\ &= \left(K^{-\frac{1}{2}}([k(x_1, z), \dots, k(x_N, z)]^T - Ke), K^{-\frac{1}{2}}u \right) \\ &= ([k(x_1, z), \dots, k(x_N, z)] - e^T K) K^{-\frac{1}{2}} K^{-\frac{1}{2}} u = ([k(x_1, z), \\ &\dots, k(x_N, z)] - e^T K) K^{-1} u \end{aligned}$$

این شبیه پاسخی است که در فرآیندهای گوسی [۱۱] به دست می آید و شامل معکوس ماتریس هسته است.

به این ترتیب ما توانستیم رابطه KPCA را با محاسبه مستقیم PCA در فضای ویژگی به دست آوریم. از دیدگاه محاسباتی حتی می توان واقعا مجموعه دادگان آموزشی را به فضای ویژگی نگاهت کرد و بعد PCA معمولی را در آنجا محاسبه کرد.

۲-۳- انجام تحلیل PCA در فضای ویژگی تابع هسته با نمایش دوگان بدون نیاز به حقه هسته

اگر از نمایش دوگان استفاده کنیم و بردار v ترکیبی خطی از دادگان آموزشی در نظر بگیریم، داریم:

$$v = \sum_{n=1}^N \alpha_n \psi(x_n) = \sum_{n=1}^N \alpha_n K^{\frac{1}{2}}(e_n - e) = K^{\frac{1}{2}}(I - Nee^T)\alpha$$

حال اگر مسئله بردار ویژه مقدار ویژه (۱) را با استفاده از رابطه فوق بازنویسی کنیم، داریم:

$$\begin{aligned} K^{\frac{1}{2}}(I - Nee^T)K^{\frac{1}{2}}v &= N\lambda v \Rightarrow \\ K^{\frac{1}{2}}(I - Nee^T)K^{\frac{1}{2}}K^{\frac{1}{2}}(I - Nee^T)\alpha &= N\lambda K^{\frac{1}{2}}(I - Nee^T)\alpha \\ \Rightarrow K^{\frac{1}{2}}(I - Nee^T)K(I - Nee^T)\alpha &= N\lambda K^{\frac{1}{2}}(I - Nee^T)\alpha \end{aligned}$$

با ضرب دوطرف از چپ در $K^{\frac{1}{2}}$ به رابطه زیر می رسیم.
 $K(I - Nee^T)K(I - Nee^T)\alpha = N\lambda K(I - Nee^T)\alpha$
 با استفاده از تعریف $\bar{K} = K(I - Nee^T)$ رابطه فوق را

می توان چنین ساده تر نوشت:

$$\bar{K}\bar{K}\alpha = N\lambda\bar{K}\alpha$$

می توان ثابت کرد (مشابه [۱۰]) که حل رابطه زیر کفایت می کند

$$\bar{K}\alpha = N\lambda\alpha$$

در این معادله اندازه α باید طوری انتخاب شود که بردار v نرمال باشد. بنابراین α باید در رابطه زیر صدق کند

$$\begin{aligned} \|v\|^2 &= \left(K^{\frac{1}{2}}(I - Nee^T)\alpha, K^{\frac{1}{2}}(I - Nee^T)\alpha \right) \\ &= \alpha^T (I - Nee^T) K^{\frac{1}{2}} K^{\frac{1}{2}} (I - Nee^T) \alpha \\ &= \alpha^T (I - Nee^T) K (I - Nee^T) \alpha = 1 \end{aligned}$$

برای داده آزمون z ویژگی متناظر با مؤلفه اصلی v چنین به دست می آید:

$$\begin{aligned} \langle \psi(z), v \rangle &= \left(K^{-\frac{1}{2}}([k(x_1, z), \dots, k(x_N, z)]^T - Ke), K^{\frac{1}{2}}(I - Nee^T)\alpha \right) \\ &= \left(K^{-\frac{1}{2}}([k(x_1, z), \dots, k(x_N, z)]^T - Ke), K^{\frac{1}{2}}(I - Nee^T)\alpha \right) \\ &: [k(x_1, z), \dots, k(x_N, z)](I - Nee^T)\alpha - e^T K(I - Nee^T)\alpha \end{aligned}$$

