تمرین پنجم درس یادگیری ماشین
دکتر باباعلی

سیدعلیرضا مولوی

فهرست مطالب

۱ داده ها

داده ها از ۳۲۵۶۱ نمونه تشکیل شده اند و تعداد ویژگی ها ۱۴ عدد است. از ۱۴ ویژگی ۸ ویژگی اسمی[۱] و ۶ ویژگی عددی است.

هدف این است که با استفاده از این ویژگی ها بتوانیم تشخیص دهیم که آیا در آمد بیش از ۵۰ هزار دلار است و یا خیر، بنابراین خروجی مسئله ۲ حالت دارد پس مسئله دسته بندی دوتایی[۲] است.

تعداد نمونه های با درآمد کم تر از ۵۰ هزار دلار ۲۴۷۲۰ و تعداد نمونه های با درآمد بیش از ۵۰ هزار دلار ۷۸۴۱ است، بنابراین برچسب داده ها بالانس نیستند.

داده های آموزش و تست ۷۰ درصد داده ها رو به عنوان داده آموزشی و ۳۰ درصد را به عنوان داده های تست در نظر گرفته ایم.

پیش پردازش: برچسب نمونه های با درآمد کم تر و مساوی ۵۰ هزار دلار را ۰ و بقیه را ۱ در نظر می گیریم. پیش پردازش مختص به هر مدل در بخش مربوط به آن مدل توضیح داده شده است.

مدل پایه: اگر مدلی را در نظر بگیریم که برای تمام نمونه های ورودی مقدار درآمد کوچکر و مساوی ۵۰ هزار دلار را خروجی دهد، دقت به صورت زیر است:

$$TrainAcc = 0.7592 \quad TestAcc = 0.7592$$

---

[۱] Nomial

[۲] Binary Classification

۲ مدل بیزین ساده[۳]

در مدل بیزین ساده فرض می گیریم که ویژگی ها از توزیع های **مستقل** بدست می آید.

$$X = \begin{pmatrix} \cdots & x^{(1)} & \cdots \\ \cdots & x^{(1)} & \cdots \\ & \vdots & \\ \cdots & x^{(N)} & \cdots \end{pmatrix} \in R^{(N,d)} \quad (۱)$$

$$x = (x_1, x_2, \cdots, x_d)^T \in R^d \quad \text{A sample} \quad (۲)$$

$$\forall i (x_i \sim P_i) \quad P_i \text{ is a disturbution} \quad (۳)$$

$$\forall i, j (i \neq j \implies P_i \perp\!\!\!\perp P_j) \quad (۴)$$

هدف ما حل معادله ۵ است:

$$\hat{y} = argmax_y P(y|x) = argmax_y logP(y|x) \quad (۵)$$

$$logP(y|x) \propto logP(x|y) + logP(y) \quad (۶)$$

$$(4 \wedge 6) \implies logP(y|x) \propto logP(\prod_i^d x_i|y) + logP(y) \quad (۷)$$

$$\implies logP(y|x) \propto \sum_i^d logP(x_i|y) + logP(y) \quad (۸)$$

۱.۲ مقدار پیشین

که $logP(y)$ مقدار پیشین[۴] و $logP(x_i|y)$ احتمال[۵] است.مقدار پیشین را به صورت زیر در نظر می گیریم.

$$P(y_j) = \frac{n_j}{N} \quad n_j = |y = j| \quad (۹)$$

۲.۲ احتمال

۱.۲.۲ ویژگی های عددی

فرض می کنیم که به ویژگی های عددی از توزیع گاووسی بدست آمده اند، بنابراین:

$$P(x|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x-\mu_y)^2}{2\sigma_y^2}\right) \quad (۱۰)$$

$$logP(x|y) = -\frac{1}{2}log2\pi - \frac{1}{2}log\sigma_y^2 - \frac{(x-\mu_y)^2}{2\sigma_y^2} \quad (۱۱)$$

۲.۲.۲ ویژگی های اسمی

ویژگی های اسمی **گسسته** و **محدود** اند. فرض می کنیم که ویژگی های اسمی از توزیع چند جمله ای[۶] بدست آمده اند، بنابراین:

$$P(x_i|y) = \frac{n_{x_i \& y}}{n_y} \quad (۱۲)$$

---

[۳] Naive Bayes
[۴] Prior
[۵] Liklihood
[۶] MultiNomial

تصحیح لاپلاس

ممکن است در هنگام آموزش ما یک متغیر را اصلا مشاهده نکنیم بنابراین احتمال آن در هنگام تست صفر می شود، برای جلوگیری از این اتفاق می توان از تصحیح لاپلاس استفاده کرد:

$$P(x_i|y) = \frac{n_{x_i \& y} + \alpha}{n_y + \alpha m}$$

α is laplace factor. m is number of unique outputs.

