



دانشگاه یزد

گروه مهندسی کامپیوتر

رشته تحصیلی: هوش مصنوعی و رباتیکز

نام درس: تشخیص الگو

پروژه پایانی

(تشخیص اعداد دست نویس)

استاد مربوطه: دکتر مهدی یزدیان

تهیه کننده: حمیدرضا نادمی

پاییز ۱۳۹۷

## چکیده

در این پروژه به بررسی متدهای مختلف جهت تشخیص اعداد دست نویس انگلیسی پرداختیم، الگوریتم های بررسی شده عبارتند از بیزین،  $k$  نزدیکترین همسایه، LDA و استفاده از GMM با گوسی فرض کردن توزیع هر کلاس (که اعداد بین صفر تا نه می باشند) و تخمین پارامترهای هر گوسی با استفاده از الگوریتم EM پرداخته ایم. و بهترین نتیجه را با متد  $k$  نزدیکترین همسایه با در نظر گرفتن  $k$  برابر با یک با دقت ۹۱ درصد بدست آوردیم.

از دیتاست MNIST جهت ارزیابی متدهای پیاده سازی شده استفاده شده است.

## ۱. مقدمه

با توجه به اهمیت تشخیص اعداد دست نویس در حوزه های مختلف مانند شناسایی خودکار کد پستی از روی تصویر، شناسایی ارقام پلاک خودرو در راهنمایی و رانندگی به بررسی الگوریتم های مختلف جهت تشخیص اعداد دست نویس پرداخته ایم.

در یادگیری ماشین و تشخیص الگو با مشاهده و پردازش حجم زیادی از داده (که به آنها داده آموزش می گویند) مدلی ساخته می شود و بر مبنای آن مدل میزان شباهت عدد داده شده به مدل به عنوان ورودی (داده تست) با اعداد صفر تا نه انگلیسی سنجیده می شود و عددی که بیشترین شباهت را به ورودی مدل داشته باشد به عنوان برچسب برای آن عدد انتخاب می شود.

در این گزارش ۲۰۰۰۰ تصویر به عنوان داده آموزش (۲۰۰۰ تصویر برای هر عدد صفر تا نه) و ۱۵۰۰ تصویر به عنوان داده تست (۱۵۰ تصویر برای هر عدد بین صفر تا نه) از دیتاست مشهور MNIST استفاده شده است.

## ۲. پیش پردازش

جهت افزایش دقت در تشخیص اعداد دست نویس، ساده کردن مدل، کاهش زمان اجرا و جلوگیری از بیش برازش شدن مدل قبل از بکارگیری داده ها یک سری پردازش روی داده ها انجام دادیم که به شرح زیر هستند:

### ۱.۲. فیلتر میانگین

فیلتر میانگین یک فیلتر غیرخطی است که جهت حذف نویز از تصویر یا یک سیگنال استفاده می شود. در شکل (۱) و شکل (۲) تصاویر بدست آمده قبل و بعد از اعمال فیلتر میانگین را برای اعداد مختلف نشان داده شده است.



شکل ۱، قبل از اعمال فیلتر میانگین



شکل ۲، بعد از اعمال فیلتر میانگین

### ۲.۲. نرمال کردن داده ها:

جهت افزایش دقت و کارایی مدل تصویر همه اعداد به ابعاد  $28 \times 28$  تبدیل شدند و سپس هر تصویر به یک وکتور  $784 \times 1$  تبدیل شد.

### ۲.۲. استخراج ویژگی با PCA

بعد از وکتورایز شدن تصاویر (داده های آموزش و تست) جهت کم کردن ویژگی های هر عکس روی همه

**Input:** Data  $X_{D \times N}$  (each column a D-dim sample)

1.  $\mu$  = sample mean of  $X$ ,  $\Sigma$  = sample covariance of  $X$
2.  $\bar{X}$  = subtract sample mean  $\mu$  from each column sample ( $\bar{X}$  has zero mean)
3. Find eigenvectors and eigenvalues of  $\Sigma$
4.  $W$  = Using  $d$  ( $d < D$ ) **eigenvectors with largest eigenvalues** form the mapping function as a  $D \times d$  matrix, each column corresponds to an eigenvector  
 - value of eigenvalues gives importance of each component
5. The transformed samples will be  $Y = W^T \bar{X}$

**Output:** Transformed data  $Y_{d \times N}$  (each column a d-dim sample)

شکل ۳، الگوریتم PCA

شده و کلاسی که بیشترین احتمال را داشته باشد

(رابطه ۲) به عنوان برچسب برای تصویر انتخاب

می‌شود. در اینجا توزیع هر کلاس گاوسی فرض شده

است (رابطه ۳).