۳-۳- مقایسه پاسخ های فرم اولیه و دوگان

نکته جالب این است که هر دو فرم اولیه و دوگان به حل مسئله محاسبه بردارهای ویژه ماتریس \bar{K} رسیدند. این مطلب عجیب است و لازم است که ابتدا نشان دهیم که هر دو پاسخ به دست آمده با هم معادل هستند. به عبارت دیگر باید نشان دهیم که بردارهای u و α ضریبی از یکدیگر هستند. از روابط $u = K^{\frac{1}{2}}v$ و $v = K^{\frac{1}{2}}(I - Nee^T)\alpha$ نتیجه می شود که

$$u = K(I - Nee^T)\alpha = \bar{K}\alpha = N\lambda\alpha$$

بنابراین می بینیم که بردار پاسخ اولیه u و بردار پاسخ دوگان α ضریبی از یکدیگر هستند. به این ترتیب ما می توانیم $\langle \psi(z), v \rangle$ را به صورت ترکیبی از پاسخ های فرم اولیه و دوگان به صورت زیر بنویسیم که ما را از محاسبه ماتریس هسته ای که میانگین دادگان در آن به مرکز دستگاه مختصات آورده شده است، بی نیاز می کند.

$$\begin{aligned} \langle \psi(z), v \rangle &= [k(x_1, z), \dots, k(x_N, z)]\alpha \\ &- e^T \alpha ([k(x_1, z), \dots, k(x_N, z)]e - N\lambda) \end{aligned}$$

۴- بصری سازی فضای ویژگی توابع هسته با t-SNE [۱۲]

الگوریتم t-SNE یک الگوریتم بسیار موفق در بصری سازی فضاهای ابعاد بالا می باشد. اگر شخص با

جزئیات این الگوریتم به صورت کامل آشنا باشد، می تواند الگوریتم t-SNE را تغییر دهد و فواصل بین دادگان در فضای ویژگی یک تابع هسته را برحسب تابع هسته بیان کند و از این نسخه اصلاح شده t-SNE برای بصری سازی فضای ویژگی استفاده کند. فاصله بین نگاشت دادگان x و z در فضای ویژگی چنین قابل محاسبه است:

$$\|\phi(x) - \phi(z)\|^2 = \langle \phi(x), \phi(x) \rangle + \langle \phi(z), \phi(z) \rangle - 2\langle \phi(x), \phi(z) \rangle = k(x, x) + k(z, z) - 2k(x, z)$$

اما روش پیشنهادی اجازه می دهد که بدون هیچ تغییری در کد t-SNE و صرفاً با تغییر دادگان ورودی به الگوریتم، فضای ویژگی یک تابع هسته بصری سازی شود. داشتن یک فضای ویژگی صریح متناهی البعد این امکان را به وجود می آورد که t-SNE در فضای ویژگی اجرا شود و نمایش دوبعدی مناسبی را از چینش دادگان در فضای ویژگی ارائه کند. در این بخش ما دادگان MNIST را در فضای ویژگی توابع هسته مختلف بصری سازی می کنیم. بهترین نتایج بر روی دادگان MNIST توسط روش های هسته با یک انتخاب هوشمندانه برای تابع هسته چندجمله ای به دست آمده است که در صفحه ۳۴۱ مرجع [۲] بیان شده است. ما در اینجا این انتخاب هوشمندانه برای هسته چندجمله ای را با انتخاب معمول هسته چندجمله ای مقایسه می کنیم و با بصری سازی فضای ویژگی آن ها نشان می دهیم که دادگان در فضای ویژگی هسته چندجمله ای مرجع [۲] جداپذیری بسیار بهتری را دارند. لازم به ذکر است که امکان بصری سازی فضای ویژگی برای توابع هسته توسط نگاشت ویژگی صریح و دقیق معرفی شده در این مقاله فراهم شده است. دادگان MNIST شامل ۶۰,۰۰۰ داده آموزشی و ۱۰,۰۰۰ داده آزمون از ارقام ۰ تا ۹ است. ارائه کنندگان الگوریتم t-SNE از ۲۵۰۰ نمونه از دادگان MNIST برای بصری سازی دادگان در همان فضای ورودی استفاده کردند. ما رقم های ۲، ۴ و ۷ از این دادگان را در نظر می گیریم و از هر کدام ۵۰۰ نمونه را از دادگان آموزش و ۵۰۰ نمونه را از دادگان