که α متغیر لاپلاس است و m تعداد اسم های منحصر به فرد است.

۳.۲ پیش پردازش داده

در هنگام استفاده از مدل بیزین ساده تنها کافی است که در هنگام ساختن مدل، توزیع هر ویژگی را مشخص کنیم و این توزیع باید با نوع ویژگی سازگار باشد؛ مثلا نمیتوان برای متغیر های عددی از توزیع چند جمله ای استفاده کرد.
لازم نیس که متغیر های عددی را نرمالیز کنیم، زیرا داریم به طور مستقیم توزیع را محاسبه می کنیم.

۴.۲ نتایج آموزش

اگر فرض بگیریم تمام متفیر ها از توزیع چند جمله ای بدست می آیند (متغیر های عددی را می توانیم بدون هیچ پیش پردازشی از نوع متغیر اسمی در نظر بگیریم) آنگاه:

$$TrainAcc = 0.8178 \quad TestAcc = 0.8204$$

و اگر فرض بگیریم که متغیر های اسمی از توزیع چند جمله ای و متغیر های عددی از توزیع گاووسی باشند آنگاه:

$$TrainAcc = 0.7957 \quad TestAcc = 0.7955$$

زمانی که فرض بگیریم که تمام ویژگی ها اسمی اند دقت بهتر است. در هیچکدام از حالت ها overfit رخ نداده است ولی مشخص است که underfit رخ داده است.

۳ رگرسیون خطی

۱.۳ مدل محاسباتی

ما حالت کلی رگرسیون را برای دسته بندی چند تایی در نظر می گیریم. بنابراین باید ازتابع فعال سازی softmax استفاده کنیم.

$$x = (x_1, x_2, \cdots, x_d)^T \in R^d \quad y \in R^{d'} \tag{۱۳}$$

$$z = Wx + b \quad W \in R^{(d',d)} \quad b \in R^{d'} \tag{۱۴}$$

$$z_i = \sum_j w_{ij} x_j + b_i \tag{۱۵}$$

$$\hat{y} = softmax(z) \in R^{d'} \tag{۱۶}$$

$$\hat{y_i} = \frac{epx(z_i)}{\sum_j^{d'} exp(z_j)} \tag{۱۷}$$

۲.۳ مدل محاسباتی برای حالت برداری

$$X = \begin{pmatrix} \cdots & x^{(1)} & \cdots \\ \cdots & x^{(1)} & \cdots \\ & \vdots & \\ \cdots & x^{(N)} & \cdots \end{pmatrix} \in R^{(N,d)} \tag{۱۸}$$

$$Z = \begin{pmatrix} \vdots \\ z^{(i)^T} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \left(Wx^{(i)} + b\right)^T \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ x^{(i)^T} W^T + b \\ \vdots \end{pmatrix} \tag{۱۹}$$

$$Z = XW^T + b \in R^{(N,d')} \tag{۲۰}$$

۳.۳ تابع خطا

به دلیل استفاده از تابع softmax خطای را Cross Entropy در نظر می گیریم.

$$L(y, \hat{y}) = -\sum I(y_i) log \hat{y_i} \tag{۲۱}$$

$$L = \frac{1}{N} \sum L(y^{(i)}, \hat{y}^{(i)}) \quad \text{N is number of training samples} \tag{۲۲}$$

۴.۳ آموزش مدل

ما مقادیر متغیر ها را به طور تصادفی انتخاب می کنیم و سپس از طریق الگوریتم کاهش گرادیان وزن های بهینه که خطا را کمینه می کنند، بدست می آوریم.

