

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

درس یادگیری ماشین
استاد ناظر فرد

تمرین چهارم

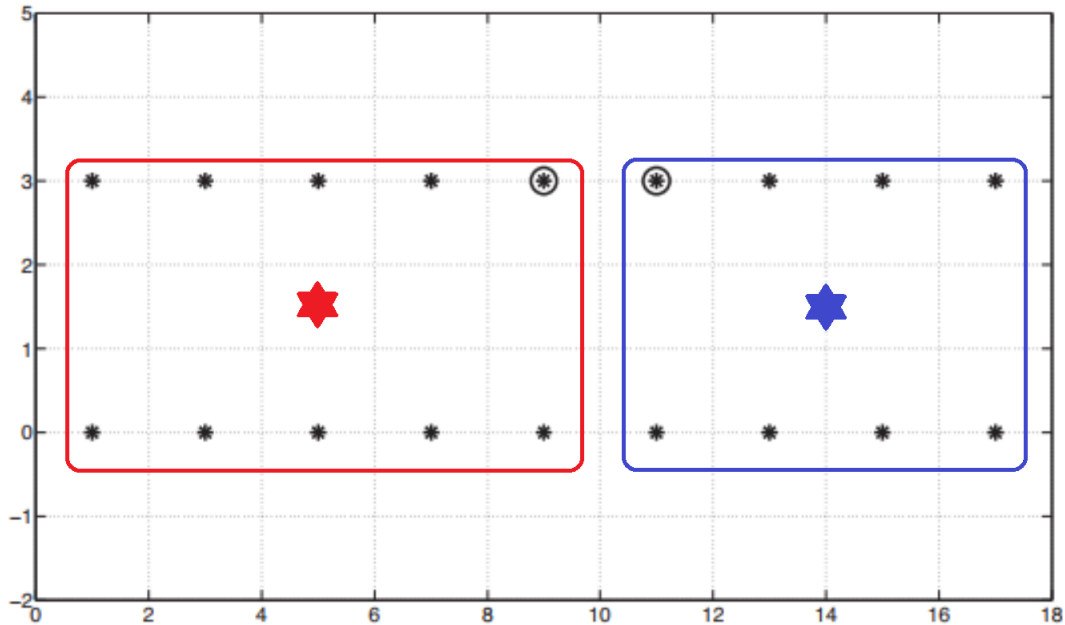
علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

بخش اول: پرسش‌های تشریحی

سوال ۱

دو خوشه آبی و قرمز به همراه مراکز که با ستاره نشان داده شده‌اند در تصویر زیر آورده شده است:



سوال ۲

الف) وقتی که یک الگوریتم خوشه‌بندی را بر روی داده‌های کاربران اعمال کنیم، کاربرانی در یک خوشه قرار خواهند گرفت که ویژگی‌های مشابه به یکدیگر داشته باشند. این همان چیزی است که در کاربرد Customer Segmentation مدنظر است.

ب) وقتی که خوشه‌بندی انجام می‌گیرد و مثلاً k خوشه داشته باشیم. اگر خوشه‌بندی سخت باشد، هر داده به یک خوشه تعلق دارد و به مابقی نه. پس می‌توان برای هر داده یک بردار k -بعدی در نظر گرفت که هر بعد متعلق به یک خوشه باشد. هر بعد مقدار یک بگیرد اگر داده به آن خوشه تعلق داشته باشد و در غیر این صورت صفر. اگر خوشه‌بندی نرم باشد، هم طبیعتاً برای هر بعد مقدار تعلق داده به آن خوشه نشان داده می‌شود. نهایتاً اینکه داده‌ها از فضای اولیه با ابعاد احتمالی بالا به ابعاد k کاهش بعد پیدا می‌کنند.

ج) در تعدادی از الگوریتم‌های خوشه‌بندی مانند DBSCAN داده‌های پرت به طور خودکار شناسایی می‌شوند و طبیعتاً می‌توان شناسایی کرد که یک داده، داده غیرعادی و پرت هست یا خیر. در سایر الگوریتم‌ها هم باز می‌توان با داشتن یک معیار فاصله محاسبه کرد که یک داده با هر خوشه چه میزان فاصله دارد؛ طبیعی است که اگر داده با تمام خوشه‌ها فاصله بالایی داشته باشد، یک داده عادی نخواهد بود.

