به نام خدا

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

گزارش تمرین دوم درس یادگیری ماشین

پاسخ مسئله‌های تشریحی

استاد درس: دکتر احسان ناظرفرد

دانشجو: فاطمه غلام‌زاده

۹۹۱۳۱۰۰۳

نیم سال دوم ۱۳۹۹-۱۴۰۰

سوال ۱)

از بیش برازش شدن جلوگیری می‌کند. زمانی که عمق درخت زیاد می‌شود لازم است هرس کردن انجام شود. با هرس کردن شاخه‌هایی که داده‌های کمی دارند و به یک پدر متعلق هستند را یکی می‌کند و برای هر کلاس احتمال آن‌را نشان می‌دهد.

روش های جلوگیری از بیش برازش در درخت تصمیم:

۱- Pre-pruning : درختی با تعداد شاخه‌های کمتری از درختی که باید تولید می‌شد ساخته می‌شود. یعنی پیش از آن که درخت تمام داده‌های آموزش را دسته بندی کند رشد آن را متوقف می‌کنیم.
۲- Post-pruning: درخت به طور کامل ساخته می‌شود و بعد قسمت‌هایی از آن حذف می‌شود. یعنی درخت به طور کامل داده ها آموزش را دسته بندی می‌کند و بعد هرس می‌شود.
۳- فیچرهای غیرمرتبط و نامناسب می‌توانند موجب بیش برازش در درخت تصمیم شوند بنابراین حذف این فیچرها می‌تواند از بیش برازش جلوگیری کند.
۴- افزایش تعداد داده‌های آموزش.

سوال۲)

الف) ماتریس در هم ریختگی به صورت زیر است:

```
=== Confusion Matrix ===

  a  b   <-- classified as
 14  6 |  a = bad
  9 28 |  b = good
```

FN = 6 , TP = 14, TN = 28, FP = 9

Precision = TP/(TP+FP) = 14/23 = 0.6

Recall = TP/(TP+FN) = 14/20 = 0.7

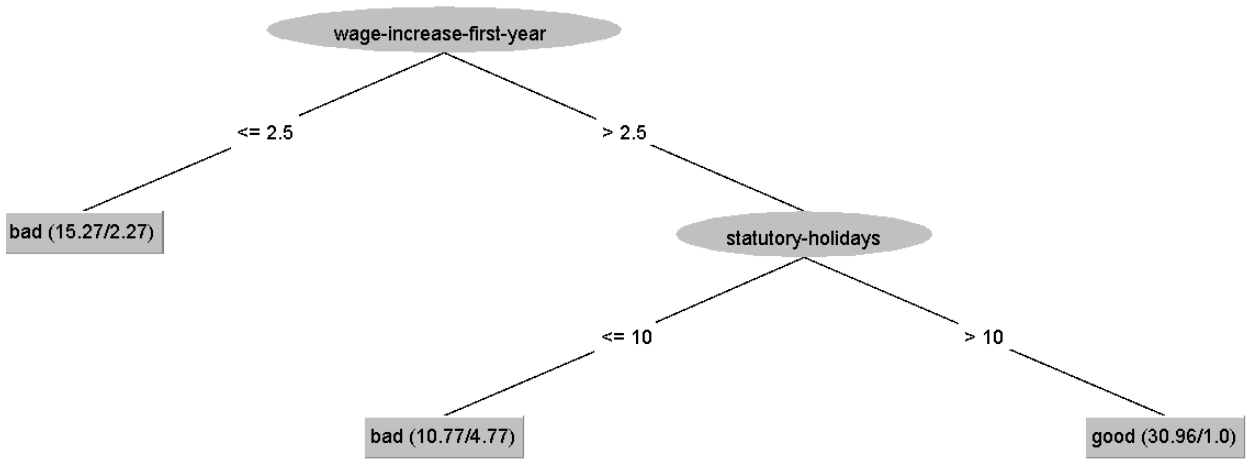
$$Accuracy = \frac{\#TP+\#TN}{\#P+\#N} = \frac{\#TP+\#TN}{\#TP+\#FN+\#TN+\#FP}$$

⇨ Accuracy = 14+28/57=44/57=0.77

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

⇨ F1-measure = 2*(0.6)*(0.7)/ (1.3) = 0.64

درخت تصمیم ساخته شده به صورت زیر است:



داده‌ی مورد سوال، در کلاس b یعنی good قرار می‌گیرد. زیرا ابتدا فیچر wage-increase-first-year مورد بررسی قرار می‌گیرد که مقدار آن ۳ است پس به شاخه سمت راست می‌رویم (بزرگتر از ۲,۵)

بعد فیچر statutory-holidays مورد بررسی قرار می‌گیرد که برای داده‌ی مورد سوال مقدار آن ۱۲ است پس به شاخه سمت راست می‌رویم که کلاس این داده مشخص می‌شود و good است.