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{\sum_{i=1}^{N_C} P(x|C_i)P(C_i)} = \frac{P(x|C_k)P(C_k)}{p(x)}$$

رابطه (۱)، تئوری بیز

$$C^* = \underset{C_k}{\operatorname{argmax}} P(C_k|x) \quad \text{رابطه (۲)}$$

$$P(x) = \mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

رابطه (۳)، تئوری بیز

### ۲.۳. k نزدیکترین همسایه

در دسته بندی با استفاده از k نزدیکترین همسایه،

یک پنجره به مرکز هر داده تشکیل می‌شود و اینقدر

پنجره بزرگ می‌شود تا k داده در آن قرار بگیرند،

سپس کلاسی که بیشترین تعداد از داده‌های موجود در

پنجره در آن قرار بگیرند به عنوان کلاس سمپل در نظر

گرفته می‌شود.

### ۳.۳. LDA

داده‌ها PCA اعمال کردیم و در شکل (۳)

الگوریتم PCA که پیاده سازی شده است را مشاهده

می‌کنید.

با استفاده از رابطه نسبت واریانس‌ها تعداد ویژگی جهت

کاهش ویژگی را بدست آوردیم

$$POV = \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^D \lambda_i}$$

در این رابطه مقدار POV برابر با ۰,۹۵ فرض شده

است و مقدار **d=289** بدست آمد.

بنابراین یک ماتریس به ابعاد ۲۸۹\*۲۰۰۰۰ که

سطر آن تعداد تصاویر و ستون آن ویژگی‌های هر

تصویر است جهت آموزش و یک ماتریس ۲۸۹\*۱۵۰۰

جهت ارزیابی مدل از مرحله پیش پردازش داده‌ها

بدست آمد.

### ۳. روش‌های بررسی شده

#### ۱.۳. تئوری بیزین

با استفاده از رابطه (۱) زیر که به تئوری بیزین

مشهور است احتمال تعلق داده به هر کلاس محاسبه

**Input:** Data  $X_{D \times N}$  (each column a D-dim sample)

1.  $X_1, X_2$ : a  $D \times N_1$  and  $D \times N_2$  matrix including samples of class  $C_1$  and  $C_2$
2.  $\mu_1, \mu_2$  = sample mean of  $X_1$  and  $X_2$
3.  $S_1, S_2$  = scatter matrix of  $X_1$  and  $X_2$
4.  $S_W = S_1 + S_2$  (within class scatter matrix)
5.  $w = S_W^{-1}(\mu_1 - \mu_2)$
6. The transformed samples will be  $Y = w^T X$

**Output:** Transformed data  $Y_{1 \times N}$  (each column a 1-dim sample)

شکل ۵، الگوریتم LDA برای یک مسئله دو کلاس

- Input: Given examples  $\{x_i\}_{i=1}^N$  as well as the label for each example  $o_i \in \{0, 1, \dots, C\}$ , and the example  $\hat{x}$ , without label;
- Output: the label  $o$  of the example  $x^*$ ;
- Algorithm:
  - Split the training data into  $c$  subsets according to the labels;
  - For each subset of a certain class  $c$ ,
  - Do:
    - Initialize:  $\theta_c = \{w_i, \mu_i, \Sigma_i, i = 1, 2, \dots, k\}$ ;
    - Repeat until convergence:
      - Expectation: compute the expectations  $E(z_i^j | x_i, \theta_c)$  using (12);
      - Maximization: update the parameters  $\theta_c \leftarrow \text{argmax}(L(\theta, \theta_c))$  using (14);
    - Store the parameters  $\theta_c$ ;
  - Compute the expectation of  $E(z_i^j)$  for  $x^*$ , and then calculate the likelihood using (13) for each class;
  - Find out  $\hat{c} = \text{argmax}(L(x, \theta_c))$ ;
  - Output  $\hat{c}$  as the label of  $x^*$ ;
- End.

