به نام خدا

دانشکده‌ی علوم و مهندسی کامپیوتر — یادگیری ماشین

تکلیف سوم – زمستان ۱۳۹٤ — مهلت ارسال: ۲۵ بهمن ۱۳۹٤

قبل از اینکه شروع به حل تمرین‌ها کنید، حتما یک بار فایل Policies.pdf (موجود در وبگاه) که حاوی نکات مهم در تحویل تکالیف هست را مطالعه فرمایید.

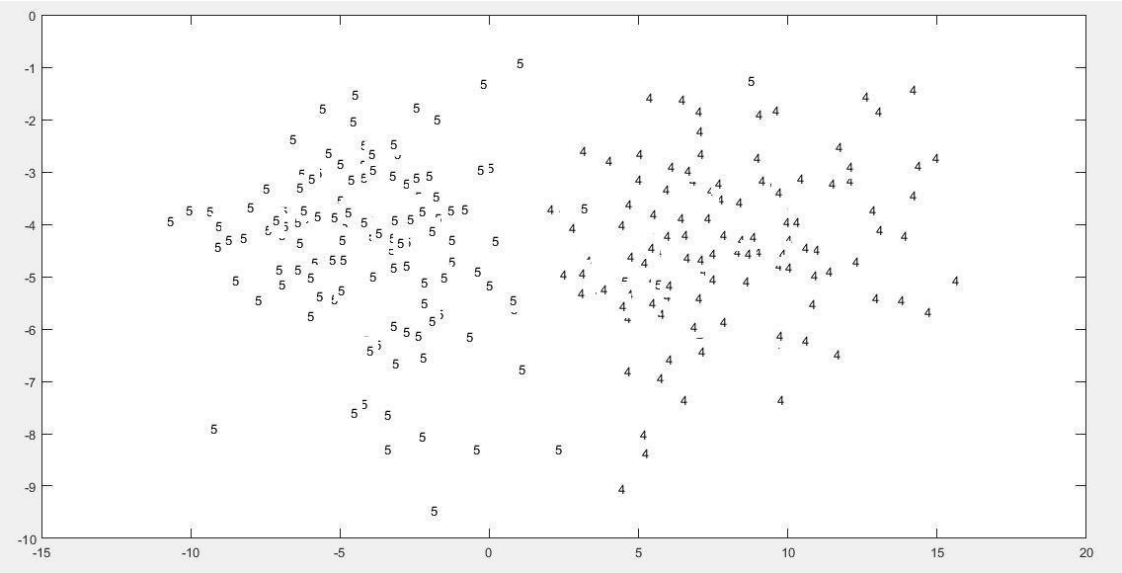
حل سوالاتی که امتیازی هستند، اجباری نبوده و نمره‌ی اضافی خواهد داشت.

طراح هریک از سوالات در زیر مشخص شده است .در صورتی که سوالی دارید می توانید سوالتان را در گروه مطرح کنید یا از طراح سوال بپرسید.

۱: آقای زندی

۲ و ۳: آقای کاهانی

۱- برای جداسازی تصاویر کاراکترهای ’5‘ و ’4‘ از یکدیگر، دو ویژگی استخراج شده است. این داده‌ها به صورت زیر در فضای ویژگی توزیع شده‌اند (Usps_4-5_2D.mat):



الف) یک جداساز خطی را با استفاده از متد گرادیان نزولی آموزش دهید که این کاراکترها را جدا نماید. روند گرادیان نزولی را خودتان پیاده‌سازی نمایید و ملاحظات زیر را در نظر داشته باشید:

- ۸۰ درصد داده‌ها را به طور تصادفی برای آموزش انتخاب کنید.
- در هر تکرار، کل داده‌های آموزش را اعمال کرده و در نهایت وزن‌ها را به‌روزرسانی نمایید.
- در هر تکرار، جداساز را رسم نمایید. برای رسم جداساز توابع plotDiagram2D و plotDiagram3D در پوشه‌ی 1 قرار داده شده‌اند و کافی است تا آنها را در کد خود فراخوانی نمایید.

ب) به ازای مقادیر زیر برای نرخ یادگیری، آموزش را انجام داده و در نهایت خطای تست و آموزش را به ازای هر یک گزارش دهید. (ترجیحا به صورت نمودار یا جدول)

$$\eta = 1, 0.1, 0.01$$

ج) به ازای بهترین میزان نرخ یادگیری، سرعت تغییر جهت بردار وزن‌ها در فاز آموزش را تحلیل نمایید. در چه مواقعی و چرا تغییرات بیشتر یا کمتر می‌شود؟

د) در هنگام آموزش، اندازه‌ی بردار وزن‌ها در شکل تابع جداساز (نه مرز جداساز!) چه تاثیری دارد؟ همچنین بیان نمایید که چرا باید جداساز اندازه‌ی بردار وزن‌ها را تغییر دهد؟

۲- در دسته بند SVM تابع هدف را به شکل زیر تغییر می دهیم، یعنی مقدار پارامتر C را متغیر و وابسته به نمونه در نظر می گیریم.