د) مشابه با سوال پیاده‌سازی می‌توان پیکسل‌ها را خوشه‌بندی کرد و بدین ترتیب پیکسل‌های هم خوشه را در یک قطعه قرار دارد. جدای از بحث خوشه‌بندی پیکسل، در هر روشی باید قسمتی از عکس را در یک قطعه قرار داد که نوعی شباهت میان ویژگی‌های درون آن قطعه باشد و این چیزی است که خوشه‌بندی انجام می‌دهد.

سوال ۳

اگر حجم داده‌ها پایین باشد و امکان اجرای چندباره الگوریتم فراهم باشد، می‌توان با رنجی از مقادیر الگوریتم DBSCAN را اجرا کرد و یک سری از شروط را در آن چک کرد؛ مثلاً تعداد خوشه در یک بازه معقول و متناسب با کاربرد باشد و یا آنکه اندازه خوشه‌ها نسبت به یکدیگر از یک آستانه‌ای کمتر باشد. بدین ترتیب حالتی که شرایط را داشته باشد مورد پذیرش است. به طور مشابه می‌توان به جای شروط باینری، به هر وضعیت یک امتیاز متناسب با آن نسبت داد و مجموعه‌ای از پارامترها که بهترین امتیاز داشت را انتخاب کرد.

نهایتاً باید توجه داشت که با یک سری بررسی آماری روی داده‌ها نظیر میانگین تراکم می‌توان یک رنج معقول اولیه پیدا کرد.

سوال ۴

در حالت value iteration یک فرآیند تکراری طی می‌شود تا برای هر وضعیت مقدار value یا امتیاز آن محاسبه شود. این فرآیند وقتی متوقف می‌شود که value ها همگرا شوند. پس از اتمام و پیدا شدن این مقادیر، نوبت به پیدا کردن policy بهینه است.

policy بهینه بر اساس جدول مقادیر بهینه بدست می‌آید. لذا این فرآیند یک بار بیشتر انجام نمی‌شود. این درحالی است در policy iteration ابتدا یک policy اولیه در نظر گرفته و سپس به صورت تکرارشونده ابتدا بر اساس policy مقادیر وضعیت‌ها مشخص می‌شوند و سپس بر اساس مقادیر وضعیت، policy بهبود پیدا می‌کند. یعنی در این حالت در هر گام policy و value با هم بروز می‌شوند و زمانی که policy همگرا شود الگوریتم متوقف می‌شود.

سوال ۵

الف) هنگامی که قصد ادغام دو خوشه را در الگوریتم سلسله مراتبی را داشته باشیم بسته به معیار فاصله‌های متفاوتی برای خوشه‌ها حاصل می‌شود:

- در Complete Link بیشترین فاصله میان یک عضو از خوشه اول و یک عضو از خوشه دوم به عنوان فاصله دو خوشه در نظر گرفته می‌شود.
- در Single Link کمترین فاصله میان یک عضو از خوشه اول و یک عضو از خوشه دوم به عنوان فاصله دو خوشه در نظر گرفته می‌شود.
- در Average Link میانگین فاصله تمام جفت داده‌ها که یکی از خوشه اول و دیگری از خوشه دوم باشد محاسبه می‌شود.

از نظر پیچیدگی زمانی برای هر سه حالت لازم است تا فاصله تمام جفت داده‌ها محاسبه شود تا بتوان به ترتیب بیشینه، کمینه و میانگین آن را محاسبه کرد. لذا پیاده‌سازی کلاسیک این سه روش تفاوتی از منظر پیچیدگی زمانی با یکدیگر نخواهند داشت.

از نظر حساسیت به نویز، طبیعتاً Average Link با توجه به حالت میانگین‌گیری که دارد کمترین حساسیت را نسبت به داده‌های نویز دارد. بین دو روش دیگر به طور قطعی نمی‌توان نظر داد ولی می‌توان گفت Single Link حساسیت بیشتری به داده‌های نویز دارد^۱. وقتی که داده‌های نویز وجود داشته باشد، این داده‌ها در میان خوشه‌های واقعی

¹ <https://stats.stackexchange.com/q/304427/318893>

قرار می‌گیرند و این امکان را ایجاد می‌کنند که برخی از خوشه‌های واقعی در مراحل اولیه در یک خوشه قرار بگیرند، اما طبیعتاً Complete Link با این داده‌ها دچار مشکل نمی‌شود.