آزمون انتخاب می کنیم. به این ترتیب ۱۵۰۰ داده آموزشی و ۱۵۰۰ داده آزمون به دست می آید. ما از دادگان آموزشی برای آموزش t-SNE استفاده می کنیم و در نهایت دادگان آزمون را با استفاده از این نگاشت بصری سازی می کنیم. دادگان MNIST شامل تصاویر 28×28 می باشد که شدت رنگ هر پیکسل عددی بین ۰ تا ۲۵۵ است. ابتدا شدت رنگ پیکسل ها با تقسیم ۲۵۵ به عددی بین ۰ تا ۱ تغییر می کند. با فرض این که x و z دو بردار 784 بعدی باشند که دو تصویر را نشان می دهند، می توان تابع هسته چندجمله ای زیر را بر روی آن ها تعریف کرد

$$k_1(x, z) = \langle x, z \rangle^9$$

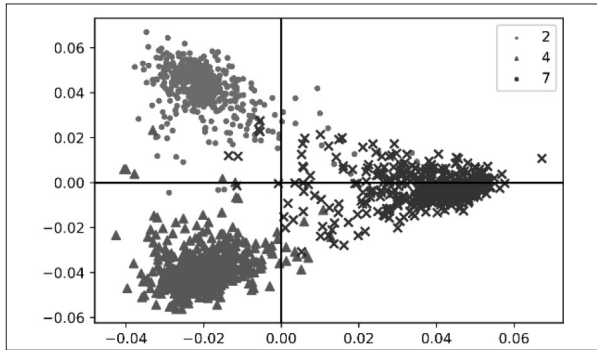
که فرض می کنیم $\langle x, z \rangle$ به صورت زیر تعریف شده است تا مقدار آن همواره بین ۰ و ۱ باشد

$$\langle x, z \rangle = \left(\frac{x}{\sqrt{784}} \right)^T \left(\frac{z}{\sqrt{784}} \right) = \frac{1}{784} \sum_{i=1}^{784} x_i z_i$$

هر چند این کار منطقی و کافی به نظر می رسد، اما نتایج عالی بر روی دادگان هسته توسط ماشین های بردار پشتیبان از تابع هسته دیگری استفاده کردند. آن ها ابتدا دامنه شدت رنگ را به بازه $[-1, 1]$ تغییر دادند. با توجه به این که بیشتر پیکسل ها سیاه یا سفید هستند، این تبدیل باعث می شود که بیشتر پیکسل ها مقدارشان برابر ۱ یا -۱ شود. به این ترتیب ضرب داخلی یک بردار در خودش همواره مقداری نزدیک به $784 \times 28 = 28 \times 28$ پیدا می کند که با تقسیم بر ۷۸۴ شباهت هر بردار با خودش تقریباً برابر یک می شود. برای آنکه تابع هسته ضرب داخلی همانند تابع هسته گوسی همواره مقداری نامنفی داشته باشد، آن ها تابع هسته زیر را معرفی کردند

$$k_2(x, z) = \frac{((2x - 1, 2z - 1) + 1)^9}{512} = \left(\frac{(2x - 1, 2z - 1) + 1}{2} \right)^9$$

که در آن ابتدا شدت رنگ پیکسل ها به بازه $[-1, 1]$ تغییر کرده است و در نتیجه $(2x - 1, 2z - 1)$ همواره عددی بین -۱ و ۱ است. در این بخش ما با استفاده از ابزار t-SNE

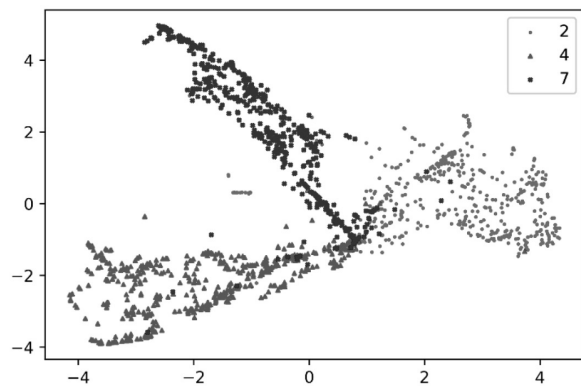


شکل ۴: ویژگی‌های استخراج شده برای دادگان آزمون توسط تحلیل فیشر برای تابع هسته ضرب داخلی مناسب k_2 بر روی سه رقم ۲، ۴ و ۷ از دادگان MNIST. k_2 تابع هسته ضرب داخلی است که بر روی پیکسل‌های با شدت $[-1, 1]$ اعمال شده است و همچنین طوری تنظیم شده است که عملکرد تابع هسته مشابه RBF باشد و شباهت هر تصویر با خودش تقریباً یک و کمترین مقدار شباهت صفر باشد.