۱.۴.۳ مقادیر اولیه وزن ها

مقادیر اولیه بایاس را ۰ قرار می دهیم و مقایر اولیه وزن ها را از توزیع نرمال با پارامتر های زیر بدست می آوریم که d و d' بعد ورودی و خروجی است.

$$W_{i,j} \sim N\left(\mu = 0, \sigma^2 = \frac{d + d'}{2}\right)$$

$$b_i = 0$$

۲.۴.۳ کاهش گرادیان

برای آموزش مدل و یافتن وزن های مناسب، از الگوریتم کاهش گرادیان[۷] استفاده می کنیم.

$$W = W - \alpha \nabla_W L = W - \frac{\alpha}{N} \sum \nabla_W L(y^{(i)}, \hat{y}^{(i)})$$
$$b = b - \alpha \nabla_b L = b - \frac{\alpha}{N} \sum \nabla_b L(y^{(i)}, \hat{y}^{(i)})$$

α is learning rate.

البته می توان به جای مشتق میانگین تمام خطا ها، مشتق میانگین تعداد محدود تری از داده ها را محاسبه کنیم، به این روش Mini Batch GD گفته می شود.

کافی است که مشتقات را بدست بیاوریم:

$$\frac{\partial L(y, \hat{y})}{\partial W_{ij}} = \sum_{k}^{d'} \left(\frac{\partial L}{\partial \hat{y}_k} \frac{\partial \hat{y}_k}{\partial z_i} \frac{\partial z_i}{\partial W_{ij}} \right) \quad (۲۳)$$

$$\frac{\partial L(y, \hat{y})}{\partial b_i} = \sum_{k}^{d'} \left(\frac{\partial L}{\partial \hat{y}_k} \frac{\partial \hat{y}_k}{\partial z_i} \frac{\partial z_i}{\partial b_i} \right) \quad (۲۴)$$

بنابراین کافی است که مشتق هر عبارت را محاسبه کنیم و در معادلات ۲۳ و ۲۴ جایگزینی کنیم.

$$\frac{\partial L(y, \hat{y})}{\partial \hat{y}_j} = \frac{-I(y_j)}{\hat{y}_j} \quad (۲۵)$$

$$\frac{\partial \hat{y}_j}{\partial z_i} = \hat{y}_i(\delta_{ij} - \hat{y}_j) \quad (۲۶)$$

$$\frac{\partial z_i}{\partial W_{ij}} = x_j \quad (۲۷)$$

$$\frac{\partial z_i}{\partial b_i} = 1 \quad (۲۸)$$

۳.۴.۳ پایداری کاهش گرادیان

در ابتدای پروسه آموزشی امکان این است که مقادیر خطا بسیار بزرگ باشند بنابراین مقادیر گرادیان ها نیز بسیار بزرگ می شوند؛ اگر بخواهیم پروسه کاهش گرادیان پایدار باشد باید مقدار نرخ آموزشی را بسیار کوچک در نظر بگیریم و یا اینکه مقادیر گرادیان را محدود کنیم.

ما تنها نیاز به جهت گرادیان داریم (اگر در خلاف جهت گرادیان حرکت کنیم خطا کاهش می یابد)، بنابراین می توانیم فقط از نرم گرادیان استفاده کنیم.

۵.۳ پایداری عددی

به دلیل اینکه اعداد در کامپیوتر به صورت گسسته و محدود ذخیره می شوند، انجام عملیات ریاضی معمولا توام با خطاست. باید تغییراتی در کد ایجاد شود که خطای های عددی را محدود نگه دارید.

۱.۵.۳ پایداری عددی Softmax

در محاسبه softmax ما از تابع نمایی استفاده می کنیم. اگر مقدار ورودی تابع نمایی بسیار بزرگ باشد، در نتیجه overflow رخ می دهد و مقدار صورت و مخرج هر دو inf می شود که در نتیجه مقدار کسر NaN[۸] می شود و کل

[۷] Gradient Decent
[۸] Not a Number

محاسبات به هم میریزد. می توانیم از تکنیک زیر استفاده کنیم و سایز ورودی تابع نمایی را محدود به صفر کنیم.