ب) در معیار Single Link دو خوشه ۱ و ۲ باهم خوشه می‌شوند و خوشه ۳ و ۴ باهم. چراکه در خوشه ۱ و ۲ داده‌هایی به هم خیلی نزدیک هستند. در معیار Complete Link دو خوشه ۱ و ۳ باهم خوشه می‌شوند و خوشه ۲ و ۴ باهم. چراکه یک چپ‌ترین داده خوشه ۱ از راست‌ترین داده خوشه ۲ فاصله زیادی دارد. در معیار Average Link هم خوشه ۱ و ۳ باهم خوشه می‌شوند و خوشه ۲ و ۴ باهم. چراکه به طور میانگین داده‌های ۱ به داده‌های ۳ نزدیک‌تر است تا ۲.

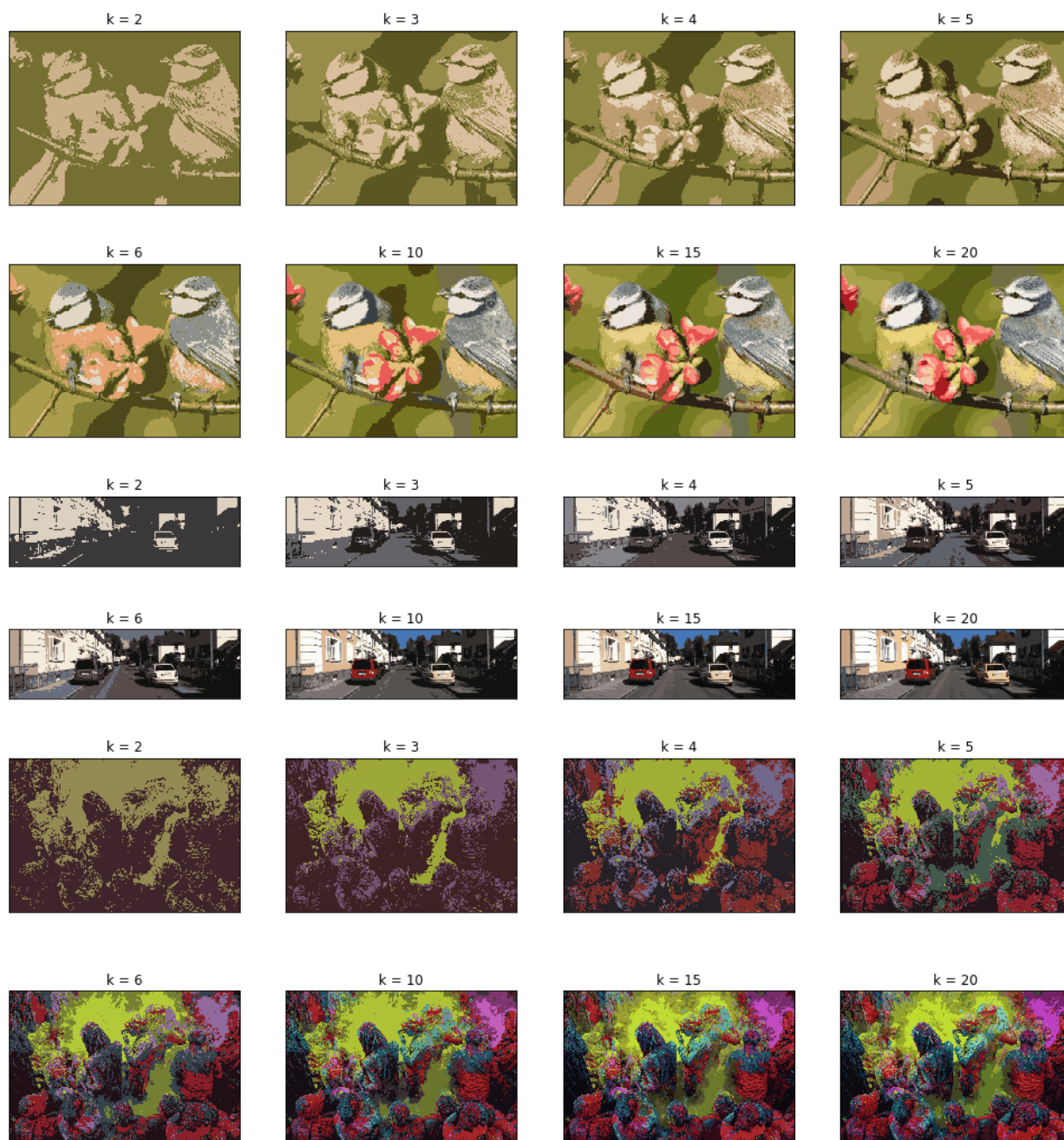
ج) برای مجموعه b معیار Single Link جواب می‌دهد چراکه موقع اجرای الگوریتم و در گام‌های اول که فاصله کم در نظر گرفته می‌شود تمام داده‌های یک خوشه به هم متصل می‌شوند چرا که در این معیار ملاک نزدیک‌ترین داده است و هر زیرخوشه از هر خوشه به زیرخوشه مجاور دیگری از آن خوشه دارای فاصله بسیار کمی است. اما طبیعی است که معیار Complete Link جواب ندهد. چراکه دو لبه‌ی هر خوشه از هم بسیار فاصله دارند و زمانی که قرار است تمام داده‌های یک خوشه به هم متصل شوند این مقدار فاصله برای دو زیرخوشه نهایی هر خوشه وجود خواهد داشت. در Average Link هم مشکل وجود خواهد داشت. چراکه داده‌های دو لبه‌ی هر خوشه میانگین فاصله را بالا می‌برند و این احتمال وجود دارد که لبه‌ی یک خوشه با داده‌های مرکزی خوشه دیگر زودتر تشکیل خوشه دهد.