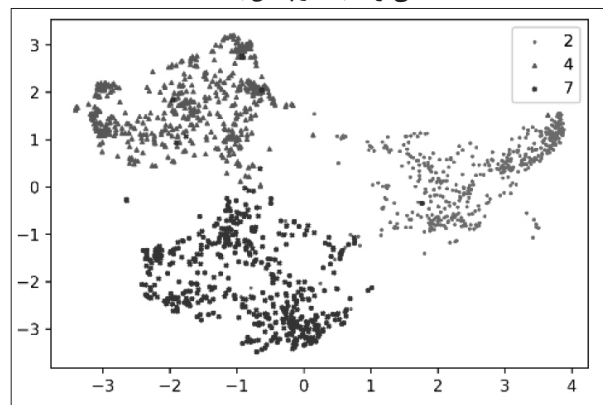
فضای ویژگی این دو تابع هسته را بصری‌سازی می‌کنیم. برای این منظور ما با استفاده از نگاشت صریح معرفی شده در این مقاله دادگان آموزشی را صراحتاً در فضای ویژگی نشان می‌دهیم و سپس از ابزار t-SNE (بدون کوچک‌ترین تغییری در آن) استفاده می‌کنیم و فضای ویژگی را بصری‌سازی می‌کنیم. شکل ۱ بصری‌سازی دادگان آزمون در فضای ویژگی تابع هسته k_1 و شکل ۲ بصری‌سازی دادگان آزمون در فضای ویژگی تابع هسته k_2 را نشان می‌دهد. همان‌طور که دیده می‌شود انتخاب مناسب تابع هسته به شکلی که در [۲ صفحه ۳۴۱] معرفی شده است نقش بسزایی در موفقیت روش‌های هسته بر روی دادگان MNIST داشته است. همچنین شکل‌های ۳ و ۴ نتیجه تحلیل تمایزی فیشر هسته‌ای را با هسته‌های k_1 و k_2 بر روی دادگان آزمون نشان می‌دهد. باز هم مشاهده می‌شود که هسته k_2 به مراتب بهتر از هسته k_1 عمل کرده است. در پیاده‌سازی ما به جای استفاده از تحلیل فیشر هسته‌ای [۱۳]، تحلیل فیشر چندکلاسه [۱۴] را بر روی فضای ویژگی صریح توابع هسته اعمال کردیم.

۵- نتیجه‌گیری

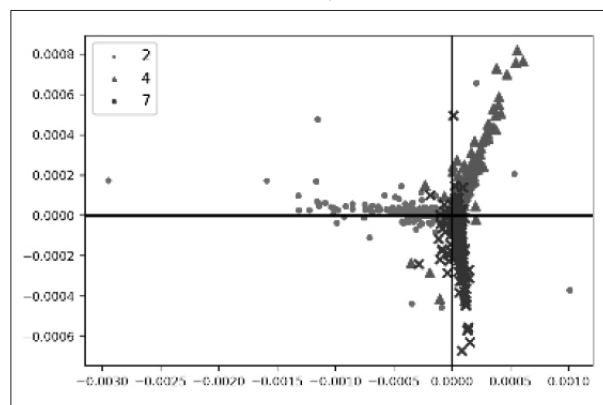
در این مقاله یک نگاشت ویژگی صریح متناهی البعد دقیق برای هر تابع هسته دلخواه معرفی شد که تا زمانی که حداقل یکی از دو ورودی تابع هسته یکی از دادگان



شکل ۱: بصری‌سازی فضای ویژگی تابع هسته ضرب داخلی نامناسب k_1 برای ارقام ۲، ۴ و ۷ از دادگان MNIST. k_1 تابع هسته ضرب داخلی با درجه ۹ است که بر روی پیکسل‌های با شدت $[0, 1]$ اعمال شده است. به‌خاطر وجود پیکسل‌های با مقدار صفر، شباهت یک داده با خودش هم می‌تواند بسیار پایین باشد.



شکل ۲: بصری‌سازی فضای ویژگی تابع هسته ضرب داخلی مناسب k_2 برای ارقام ۲، ۴ و ۷ از دادگان MNIST. k_2 تابع هسته ضرب داخلی است که بر روی پیکسل‌های با شدت $[-1, 1]$ اعمال شده است و همچنین طوری تنظیم شده است که عملکرد تابع هسته مشابه RBF باشد و شباهت هر تصویر با خودش تقریباً یک و کمترین مقدار شباهت صفر باشد.



