تکلیف کامپیوتری چهارم - درس شناسایی الگو

وحید ملکی
شماره دانشجویی: ۴۰۳۱۳۰۰۴

۲۹ آذر ۱۴۰۴

چکیده

در این گزارش، پیاده‌سازی دسته‌بند ماشین بردار پشتیبان (SVM) بر روی مجموعه داده Fashion-MNIST بررسی می‌شود. این تمرین شامل دو بخش اصلی است: پیاده‌سازی الگوریتم از پایه با استفاده از کتابخانه NumPy و حل دوگان مساله بهینه‌سازی، و سپس مقایسه نتایج با پیاده‌سازی استاندارد کتابخانه Scikit-learn. تأثیر پارامترهای مختلف مانند C و نوع کرنل بر دقت مدل ارزیابی شده و تکنیک‌های کاهش داده‌های مبهم برای بهبود سرعت پیش‌بینی تحلیل شده‌اند. نتایج نشان می‌دهند که اگرچه پیاده‌سازی دستی با کرنل RBF عملکرد قابل قبولی دارد، اما در کرنل خطی به دلیل ناپایداری‌های عددی و عدم همگرایی گرادیان، عملکرد ضعیفی از خود نشان می‌دهد.

۱ مقدمه و آماده‌سازی داده‌ها

مجموعه داده Fashion-MNIST شامل تصاویر سیاه‌وسفید 28×28 از ۱۰ دسته مختلف پوشاک است. در این تمرین، طبق دستورالعمل، زیرمجموعه‌ای از داده‌ها بارگذاری شد.

۱۰۱ بارگذاری و پیش‌پردازش

برای افزایش سرعت اجرا، از 2000 نمونه داده استفاده شد. داده‌ها با استفاده از تابع fetch_openml بارگذاری و سپس به صورت تصادفی مخلوط (Shuffle) شدند. همچنین، داده‌ها نرمال‌سازی شدند (میانگین صفر و واریانس یک) تا همگرایی الگوریتم‌های مبتنی بر گرادیان بهبود یابد. تقسیم داده‌ها به صورت 80% آموزش و 20% تست انجام شد.

۲ پیاده‌سازی SVM از پایه (Scratch)

۱۰۲ تئوری و فرمول‌بندی ریاضی

برای پیاده‌سازی SVM، از فرم دوگان (Dual Form) استفاده شده است. هدف در SVM پیدا کردن ابرصفحه‌ای است که حاشیه (Margin) را بیشینه کند. مساله اولیه (Primal) به صورت زیر است:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \qquad (۱)$$

با استفاده از ضرایب لاگرانژ (α_i)، مساله دوگان که ما آن را حل می‌کنیم به صورت زیر است:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j K(x_i, x_j) \qquad (۲)$$

با قیود:

$$0 \le \alpha_i \le C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

که در آن $K(x_i, x_j)$ تابع کرنل است. در کد پیاده‌سازی شده، برای یافتن αهای بهینه از روش Gradient Ascent استفاده شده است. گرادیان تابع هدف نسبت به α برابر است با:

$$\nabla L = 1 - y \cdot (K \cdot (\alpha \odot y)) \qquad (۳)$$

که در کد پایتون به صورت gradient = 1 - y * np.dot(K, self.alpha * y) پیاده‌سازی شده است. پس از هر به‌روزرسانی، مقادیر α با استفاده از پارامتر C برش (Clip) داده می‌شوند تا قید $0 \le \alpha_i \le C$ رعایت شود.

۲.۲ توابع کرنل

دو نوع کرنل پیاده‌سازی شد:

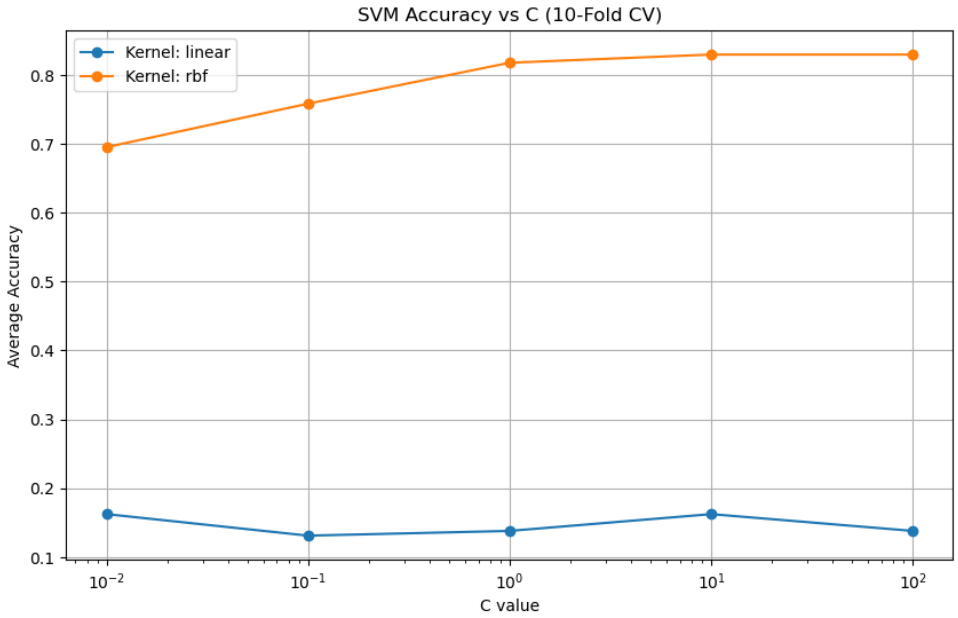
- **کرنل خطی (Linear):** $K(x_i, x_j) = x_i^T x_j$
- **کرنل RBF:** $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$

۳ ارزیابی و بهینه‌سازی پارامترها (پیاده‌سازی دستی)

در این بخش تاثیر مقادیر مختلف C و نوع کرنل با استفاده از 10-Fold Cross Validation بررسی شد.

۱.۳ تحلیل نتایج پیاده‌سازی دستی

نمودار زیر دقت مدل را بر حسب مقادیر مختلف C نشان می‌دهد:



شکل ۱: دقت متوسط در اعتبارسنجی متقاطع برای پیاده‌سازی دستی SVM

همانطور که در نمودار ۱ و نتایج خروجی مشاهده می‌شود:

- **عملکرد کرنل RBF:** دقت با افزایش C افزایش یافته و در حدود $C = 1$ تا $C = 10$ به ثبات می‌رسد (حدود 82%).
- **شکست کرنل خطی:** دقت کرنل خطی بسیار پایین (حدود 12% تا 14%) است که نزدیک به حدس تصادفی برای ۱۰ کلاس است.

۱.۱.۳ چرا کرنل خطی در کد دستی دقت بسیار پایینی دارد؟

این یک پدیده رایج در پیاده‌سازی‌های ساده SVM است. دلایل اصلی عبارتند از:

۱. **نرخ یادگیری (Learning Rate):** مقادیر حاصل از ضرب داخلی در کرنل خطی ($x^T x$) می‌تواند بسیار بزرگ باشد (برخلاف RBF که بین ۰ و ۱ است). استفاده از نرخ یادگیری ثابت (0.001) برای هر دو کرنل باعث می‌شود که در حالت خطی، گرادیان‌ها منفجر شده (Exploding Gradients) یا نوسان کنند و الگوریتم هرگز همگرا نشود.

۲. **عدم ارضای دقیق قیود:** روش گرادیان ساده پیاده‌سازی شده، قید تساوی $\sum \alpha_i y_i = 0$ را به طور صریح تضمین نمی‌کند (روش‌هایی مثل SMO این کار را انجام می‌دهند). این قید برای بایاس مدل حیاتی است. در فضای ویژگی بسیار بزرگ تصاویر، بدون این قید، ابرصفحه جداکننده به درستی تشکیل نمی‌شود.