برای مجموعه c اوضاع دو معیار Complete Link و Average Link متفاوت نخواهد. در این حالت روش Single Link هم به مشکل خواهد خورد. چراکه ممکن است لبه‌ی یک خوشه از طریق داده‌های جدید به مرکز یک خوشه دیگر متصل شود.

بخش دوم: پیاده‌سازی

سوال ۱

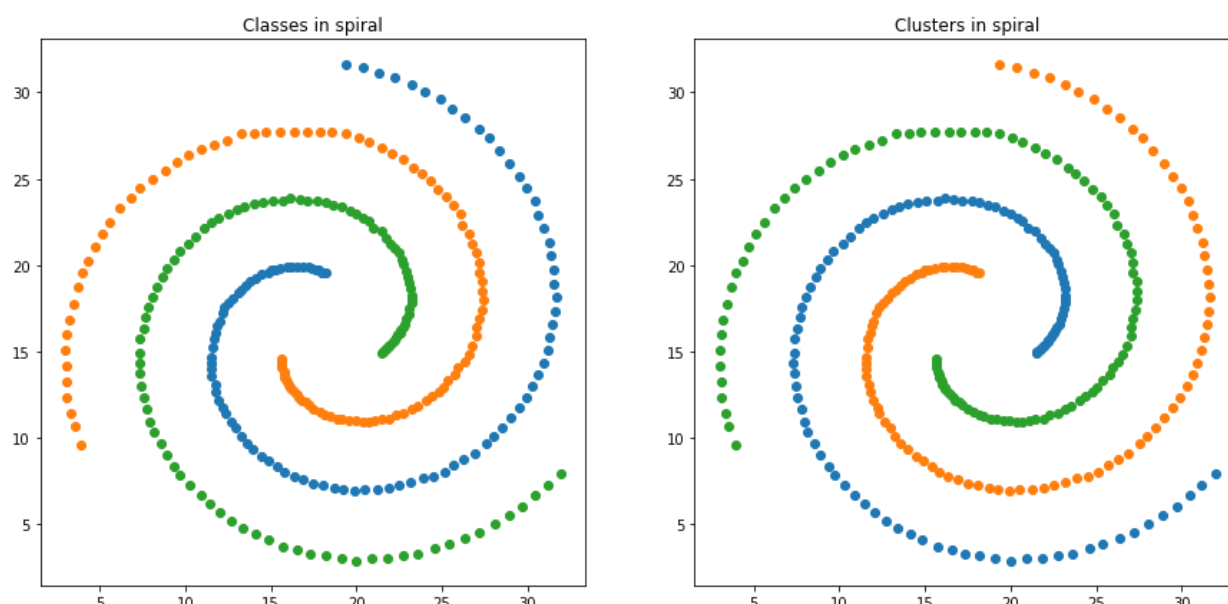
نتایج برای سه تصویر در ادامه آورده شده است:



سوال ۲

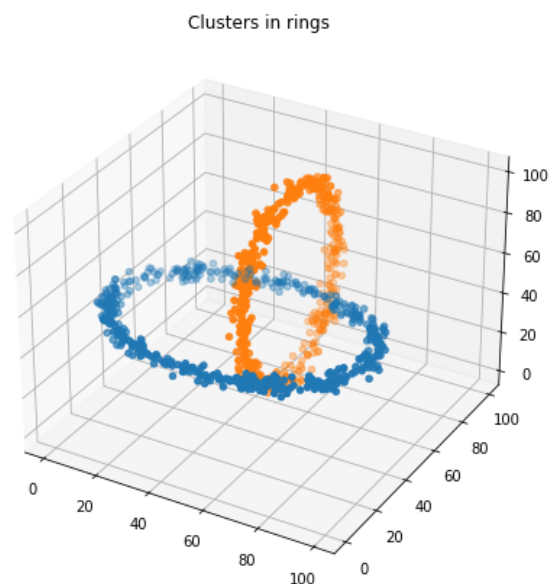
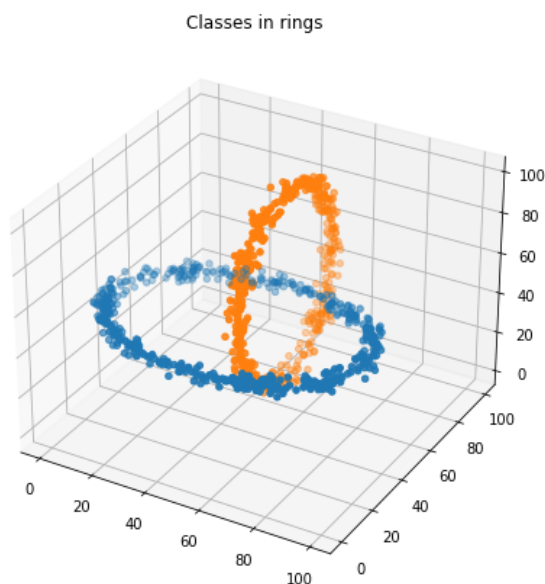
به غیر از مجموعه داده rings که دارای سه ویژگی است، مابقی مجموعه داده ها دارای دو ویژگی هستند. پس از پیاده سازی الگوریتم DBSCAN برای هر مجموعه داده یک مجموعه پارامتر مناسب که منجر به جواب قابل قبولی شود را بدست آوردم ولی باتوجه به آنکه در سوال تاکید بر یافتن جواب بهینه نبود این امکان وجود دارد که جوابی با خروجی بهتر وجود داشته باشد.