$$M = max_i(z_i) \implies \forall i(z_i - M \leq 0) \quad (۲۹)$$

$$softmax(z) = \frac{epx(z_i - M)}{\sum_j exp(z_j - M)} \quad (۳۰)$$

$$= \frac{epx(z_i)exp(-M)}{\sum_j exp(z_j)exp(-M)} = \frac{epx(z_i)}{\sum_j exp(z_j)} \quad (۳۱)$$

۲.۵.۳ پایداری عددی تابع خطا

برای برقراری روابط برقرار باشند:

$$0log0 = 0 \quad 0(\frac{1}{0}) = 0$$

ولی در پایتون $log0 = -\inf$ و $0log0 = NaN$ و محاسبات به هم میریزد. برای رفع این مشکل:

$$\hat{y} = \begin{cases} \hat{y} + \epsilon & \hat{y} = 0 \\ \hat{y} - \epsilon & \hat{y} = 1 \end{cases} \quad (۳۲)$$

۶.۳ پیش پردازش داده

داده های ورودی رگرسیون باید عددی باشند. برای تبدیل متغیر های اسمی به متغیر های عددی از **رمزگذار عددی**[۹] استفاده می کنیم.

نحوه کار رمزگذار عددی: به هر اسم منحصر به فرد در متغیر اسمی، یک عدد نسبت می دهیم.
در هنگام استفاده از رگرسیون **بهتر** است که ویژگی داده ها نرمالیز کنیم، این کار سبب افزایش دقت و کاهش زمان همگرایی الگوریتم می شود.

$$\mu_i = \frac{1}{N}\sum_n x_i^{(n)} \quad (۳۳)$$

$$\sigma_i^2 = \frac{1}{N}\sum_n \left(x_i^{(n)} - \mu_i\right)^2 \quad (۳۴)$$

$$x_{i,norm} = \frac{x_i - \mu}{\sqrt{\sigma_i^2 + \epsilon}} \quad (۳۵)$$

دلیل استفاده از ϵ جلوگیری از underflow و صفر شدن مخرج است.

۷.۳ نتایج آموزش

برای آموزش ابرپارامتر ها به صورت زیر در نظر گرفتیم:

- اندازه MiniBatch را ۲۵۶ در نظر گرفتیم.
- نرخ آموزشی را ۰/۲۵ در نظر گرفتیم.
- انداز نرم گرادیان ها را به ۱ محدود کرده ایم.
- تعداد تکرار ها را ۲۰۰۰ در نظر گرفتیم (تکرار آموزش بر روی Mini Batch)

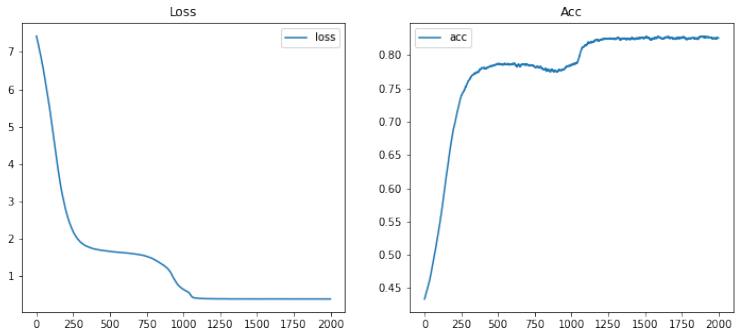
در شکل ۱ کارکرد مدل بر روی داده آموزشی در طول آموزش نمایش داده شده است. دقت بر روی داده آموزشی و تست به صورت زیر است:

$$TrainAcc = 0.8258 \quad TestAcc = 0.8228$$

با توجه به نتایج overfit رخ نداه است، بلکه underfit رخ داده است.

[۹] Ordinal Encoder

شکل ۱: دقت و خطا بر روی داده آموزشی در طول آموزش



۸.۳ رگرسیون برای دسته بندی ۲ تایی

در کد رگرسیون دسته بندی چند تایی[۱۰] از طریق تابع فعال سازی Softmax پیاده سازی شده است. حالت دسته بندی ۲ تایی حالت خاصی از دسته بندی چند تایی است. (تابع خطا BinaryCrossEntropy و تابع فعال سازی Sigmoid به همراه مشتقات شان در کد پیاده سازی شده، ولی در فرآیند رگرسیون استفاده نشده اند.)

[۱۰] Multilabel Classification