۲.۱.۳ چرا تغییرات C تأثیر محسوسی در دقت نهایی ندارد؟

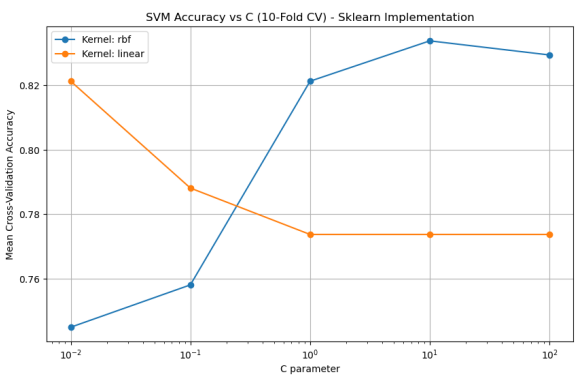
در نمودار مشاهده می‌شود که پس از یک مقدار اولیه، تغییر C (مثلاً از ۱۰ به ۱۰۰) تغییر زیادی در دقت ایجاد نمی‌کند. دلیل آن پدیده **اشباع (Saturation)** است. پارامتر C جریمه خطای دسته‌بندی است. وقتی C به اندازه کافی بزرگ باشد، مدل سعی می‌کند تمام داده‌های آموزشی را درست دسته‌بندی کند. در پیاده‌سازی دستی، به دلیل مکانیزم Clipping ($\alpha = \min(\alpha, C)$)، وقتی آلفاها به مقدار ماکزیمم می‌رسند، افزایش بیشتر C صرفاً حد بالای برش را زیاد می‌کند اما اگر ساختار هندسی داده‌ها (که توسط گاما در RBF تعیین می‌شود) اجازه ندهد، مرز تصمیم تغییر شگرفی نمی‌کند. به عبارت دیگر، مدل به "ظرفیت نهایی" خود با آن کرنل و داده خاص رسیده است.

۴ مقایسه با پیاده‌سازی Scikit-Learn

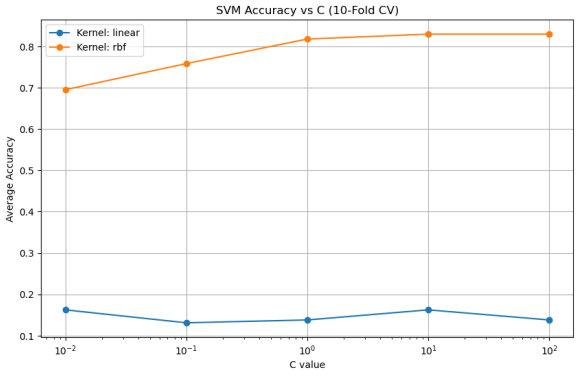
در این بخش نتایج پیاده‌سازی دستی با کتابخانه استاندارد Sklearn مقایسه می‌شود.

۱.۴ مقایسه نمودارهای دقت

در تصویر زیر، نمودارهای دقت بر حسب C برای هر دو پیاده‌سازی در کنار هم قرار گرفته‌اند:



(ب) نتایج کتابخانه Scikit-Learn



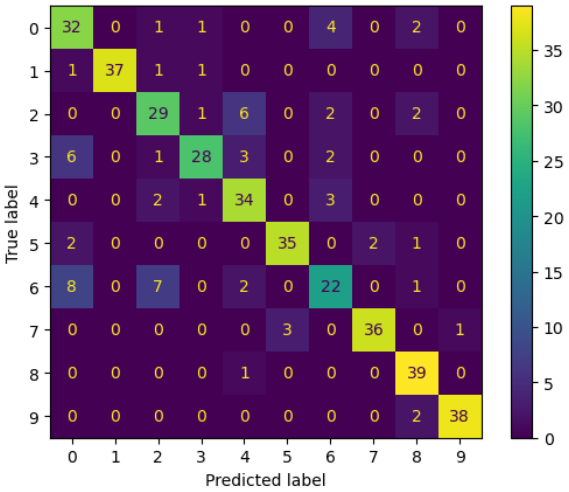
(آ) نتایج پیاده‌سازی دستی (NumPy)

شکل ۲: مقایسه روند دقت اعتبارسنجی متقاطع با تغییر پارامتر C

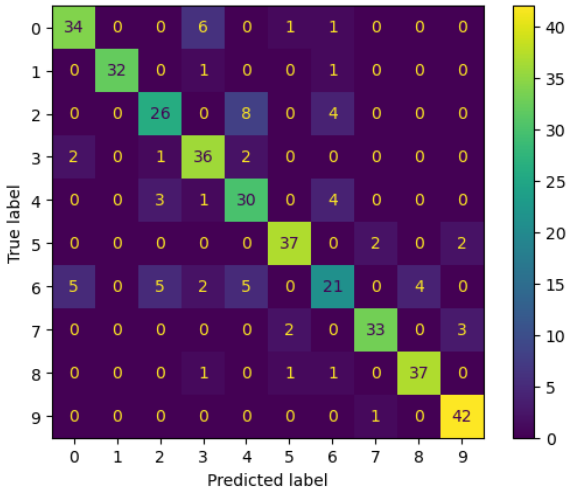
در پیاده‌سازی Sklearn (سمت چپ)، مشاهده می‌شود که کرنل خطی نیز عملکردی قابل قبول (شروع از حدود 82%) دارد، که تفاوت اصلی آن با پیاده‌سازی دستی به دلیل استفاده از حل گرهای بهینه LibSVM و مدیریت صحیح نرخ یادگیری و قیود است. همچنین در Sklearn با افزایش بیش از حد C در کرنل خطی، شاهد افت دقت هستیم (Overfitting)، اما کرنل RBF رفتار صعودی و پایدار خود را حفظ کرده است.