مجموعه داده Spiral



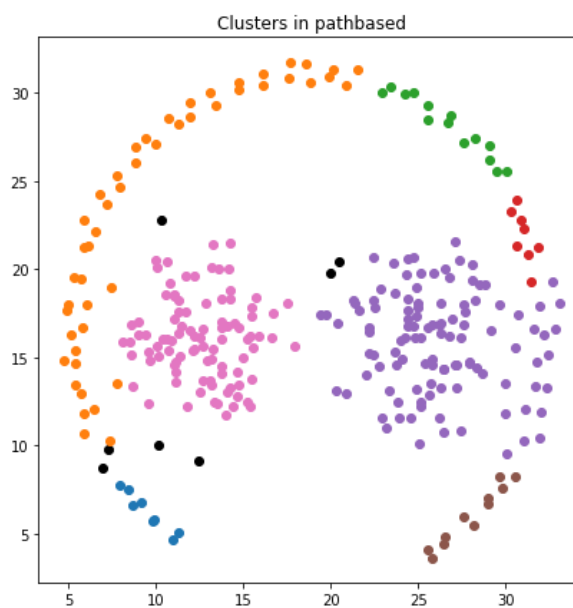
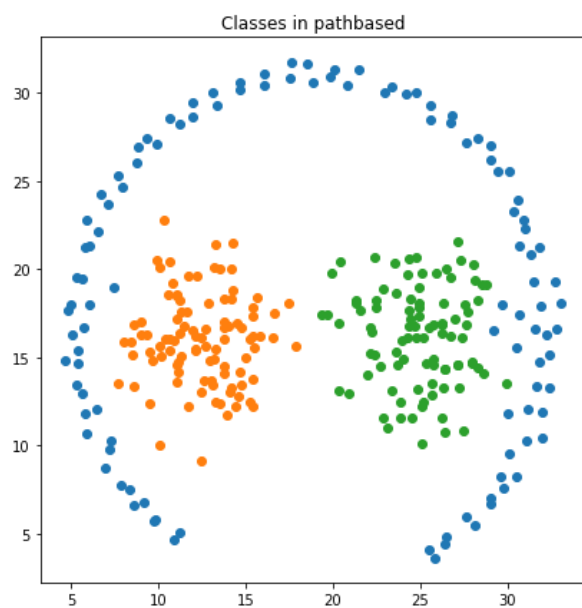
برای این مجموعه داده به خلوص ۱۰۰٪ می‌رسیم و در عین حال تعداد خوشه‌های برابر با تعداد خوشه واقعی و عدم تشخیص اشتباه داده پرت را هم داریم. در این شرایط خروجی به بهترین نحو با الگوریتم DBSCAN قابل حصول است. چنین خروجی با الگوریتم‌هایی مانند K-means قابل دستیابی نیستند. چراکه میانگین هر خوشه تقریباً در مرکز حلقه‌ها قرار می‌گیرد ولی هر خوشه در یک رشته با تراکم بالا و یکسان قرار گرفته است که مناسب یک الگوریتم برپایه چگالی مانند DBSCAN است.

مجموعه داده Rings



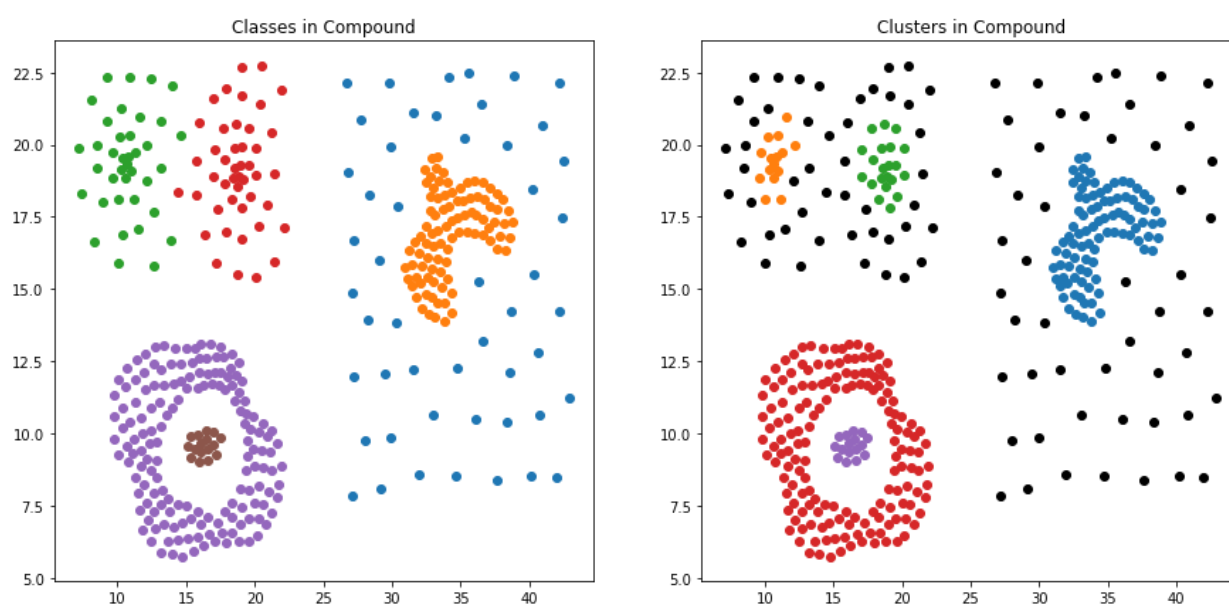
خروجی الگوریتم برای این مجموعه داده هم مانند قبل در بهترین حالت ممکن است.
از منظر تحلیلی هم مشابه هم هستند.