ب) پارامتر unpruned این مورد را تعیین می‌کند که درخت تصمیم هرس بشود یا خیر، اگر مقدار آن False باشد درخت ساخته شده هرس می‌شود و اگر مقدار آن True باشد درخت هرس نمی‌شود. در واقع در حالتی که عمق درخت زیاد شود و به اصطلاح overfitting رخ دهد، با هرس کردن درخت بعضی از شاخه‌ها حذف شده و در ریشه ی مربوطه احتمال آنها نوشته می‌شود. انتظار میرود که با فعال کردن این گزینه عمق درخت بیشتر شود.

در این قسمت پارامتر unpruned به true مقداردهی شد. ماتریس درهم‌ریختگی به صورت زیر در آمد:

```
=== Confusion Matrix ===

  a  b   <-- classified as
 14  6 |  a = bad
  6 31 |  b = good
```

FN = 6 , TP = 14, TN = 31, FP = 6

Precision = TP/(TP+FP) = 14/20 = 0.7

Recall = TP/(TP+FN) = 14/20 = 0.7

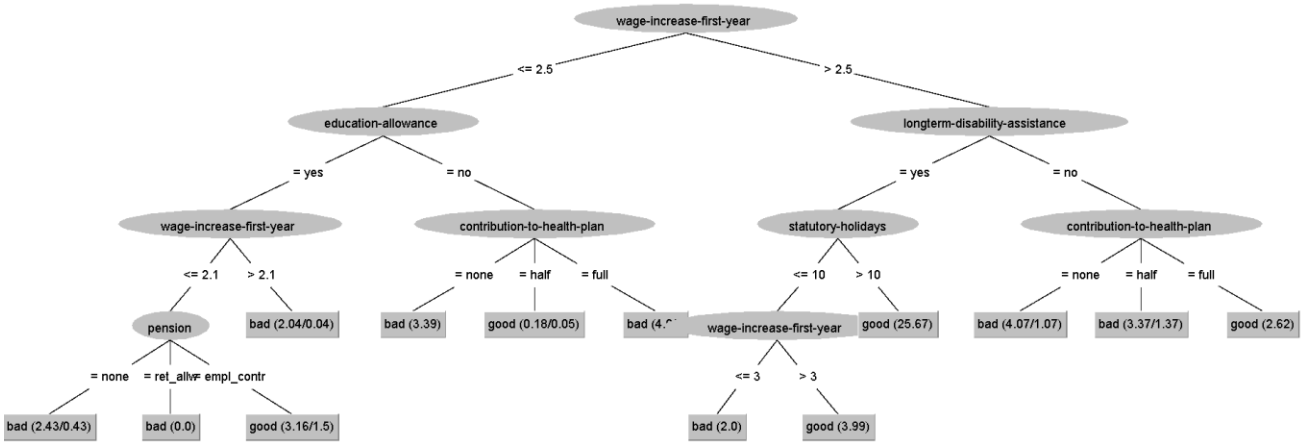
$$Accuracy = \frac{\#TP+\#TN}{\#P+\#N} = \frac{\#TP+\#TN}{\#TP+\#FN+\#TN+\#FP}$$

⇨ Accuracy = 14+28/57=45/57=0.78

$$F_1 = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} = \frac{\mathrm{TP}}{\mathrm{TP} + \frac{1}{2}(\mathrm{FP} + \mathrm{FN})}$$

F1-measure = 2*(0.7)*(0.7)/ (1.4) = 0.7

درخت تصمیم ساخته شده به این صورت است:



برای داده مورد سوال ابتدا فیچر wage-increase-first-year بررسی می‌شود و چون ۳<۲,۵ است به شاخه سمت راست می‌رویم. سپس فیچر longterm-disability-assistant بررسی می‌شود و چون yes است به شاخه سمت چپ می‌رویم. برای فیچر statutory-holidays چون مقدارش ۱۲ است و از ۱۰ بیشتر است به شاخه سمت راست می‌رویم و کلاس داده برابر b می‌شود.

تفاوت درخت‌ها: در این قسمت چون پارامتر unpruned را true کردیم درخت رسم شده هرس نشده و عمق و عرض آن بیشتر است و فیچرهای بیشتری در آن مورد بررسی قرار می‌گیرد. با توجه به دقت‌های به دست آمده مشاهده می‌شود که با هرس نکردن درخت، نسبت به حالت هرس شده دقت ها افزایش یافته اند.