۲.۴ مقایسه ماتریس درهم‌ریختگی (Confusion Matrix)

بهترین مدل یافت شده در هر دو حالت (برای دستی $C = 10, RBF$ و برای Sklearn نیز $C = 10, RBF$) برای پیش‌بینی روی داده‌های تست استفاده شدند.



(ب) ماتریس درهم‌ریختگی - Sklearn



(آ) ماتریس درهم‌ریختگی - دستی

شکل ۳: مقایسه عملکرد دسته‌بندی روی داده‌های تست نهایی

هر دو مدل الگوی خطای مشابهی دارند. به طور خاص کلاس شماره ۶ (Shirt) بیشترین خطا را دارد و اغلب با کلاس‌های دیگر (مانند T-shirt/Top کلاس ۰ یا Coat کلاس ۴) اشتباه گرفته می‌شود که با توجه به شباهت بصری این لباس‌ها منطقی است. دقت نهایی تست برای مدل دستی حدود 83.38%

و برای مدل Sklearn حدود 82.50% به دست آمد که نشان‌دهنده موفقیت‌آمیز بودن پیاده‌سازی دستی در حالت کرنل RBF است.

۵ بهینه‌سازی: حذف داده‌های مبهم و تحلیل زمان

۱۰۵ حذف نمونه‌های مبهم

طبق دستورالعمل، نمونه‌هایی از داده‌های آموزش که مدل (با پارامترهای بهینه) نتوانست آن‌ها را به درستی دسته‌بندی کند، به عنوان "داده‌های مبهم" یا Noisy در نظر گرفته شده و حذف شدند.

- تعداد کل داده‌های آموزش: 1600
- تعداد داده‌های حذف شده: 7 نمونه (حدود 0.44%)
- تعداد داده‌های باقی‌مانده: 1593

پس از آموزش مجدد مدل روی داده‌های پاک‌سازی شده، دقت روی داده‌های تست برابر با 82.00% شد. کاهش جزئی دقت نشان می‌دهد که نمونه‌های حذف شده احتمالاً حاوی اطلاعات مرزی مهمی بوده‌اند (بردار پشتیبان) و حذف آن‌ها باعث تغییر جزیی مرز تصمیم شده است.

۲۰۵ تحلیل آماری زمان اجرا (t-test)

زمان پیش‌بینی روی مجموعه تست برای ۵۰ بار اجرا در دو حالت (مدل اولیه و مدل کاهش‌یافته) اندازه‌گیری شد:

- میانگین زمان مدل اولیه: 0.5614 ثانیه
- میانگین زمان مدل کاهش‌یافته: 0.5532 ثانیه

برای بررسی معناداری این اختلاف، آزمون Paired t-test انجام شد. مقدار p-value برابر با 1.26×10^{-7} به دست آمد. از آنجا که $p\text{-value} < 0.05$ است، فرض صفر رد می‌شود. این بدین معناست که **تفاوت زمان اجرا از نظر آماری معنادار است.** هرچند مقدار مطلق کاهش زمان کم است (حدود ۱ صدم ثانیه)، اما حذف بردارهای پشتیبان (حتی تعداد کم) باعث کاهش محاسبات کرنل در زمان تست می‌شود.

۶ نتیجه‌گیری

در این تمرین، دسته‌بند SVM با موفقیت پیاده‌سازی شد. نتایج نشان داد که پیاده‌سازی دستی با کرنل RBF توانایی رقابت با کتابخانه‌های استاندارد را دارد، اما در کرنل‌های خطی بدون مکانیزم‌های کنترلی پیشرفته (مانند نرمال‌سازی گرادیان یا SMO) دچار مشکل می‌شود. همچنین مشخص شد که حذف داده‌های مبهم (که در واقع بردارهای پشتیبان سخت هستند) می‌تواند سرعت را اندکی بهبود بخشد اما ممکن است به قیمت کاهش جزئی دقت تمام شود.