مجموعه داده Pathbased



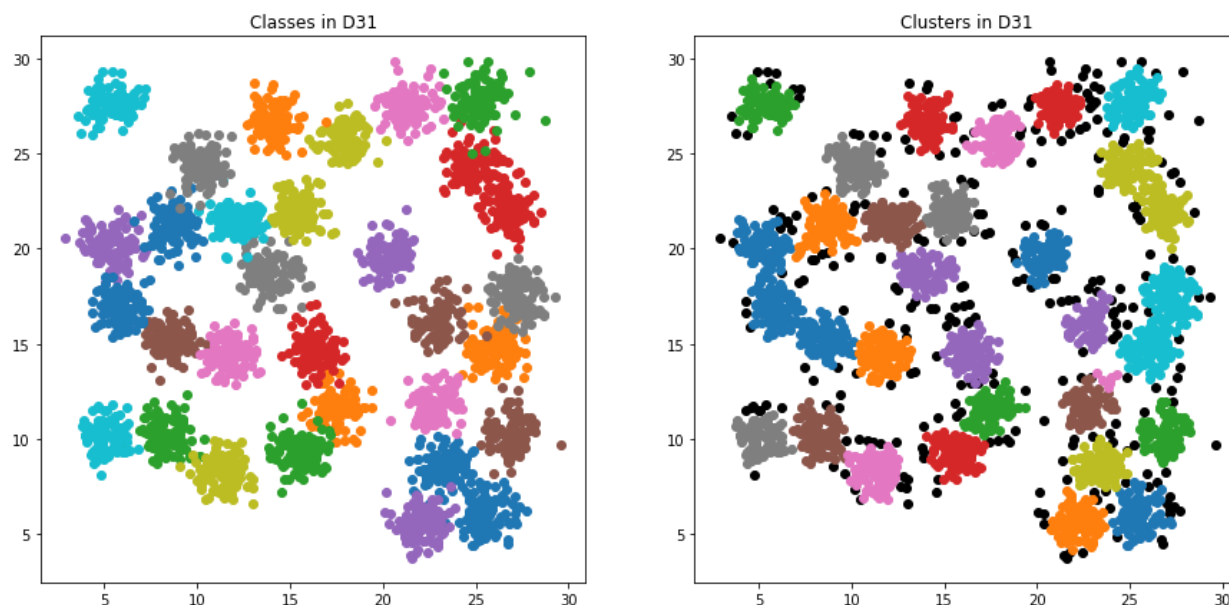
برای این مجموعه داده خلوص ۹۲/۸٪ بدست می‌آید که مناسب است. اما باید توجه داشت که برای این حالت تعداد خوشه‌ها به جای ۳ به عدد ۷ رسیده است و همچنین حدود ۲٪ داده‌ها نویز شناسایی شده‌اند. خوشه سبز رنگ تقریباً تراکم یکسانی با خوشه حلقوی دارد و طبیعتاً کار را برای الگوریتم دشوار کرده است. در عین حال خوشه حلقوی در چند قسمت بریدگی‌هایی دارد که باعث شده تا آن به چندین خوشه بشکند.