سوال ۳)

بله افزایش می‌یابد. هر چقدر که ابعاد داده‌ها بیشتر باشد این فضای با ابعاد بالا با چگالی و فشردگی کمتری از نقاط داده‌های آموزش روبه روست و نیاز داریم تا حجم زیادی از فضا را جستجو کنیم تا همسایه‌های داده تست را پیدا کنیم. هم‌چنین با افزایش ابعاد داده‌ها، فاصله دوبه دوی میان نقاط نیز افزایش پیدا می‌کند. در این صورت همسایه‌های یک داده تست ممکن است به حدی از آن دور باشند که در واقع هیچ وجه اشتراکی با این داده نداشته باشند و این امر موجب خطا در دسته بندی می‌شود. به این مسئله، Curse of dimensionality گفته می‌شود.

فرض کنیم که داده‌های آموزش در یک فضای D بعدی به طور یکنواخت در مکعب واحد پراکنده شده‌اند و هم چنین فرض کنیم که می‌خواهیم چگالی برچسب‌های کلاس ها را با رشد یک مکعب واحد در اطراف داده تست تخمین بزنیم. یعنی مکعب آن قدر رشد می کند تا تعداد داده مورد نظر ما در آن قرار بگیرد و فرض کنیم این تعداد مورد نظر f باشد. امید ریاضی طول ضلع این مکعب از رابطه زیر به دست می‌آید:

$$e_D(f) = f^{\frac{1}{D}}$$

این رابطه نشان می‌دهد که سایز ضلع مکعب مورد نیاز برای اینکه تعداد مشخصی داده آموزشی در آن قرار بگیرد، با افزایش ابعاد داده‌ها به شدت افزایش پیدا می‌کند. شکل زیر این مورد را نشان می‌دهد:

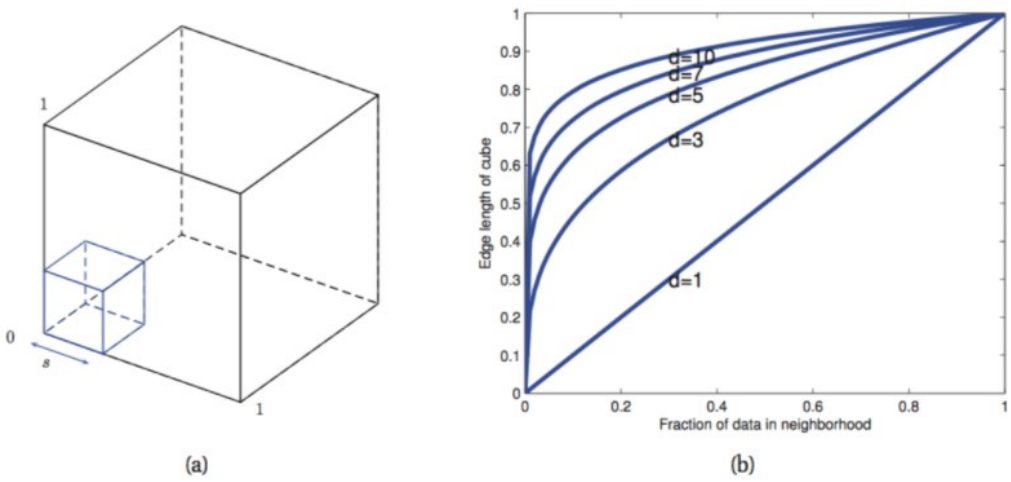


Figure 1.16 Illustration of the curse of dimensionality. (a) We embed a small cube of side s inside a larger unit cube. (b) We plot the edge length of a cube needed to cover a given volume of the unit cube as a function of the number of dimensions. Based on Figure 2.6 from (Hastie et al. 2009). Figure generated by `curseDimensionality`.

سوال۴)

مدل های مولد توزیع واقعی داده ها را مدل می کنند و مدل ها تمایزگر یک مرز تصمیم را مدل میکنند. در الگوریتم های مولد دسته بند مقادیر $p(Y|X)$ را محاسبه می‌کند. برای محاسبه ی این احتمال $P(X|Y)P(Y)$ را بدست می‌آورد. با این محاسبات الگوریتم توزیع داده‌ها را فرا می‌گیرد و می‌تواند خودش داده‌ها را دوباره تولید کند.مدل تمایزگر (Discriminative) برای دسته بندی تنها یک مرز تصمیم را می‌آموزد و بر اساس آن دسته‌بندی را انجام می دهد. مدل مولد احتمال توزیع توام $p(x,y)$ را می آموزد. این مدل احتمال شرطی را با کمک قضیه بیز پیش بینی می کند اما مدل تمایزگر احتمال توزیع شرطی $p(y|x)$ را فرا میگیرد.