شکل ۳: ویژگی‌های استخراج شده برای دادگان آزمون توسط تحلیل فیشر برای تابع هسته ضرب داخلی نامناسب k_1 بر روی سه رقم ۲، ۴ و ۷ از دادگان MNIST. k_1 تابع هسته ضرب داخلی با درجه ۹ است که بر روی پیکسل‌های با شدت $[0, 1]$ اعمال شده است. به‌خاطر وجود پیکسل‌های با مقدار صفر، شباهت یک داده با خودش هم می‌تواند بسیار پایین باشد.

$$\left(\frac{1}{N} \sum_{n=1}^N \psi(x_n) \psi(x_n)^T\right) v = \lambda v$$

با بازنویسی رابطه فوق رابطه زیر به دست می آید

$$\left(\frac{1}{N} \sum_{n=1}^N \psi(x_n) \psi(x_n)^T\right) \sum_{m=1}^N \alpha_m \psi(x_m) = \frac{1}{N} \sum_{n=1}^N \psi(x_n) \sum_{m=1}^N \alpha_m \psi(x_m)^T \psi(x_m)$$

با وجود این که تابع هسته ظاهر شده است، اما هنوز

بخشی از رابطه بر حسب نگاشت ψ است که برای خلاص شدن از آن دوطرف رابطه از سمت چپ در $\psi(x_\ell)^T$ ضرب می شود.

$$\psi(x_\ell)^T \sum_{n=1}^N \psi(x_n) (\bar{K} \alpha)_n = N \lambda \psi(x_\ell)^T \sum_{n=1}^N \alpha_n \psi(x_n)$$

که از آن نتیجه می شود

$$\sum_{n=1}^N \bar{k}(x_\ell, x_n) (\bar{K} \alpha)_n = N \lambda \sum_{n=1}^N \alpha_n \bar{k}(x_n, x_\ell)$$

رابطه فوق را به زبان ماتریسی می توان چنین بیان کرد:

$$(\bar{K} \bar{K} \alpha)_\ell = N \lambda (\bar{K} \alpha)_\ell$$

از آنجا که رابطه فوق برای هر ℓ برقرار است، رابطه

زیر به دست می آید.

$$\bar{K} \bar{K} \alpha = N \lambda \bar{K} \alpha$$

سپس ثابت می شود که کافی است پاسخ مسئله زیر

به دست آید:

$$\bar{K} \alpha = N \lambda \alpha$$

که

$$\bar{K} = K - e^T K - K e + e^T K e$$

همچنین در هنگام استفاده از الگوریتم $KPCA$ ، برای آنکه

تصویر داده ورودی x بر روی یک مؤلفه اصلی در فضای

ویژگی محاسبه شود، ابتدا داده x با نگاشت ψ به فضای

ویژگی با میانگین صفر می رود و سپس در مؤلفه اصلی

ضرب می شود

$$\left\langle \psi(x), \sum_{n=1}^N \alpha_n \psi(x_n) \right\rangle = \sum_{n=1}^N \alpha_n \langle \psi(x), \psi(x_n) \rangle = \sum_{n=1}^N \alpha_n \left(\phi(x) - \frac{1}{N} \sum_{m=1}^N \phi(x_m), \phi(x_n) - \frac{1}{N} \sum_{m=1}^N \phi(x_m) \right)$$

آموزشی است، مقدار آن دقیقاً با مقدار تابع هسته برابر است. از آنجا که هم در مرحله آموزش و هم در مرحله آزمون همواره یکی از دو ورودی تابع هسته یکی از دادگان آموزشی است، تا آنجا که به یادگیری ماشین مربوط می شود، نگاشت ویژگی صریح به دست آمده دقیق نیز هست. این مطلب نتایج مهمی دارد از جمله:

۱- امکان حل مسائل یادگیری ماشین در فضای ویژگی

توابع هسته در فرم اولیه بدون نیاز به فرم دوگان،

۲- امکان حل مسائل یادگیری ماشین در فضای ویژگی

در فرم دوگان بدون نیاز به حقه هسته،

۳- امکان نگاشت صریح داده ها به فضای ویژگی و

اجرای یک الگوریتم یادگیری ماشین بر روی این دادگان. به

این ترتیب امکان ارائه نسخه هسته ای الگوریتم با استفاده از

ابزارهای آماده صرفاً با تغییر دادگان ورودی به الگوریتم

فراهم می شود.

لازم به ذکر است که روش پیشنهادی در این مقاله به

این دلیل که ابعاد نگاشت ویژگی معرفی شده برابر با تعداد

دادگان است، از دیدگاه محاسباتی منجر به پارامتریک شدن

الگوریتم های هسته نمی شود و دنیای علم را از تقریب توابع

هسته با هدف افزایش سرعت محاسبات بی نیاز نمی کند.