شکل ۶، الگوریتم GMM

LDA یکی دیگر از روش های استخراج ویژگی می باشد با این تفاوت که کاهش بعد طوری انجام می شود که تمایز بین کلاس ها حداکثر شود، از این رو از این الگوریتم (شکل ۵) جهت حل یک مسئله دسته بندی نیز استفاده می شود. به منظور تعیین کلاس داده  $Z$  از رابطه زیر استفاده می شود [۱]:

$$\underset{k}{\operatorname{argmin}} d(zW, \bar{X}_k W)$$

که در این رابطه  $d$  تابعی است که فاصله اقلیدسی بین دو وکتور را محاسبه می کند،  $K$  تعداد کلاس های مسئله،  $W$  یک ماتریس به ابعاد:  $k - 1 * D$  می باشد که با توجه به داده های مورد آزمایش در این پروژه ماتریسی  $9 * 289$  می باشد و  $\bar{X}_k$  بردار میانگین هر کلاس می باشد.

### ۴.۳. دسته بندی با استفاده از GMM

با استفاده از الگوریتم  $em$  هر بار مقدار ماتریس کوواریانس، بردار میانگین و وزن هر گوسی تخمین زده می شود تا زمانی که به یک همگرایی برسیم. تعداد گوسی فرض شده برای هر کلاس یک می باشد.

الگوریتم پیاده سازی شده جهت کلاس بندی در شکل (۶) نمایش داده شده است [۲].

#### ۴. ارزیابی نتایج

در کلاس بندی به روش  $k$  نزدیکترین همسایه از آنجایی که این الگوریتم بصورت lazy عمل می کند، به این صورت که هیچ مدلی جهت پیش بینی برچسب یک داده درست نمی کند و برای هر بار برای کلاسبندی یک داده تمامی داده های آموزش را در حافظه خود بارگذاری می کند باعث شد تا بیشترین زمان اجرا را جهت انجام کلاس بندی روی داده های تست داشته باشد. در جدول ۱ زیر نتیجه بدست آمده مقدار های مختلف  $k$  را مشاهده می کنید.

	Mean Accuracy	Mean F1 score
$k = 1$	۰٫۹۱	۰٫۹۱

جدول ۱، میانگین دقت و  $F1$  Score بدست آمده با مقدارهای مختلف  $k$

در جدول ۲ میانگین دقت و  $F1$  score محاسبه شده از خروجی هریک از روش های ذکر شده قرار داده شده است.

	Mean Accuracy	Mean F1 score
Bayesian	۰٫۷۸	۰٫۷۸
1-NN	۰٫۹۱	۰٫۹۱
LDA	۰٫۶۸	۰٫۶۹
GMM	۰٫۵۷	۰٫۵۷

جدول ۲

لازم به ذکر است در کلاسبندی با الگوریتم GMM برای سرعت بیشتر در اجرای محاسبات داده های تست و آموزش به ابعاد ۵\*۵ نرمال شده اند.

طبق رابطه زیر 1-NN برای داده های با حجم بالا تقریباً بهینه عمل می کند:

$$P[\text{error}]_{\text{Bayes}} \leq P[\text{error}]_{1\text{NN}} \leq 2 P[\text{error}]_{\text{Bayes}}$$

#### ۵. نتیجه گیری

باتوجه به نتایج بدست آمده با 1-NN بیشترین دقت را بدست آوردیم ولی مدت زمان کلاسبندی این مدت نسبت به سایر الگوریتم ها از همه بیشتر و دلیل این امر lazy بودن KNN می باشد.

در LDA کمترین زمان اجرا را داشتیم ولی به دلیل برابر نبودن ماتریس کوواریانس همه کلاس ها به دقت بالایی دست نیافتیم [۳].

#### ۶. مراجع

[۱] Mengfei Cao, “Practice on Classification using Gaussian Mixture Model” Course Project Report for COMP-135

[۲] T. Young, D. Devamannyu, S. Poria, E. Cambria “Using discriminant analysis for multi-class classification” in Third IEEE International Conference on Data Mining,

[۳] Hu Yan ; Yu Dai “The Comparison of Five Discriminant Methods” 2011 International Conference on Management and Service Science