

پروژه درس هوش مصنوعی

شیرین محمدی - ۹۷۱۲۰۴۶

۲۰۲۲ January

۱ تعریف مسئله

اگر شبکه اجتماعی را یک گراف $G = (V, E)$ در نظر بگیریم یک اجتماع را میتوان مجموعه ای از رئوس تعریف کرد که تراکم یال ها در بین رئوس داخل آن زیاد و در بین رئوس یک اجتماع و اجتماع دیگر کمتر می باشد

برای دسترسی به گراف از ماتریس مجاورت مربوط به آن استفاده میکنیم و با استفاده از داده های فایل ماتریس مجاورت گراف مورد نظر را می سازیم. حال مسئله تشخیص اجتماعات تبدیل به بخش بندی ماتریس مجاورت به تعدادی زیرماتریس که مجموع تراکم زیرماتریس ها را ماکسیمم میکند. پس اینجا مفهومی به نام *communityscore* معرفی می شود.

از آنجاییکه گراف بدون جهت است ماتریس مجاورت متقارن می باشد. فرض میکنیم $S = (I, J)$ زیرماتریس ماتریس مجاورت A می باشد. همچنین فرض میکنیم $a_{i,j}$ میانگین i امین ردیف ماتریس S می باشد و مشابه $a_{I,j}$ میانگین j امین ستون ماتریس S باشد به عبارت دیگر:

$$a_{I,j} = \frac{1}{|I|} \sum_{i \in I} a_{ij} \text{ and } a_{i,J} = \frac{1}{|J|} \sum_{j \in J} a_{ij}$$

همچنین v_s را بصورت زیر تعریف میکنیم:

$$v_s = \sum_{i \in I, j \in J} a_{ij}$$

با داشتن زیرماتریس S میانگین توان S از مرتبه r که $M(S)$ نام گذاری می کنیم بصورت زیر تعریف می شود:

$$M(S) = \frac{\sum_{i \in I} (a_{i,J})^r}{|I|}$$

حال با داشتن این مقادیر امتیاز زیرماتریس S بصورت $Q(S) = M(S) * v_s$ تعریف می شود در نتیجه امتیاز اجتماع بصورت زیر محاسبه می شود:

$$CS = \sum_i^k Q(S_i)$$

۲ الگوریتم رتتیک

در این الگوریتم n ها و جزئیاتشان نمایانگر گراف $G = (V, E)$ می باشند که یک شبکه اجتماعی SN را مدل می کند. همچنین مقدار j داده شده به i امین n بیانگر وجود یک لینک بین نود i و j در V می باشد.

۱.۲ تابع هدف

میدانیم بررسی یک فرد حالت های متفاوتی از بخش بندی گراف را ارائه می دهد. ما می خواهیم این بخش بندی دارای بیشترین امتیاز اجتماع باشد. پس همانطور که در بخش قبلی گفته شد تابع هدف ما $CS = \sum_i^k Q(S_i)$ می باشد.

۲.۲ مقدار دهی اولیه

فرآیند مقدار دهی اولیه ما شامل در نظر گرفتن اتصالات موثر در بین گره هاست. انتخاب تصادفی ممکن است مولفه هایی به ما ارائه دهد که در گراف اصلی به هم متصل نمی باشند، به عنوان مثال ممکن است مقدار j در i امین جایگاه وجود داشته باشد در حالی که نود (i, j) در گراف ناموجود است. جهت رفع این مسئله، باید وجود یال موثر را بررسی کنیم و در صورت عدم وجود j را با یکی از همسایگان i تعویض کنیم. این روند موجب همگرایی سریعتر الگوریتم ما می شود

۳.۲ cross over یکنواخت

در اینجا ما از cross over یکنواخت استفاده می کنیم چراکه وجود یال موثر بین نود ها را تضمین می کند. با توجه به مقداردهی اولیه میدانیم که اگر هر ژن i که دارای j باشد آنگاه نود (i, j) وجود دارد. حال انتخاب ژن ها به اینصورت است که بردار مربوط به والد اول برابر با یک و بردار مربوط به والد دوم برابر با صفر است بنابراین فرزند در جایگاه i حاوی مقدار j از طرف یکی از والدین خود هست و در نتیجه نود (i, j) حتما وجود دارد.

۴.۲ جهش

جهش مناسب این الگوریتم مشابه روند بالا ایجاد فرزندان جهش یافته که حداقل به یکی از همسایگان متصل هستند را تضمین می کند.