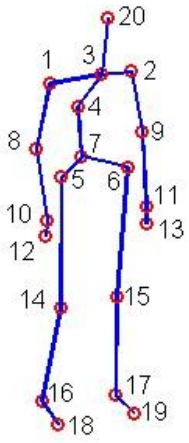
$$\tfrac{1}{2}W^T W + \sum C_i \xi_i$$
$$\text{Subject to: } d_i (W^T X_i + b) \geq 1 - \xi_i$$

الف) این کار چه مزایا و معایبی دارد؟

ب) در این صورت چگونه می توان مقدار مناسب پارامتر C برای هر داده را تعیین کرد؟

۳- داده های مربوط به Skeleton یک مجموعه داده Human Action را در اختیار داریم (مجموعه داده ی MSR Action3D). این مجموعه شامل بیست Action مختلف است که هر کدام توسط ۱۰ نفر با ۲ تا ۳ تکرار انجام شده و در مجموع ۵۵۷ دنباله (نمونه) دارد. این داده ها برای هر فریم اطلاعات مکان مربوط به ۲۰ نقطه (مفصل) از بدن را دارا می باشد (شکل ۱). مکان هر مفصل در هر فریم با ۳ مقدار مشخص شده است. یعنی (x, y, z) که z مقدار عمق را داراست.

آرایه ی سلولی data شامل ۵۵۷ سلول است که هر سلول یک ماتریس و هر ماتریس ویژگی های یک نمونه را داراست. به این ترتیب سطر i ام هر ماتریس ویژگی‌های فریم i ام را داراست (که ۶۰ بعد است-۲۰ نقطه که هر کدام سه ویژگی دارند). برچسب متناظر با هر نمونه نیز در بردار labels موجود است. (کل داده ها در فایل MSR.mat قرار دارد).



الف) با استفاده از مدل مخفی مارکوف عمل یادگیری این مجموعه را انجام دهید.

راهنمایی: ابتدا لازم است که هر دنباله ی تصویر به دنباله ای از کاراکتر ها تبدیل شود. یک راه حل این است که از خوشه بندی کردن فریم ها استفاده کنید. در این صورت هر فریم برچسب یک خوشه را گرفته و معادل یک کاراکتر می شود.

ب) دقت دسته بندی را روی داده های آزمایش برای مقادیر مختلف پارامتر ها بدست آورید.

ج) این بار پس از خوشه بندی یک هیستوگرام از تعداد رخداد های هر خوشه برای هر ویدئو بدست آورده و با استفاده از الگوریتم های Adaboost (به وسیله ی یادگیرنده پایه درخت تصمیم) و SVM عمل یادگیری را انجام دهید.

د) دقت های بدست آمده از روش های مختلف را مقایسه و علت آن را تحلیل کنید.

توجه - در آزمایش ها از۶۰ درصد داده ها برای آموزش استفاده کنید. (برای محاسبه ی دقت کل دسته بندی، تعداد نمونه های درست شناسایی شده به تعداد کل نمونه ها را محاسبه نمایید).

توجه – در صورت داشتن مشکل با زمان محاسبات می توانید تنها از ۱۰ کلاس اول داده ها استفاده کنید.

توجه – برای SVM چند کلاسه از روش One VS. Rest استفاده کنید.

راهنمایی: برخی از توابع متلب که احتمالا مورد استفاده قرار گیرند:

- برای مشخص کردن داده های آموزشی و آزمایشی می توانید از متد HoldOut در تابع crossvalind() استفاده کنید.

- برای Adaboost از تابع fitensemble() استفاده کنید.

- برای SVM، در صورتی که از متلب استفاده می کنید: در نسخه های ۲۰۱۴ و به بعد پیاده سازی حالت چند کلاسه آن در تابع fitcecoc() موجود است. در غیر این صورت برای SVM دو کلاسه توابع svmtrain() و svmclassify() موجود است.

- برای خوشه بندی از تابع kmeans() می توانید استفاده کنید (برای افزایش سرعت می توانید از پیاده سازی بهینه ی آن در کتابخانه‌ی VLFeat استفاده کنید، تابع vl_kmeans())

- برای مارکوف توابع hmmtrain() و hmmdecode() مورد استفاده قرار خواهند گرفت.

- توابع دیگری که ممکن است لازم شود، grp2idx() – confusionmat() – bsxfun() هستند.

موفق باشید