نمونه هایی از دسته بندهایی که مولد هستند:

- Naïve Bayes
- Bayesian networks
- Markov random fields
- Hidden Markov Models (HMM)

نمونه هایی از دسته بند های تمایزگر:

- Logistic regression
- Scalar Vector Machine
- Traditional neural networks
- Nearest neighbour
- Conditional Random Fields (CRF)s

سوال۵)

این دو دسته بندی کننده از دو نوع مختلف دسته بندی کننده های generative و discriminative هستند. بیز ساده از نوع generative ؛یعنی مولد است. در الگوریتم های مولد دسته بند مقادیر $p(Y|X)$ را محاسبه می کند. برای محاسبه ی این احتمال $P(X|Y)P(Y)$ را بدست می‌آورد. با این محاسبات الگوریتم توزیع داده‌ها را فرا می‌گیرد و می‌تواند خودش داده‌ها را دوباره تولید کند. همانطور که میدانیم، الگوریتم بیز ساده این احتمال ها را بدست می‌آورد.

رگرسیون لاجستیک از نوع discriminative ؛ یعنی تمایزدهنده است. به این معنا که برای دسته بندی تنها یک مرز تصمیم را می‌آموزد و بر اساس آن دسته‌بندی را انجام می دهد. در بیز ساده ابتدا احتمال $p(Y|X)$ به شکل زیر بازنویسی می‌شود. سپس مقادیر $P(X|Y)$ از داده‌های آموزشی محاسبه می‌شود.

$$\mathbf{X} = < X_1, X_2, \dots, X_n > \ , \ X_i \perp\!\!\!\perp X_j \mid Y \quad (s.t:\ i,j = 1,2,\dots,n \ \&\ i \neq j)$$

$$y_{MLE} = \underset{y\in\{+,-\}}{\operatorname{argmax}}\ p(\boldsymbol{X}, Y = y) = \underset{y\in\{+,-\}}{\operatorname{argmax}}\ p(\boldsymbol{X}|Y = y)p(Y = y)$$

$$= \underset{y\in\{+,-\}}{\operatorname{argmax}}\ p(X_1|Y = y)p(X_2|Y = y)\dots p(X_n|Y = y)p(Y = y)$$

$$= \underset{y\in\{+,-\}}{\operatorname{argmax}}\ p(Y = y)\prod_{i=1}^{n} p(X_i = x_i|Y = y)$$

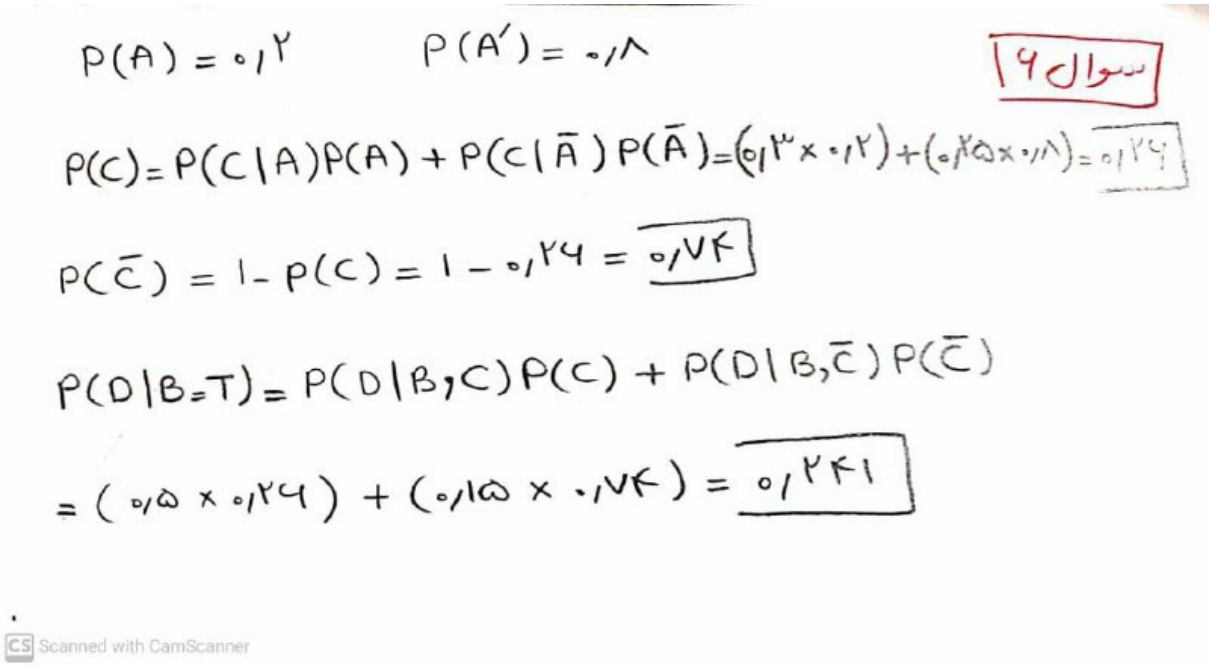
رگرسیون لاجستیک برای دسته‌بندی مقدار $p(Y|X)$ را مستقیما محاسبه می کند و بر اساس مقدار آن دسته‌بندی را انجام می‌دهد . در ادامه توابعی که احتمال از آنها محاسبه میشود برای حالت دو کلاسه آورده شده است. رگرسیون لاجستیک از تابع سیگموئید برای محاسبه کلاس مثبت استفاده می‌کند و ورودی این تابع یک ترکیب خطی از ویژگیهای مختلف داده‌هاست. پس در واقع لاجستیک رگرسیون یک دسته‌بندی کننده خطی است. وزنهای این ترکیب خطی به کمک داده‌های آموزشی و روش های بهینه‌سازی بدست می‌آیند. (البته لاجستیک غیرخطی نیز وجود دارد که ترکیب غیر خطی ویژگی‌ها به عنوان ورودی تابع سیگموئید در نظر گرفته می‌شود.)