ضمیمه الف: روش کلاسیک به دست آوردن نسخه

هسته ای الگوریتم PCA با حقه هسته

روش کلاسیک برای به دست آوردن نسخه هسته ای

الگوریتم PCA این است که ابتدا ثابت می شود که بردارهای

ویژه به صورت ترکیبی خطی از دادگان آموزشی قابل بیان

هستند. سپس با فرض این که ψ نگاشتی به یک فضای

ویژگی باشد که میانگین داده های آموزشی در آن صفر

است، بردار ویژه به صورت $v = \sum_{n=1}^N \alpha_n \psi(x_n)$ در

فضای ویژگی بیان می شود. همچنین فرض کنید که تابع

هسته متناظر با این نگاشت برابر \bar{k} باشد. سپس با توجه

به این که در نسخه هسته ای PCA، پاسخ بردارهای ویژه

ماتریس کواریانس داده ها در فضای ویژگی است، مسئله

به صورت زیر بیان می شود

- [6].Vedaldi, Andrea, and Andrew Zisserman. "Efficient additive kernels via explicit feature maps." IEEE transactions on pattern analysis and machine intelligence, Vol. 34, No. 3, pp. 480-492, 2012.
- [7].Zafeiriou, Stefanos, and Irene Kotsia. "On One-Shot Similarity Kernels: explicit feature maps and properties." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2392-2399, 2013
- [8].Hamid, Raffay, Ying Xiao, Alex Gittens, and Dennis De-Coste. "Compact random feature maps." In International conference on machine learning, pp. 19-27. PMLR, 2014.
- [9].Francis, Deena P., and Kumudha Raimond. "A fast and accurate explicit kernel map." Applied Intelligence, Vol. 50, No. 3, pp. 647-662, 2020.
- [10].Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. "Nonlinear component analysis as a kernel eigenvalue problem." Neural computation, Vol. 10, No. 5, pp. 1299-1319, 1998.
- [11].Williams, Christopher K., and Carl Edward Rasmussen. Gaussian processes for machine learning. Vol. 2, no. 3. Cambridge, MA: MIT press, 2006.
- [12].Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research, Vol. 9, No. 11, 2008.
- [13].Baudat, Gaston, and Fatiha Anouar. "Generalized discriminant analysis using a kernel approach." Neural computation, Vol. 12, No. 10, pp. 2385-2404, 2000.
- Fukunaga, Keinosuke. "Introduction to statistical pattern recognition." 1990.

$$\begin{aligned}
 &= \sum_{n=1}^N \alpha_n \left(k(x, x_n) - \frac{1}{N} \sum_{m=1}^N k(x, x_m) \right) \\
 &- \sum_{n=1}^N \alpha_n \left(\frac{1}{N} \sum_{m=1}^N k(x_m, x_n) + \frac{1}{N^2} \sum_{m=1}^N \sum_{p=1}^N k(x_m, x_p) \right) \\
 &= \sum_{n=1}^N \left(\alpha_n - \frac{1}{N} \sum_{m=1}^N \alpha_m \right) k(x, x_n) - e^T K \alpha + e^T K e \alpha^T e
 \end{aligned}$$

مراجع

- [1]. Aronszajn, Nachman. "Theory of reproducing kernels." Transactions of the American mathematical society, Vol. 68, No. 3, pp. 337-404, 1950.
- [2]. Schölkopf, Bernhard, Alexander J. Smola, and Francis Bach. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.
- [3].Francis, Deena P., and Kumudha Raimond. "Major advancements in kernel function approximation." Artificial Intelligence Review, Vol. 54, No. 2, pp. 843-876, 2021.
- [4].Williams, Christopher, and Matthias Seeger. "Using the Nyström method to speed up kernel machines." In Proceedings of the 14th annual conference on neural information processing systems, pp. 682-688. 2001.
- [5].Rahimi, Ali, and Benjamin Recht. "Random Features for Large-Scale Kernel Machines." In NIPS, Vol. 3, No. 4, p. 5, 2007.



کار عمیق
نوشته کل نیوپورت
ترجمه اراهم نعیم

جدیدترین کتاب
از انتشارات انجمن انفورماتیک ایران
منتشر شد!

کار عمیق

برای تهیه کتاب با دفتر انجمن انفورماتیک ایران
تماس بگیرید ۶۶۴۱۲۸۶۱

چاپ پنجم