

درس یادگیري ماشین

مدرس: دکتر سامان هراتی زاده

تمرین شمارهي شش)جبرانی(

1395/11

830595010

هادی رهجو

در این تمرین به حل یک مسئله با استفاده از روش های بیزی پرداخته می شود.

صورت مسئله دسته بندی متن ها بر اساس گروه آن ها است. دسته داده مورد استفاده Data.csv است. در این دسته داده، متن هایی به صورت عددی به فرمتی مشابه یک ماتریس قرار داده شده است. هر متن دارای تگ هایی در سر تیتر و بدنه متن است. در دسته داده، متن ها بر اساس موضوع و گروه قابل دسته بندی هستند. این دسته بندی به این صورت است که متن ها در گروه های مختلف تقسیم شده اند و چند گروه می توانند در قالب موضوعی عام تر قرار گیرند.

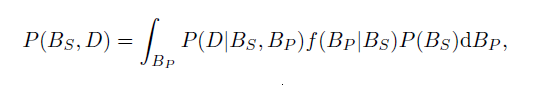
گروه بندی ها و تقسیم موضوعات در شکل بالا قرار داده شده اند.

استفاده از شبکه باور بیزی برای دسته بندی

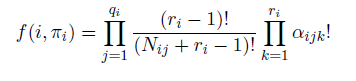
با استفاده از دسته بندی کننده بیزی ، دسته بندی کننده ای بسازید که بتواند کلاس1 را از 2 تشخیص دهد.

هدف مسئله دسته بندی متن ها بر اساس **گروه** **ها** است.

برای پیاده سازی این تمرین از روش شبکه بیزین استفاده میکنیم و برای رسیدن از داده به درخت روشی به نام K2 ارائه شده که می تواند درخت نسبتا مناسبی از روی داده ها ، برای دسته بندی داده بسازد فرمول آن :



که پس از خلاصه سازی و اثبات ریاضی به این فرمول میرسیم که :



بنده پیاده سازی این الگوریتم را به صورت ماژولار و با در نظر گرفتن توابع مختلف پیاده سازی نمودم :

تابع factoriel یک عدد را میگیرد و فاکتوریل آن را بر میگرداند

تابع read\_database یک آدرس را خوانده آن را در آرایه Numpy میریزد و سطر و ستون ردیف را حذف میکند و برمیگرداند

تابع DiffrageDS شماره ستون مورد نظر یعنی Xi را میگیرد و یک بردار از پدران Xi می گیرد و یک آرایه که حاوی دیتاست است و ابتدا بر اساس مقادیر موجود در هر یک از پدران دیتاست را به دو یا چند تکه میشکند و در آخر بر اساس مقادیر ستون Xi هر یک از تکه ها را به Ri مقدار تقسیم مینماید و Nij و دیتاست دسته بندی شده را باز می گرداند

تابع Equation که دیتاست و بردار پدران و شماره ستون مورد نظر با از ورودی دریافت میکند به تابع DiffrageDS پاس می دهد ، خروجی حاصل از صدا زدن آن تابع را که همان آلفا ها و Nij هاست را درون فرمول قرار می دهد و مقدار را بر میگرداند.

تابع GetAllPosibleParent به عنوان ورودی شماره ستون مورد نظر را میگیرد و به عنوان خروجی تمام پدران ممکن برای آن را در قالب یک بردار بر میگرداند .

تابع K2 که اصل کد الگوریتم را اجرا میکند که در آن با صدا زدن دو تابع قبل ماتریس مجاورت را تشکیل می دهد و برمیگرداند

تابع Cv که مربوط به Cross validation است و با وارد کردن مقدار K و دیتاست مربوطه K Fold را به اجرا می گذارد و در هر بار اجرا تابع K2 فراخوانی میشود و سپس Test انجام میگردد.

تابع P که همان تابع محاسبه شرط ماست و یک رشته ورودی شرطی را از ورودی دریافت می کند و به همراه دیتاست ورودی نتایج شرط را در قالب یک عدد اعشاری بر میگرداند. ( مثال : P(‘ X1 = 0 | X2 =1 , X3 = 1)

و تابع BayesianNet\_Test که با گرفتن دیتاست های Train , Test و ماتریس مجاورت حاصل از تابع K2 به پیشگویی مقادیر هر سطر و مقایسه آن با Lable میپردازد و مقادیر Accuracy , precision , recall , F measure را اعلام می نماید

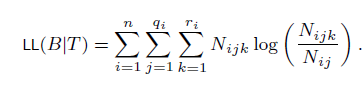
Accuracy = 0.7629629629629631

Precison = 0.8615384615384615

Recall = 0.7422680412371133

F1 SCORE = 0.7974683544303798

LLC : در این رویکرد چون شروط Likely hood عدد بسیار کوچکی میشود از Log Likelyhood استفاده می شود و P(C) و P(E) از فرمول حذف می شود . ضعف این رویکرد درآن است که این الگوریتم با فرض مستقل بودین ویژگی ها با پیاده سازی می پردازد فلذا الگوریتم برای داده های غیر مستقل چندان پاسخ ارزشمندی نخواهد داشت . یک مشکل دیگر که ممکن است در اینجا بروز کند پیچیدگی بیش از حد گراف خواهد بود



TP = 99.2 TN = 64.8 FP = 16.0 FN = 36.0

Accuracy = 0.7592592592592593

Precison = 0.8611111111111112

Recall = 0.7337278106508877

F1 SCORE = 0.792332268370607

AIC : برای مرتفع کردن مشکل وقوع پیچیدگی گراف در LLC ما می آیم یک تابع جریمه به آن اضافه می کنیم تا به ازای هر تعداد یال مقدار جریمه افزایش بیابد ، ضریب آن دلخواه است ولی در AIC غالبا 1 در نظر گرفته می شود



TP = 96.6 TN = 65.2 FP = 16.4 FN = 37.8

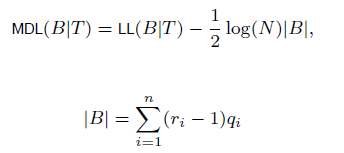
Accuracy = 0.7490740740740741

Precison = 0.8548672566371681

Recall = 0.7187500000000001

F1 SCORE = 0.7809215844785772

MDL : در این روش نیز به مانند AIC ما یک تابع جریمه اعمال میکنیم که ضریب آن دیگر این صفر نیست



TP = 99.0 TN = 63.0 FP = 18.0 FN = 36.0

Accuracy = 0.75

Precison = 0.8461538461538461

Recall = 0.7333333333333333

F1 SCORE = 0.7857142857142857