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

$$P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

مقایسه عملکرد مدلهای لاجستیک و بیزساده‌ی گوسی:

بیز ساده بایاس زیادی نسبت به داده‌های آموزش دارد. رگرسیون لاجستیک واریانس زیادی دارد که می‌توان میزان آن را با تغییر در تابع بهینه‌سازی بهبود بخشید .اگر تعداد دادههای آموزش به سمت بی‌نهایت برود و بیز ساده مفروضاتی مانند برابر بودن پراکندگی دادهها در کلاس های مختلف داشته باشد ، نتایج دو مدل همگرا است یعنی هر دو مدل یک نتیجه را می‌دهند . اما در غیر این صورت رگرسیون لاجستیک به دلیل اینکه پارامترهای خود را براساس تمام داده‌ها محاسبه می‌کند دقت بیشتری نسبت به بیز ساده دارد.



سوال ۷)

شکل ۳ :

برچسب داده ی تستA است. میانگین داده های A و B در راستای X1 و X2 تقریبا با هم برابر است. اما واریانس داده های A در هر دو راستا از B بیشتر است. هم چنین احتمال اولیه‌ی تعلق به کلاس A بیشتر است چون نسبت تعداد داده های A به کل داده ها از تسبت تعداد داده های B به کل داده ها بیشتر است. بنابراین ۲ مورد از ۳ موردی که در هم ضرب می شوند برای A بزرگتر و مورد سوم هم تقریبا با هم برابر است و احتمال تعلق داده تست به کلاس A بیشتر خواهد شد.

شکل۴:

برچسب داده ی تست B است. چون در راستای x1 انحراف معیار B از A بیشتر است و احتمال متعلق بودن داده ی تست به کلاس B بیشتر مساوی از احتمال متعلق بودن آن به کلاس A است. از طرف دیگر در راستای x2 نیز احتمال متعلق بودن داده‌ی تست به کلاس B بیشتر است. چون اگر فرض کنیم میانگین هر دو کلاس در یک نقطه است و داده ی تست روی میانگین قرار دارد، چون انحراف معیار A بیشتر از B است پس ارتفاع قله ی B بیشتر از A است و در نتیجه احتمال متعلق بودن داده ی تست به کلاس B بیشتر است. و دلیل آخر اینکه احتمال اولیه ی کلاس B نیز

بیشتر از کلاس A است. به همین سه دلیل زمانی که احتمال‌ها در هم ضرب می شوند حاصل نهایی برای کلاس B بیشتر می‌شود و داده ی تست برچسب کلاس B را می‌گیرد.

سوال ۸)

عمل regularization موجب می‌شود که بایاس مدل افزایش یابد و overfitting اتفاق نیفتد. وقتی لاندا را افزایش می‌دهیم در واقع این بایاس را بیشتر می‌کنیم اما وقتی پارامتر لاندا را بیش از اندازه بزرگ کنیم میزان فیت شدن آن به داده به حدی کاهش می‌یابد که موجب می‌شود مدل به درستی داده‌های آموزش را فرا نگیرد، در نتیجه عملکرد تست و train هر دو بدتر می‌شود و بهم نزدیک می‌شود.