مجموعه داده Compound



برای این مجموعه داده خلوص ۱۰۰٪ است. اما نزدیک ۲۵ درصد داده‌ها به عنوان داده‌ی پرت شناسایی شده است. به طوری که یک خوشه کاملاً حذف شده است؛ ولی مابقی خوشه‌ها به طور مستقل و یکپارچه شناسایی شده‌اند. مشکلی که الگوریتم برای این مجموعه داده با آن مواجه است تراکم‌های مختلف خوشه‌ها است. یعنی خوشه قرمز و سبز تراکم کمی دارند ولی خوشه نارنجی و قهوه‌ای تراکم بالایی دارند. در این شرایط یا باید نرخ داده‌های پرت را زیاد کرد و یا آنکه احتمال تجمع چند خوشه با یکدیگر را پذیرفت که من حالت اول را انتخاب کرده‌ام. نهایتاً در مورد خوشه آبی چندان ظاهر یک خوشه را ندارد و انتظار زیادی است که یک الگوریتم خوشه‌بندی بتواند آن را مجزای از بقیه شناسایی کند.

مجموعه داده D31



خلوص این مجموعه داده برابر با $7/85\%$ ، نرخ داده‌های پرت برابر با $1/9\%$ است. تعداد خوشه‌های واقعی برابر با ۳۱ ولی تعداد خوشه‌های پیش‌بینی شده ۲۸ تاست. باتوجه به آنکه خوشه‌ها مرز مشترک با یکدیگر دارند، بدیهی است که یا باید نرخ داده‌های پرت را زیاد کرد و یا آنکه احتمال تجمع خوشه‌های واقعی در یک خوشه را پذیرفت که در اینجا من مجدداً مورد اول را انتخاب کردم. به نظر باتوجه به نحوه قرارگیری داده‌ها خروجی قابل قبول است. به علت ادغام ۳ خوشه مقدار خلوص پایین آمده است مگر نه این مولفه هم می‌توانست اعداد بسیار بالایی اتخاذ کند.

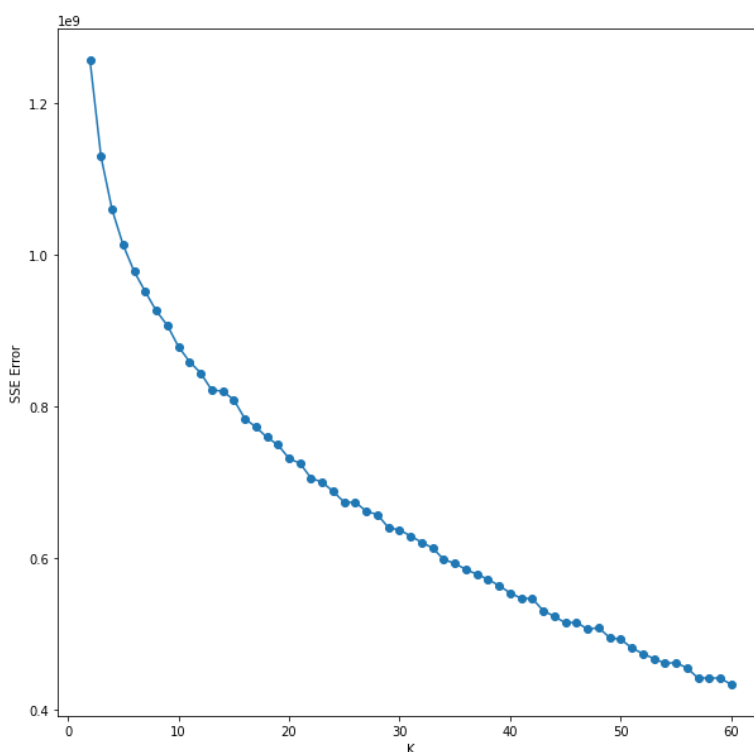
نهایتاً توجه کنید که برای این مجموعه داده الگوریتم k-means می‌توانست خروجی یکسان و یا بهتر (در صورت تعیین k و مراکز اولیه مناسب) را بدست بیاورد در صورتی که در سایر مجموعه داده‌ها چنین چیزی ممکن نیست.

سوال ۳

برای این سوال نیازی به مجموعه validation احساس نمی‌شود و داده‌ها به دو دسته آموزش و تست تقسیم می‌شود. از هر کلاس ۸ داده برای آموزش و ۲ داده برای تست انتخاب می‌شود. همچنین توجه کنید با توجه به پیاده‌سازی K-means در قسمت‌های قبل، برای این سوال از کتابخانه آماده استفاده کرده‌ام.

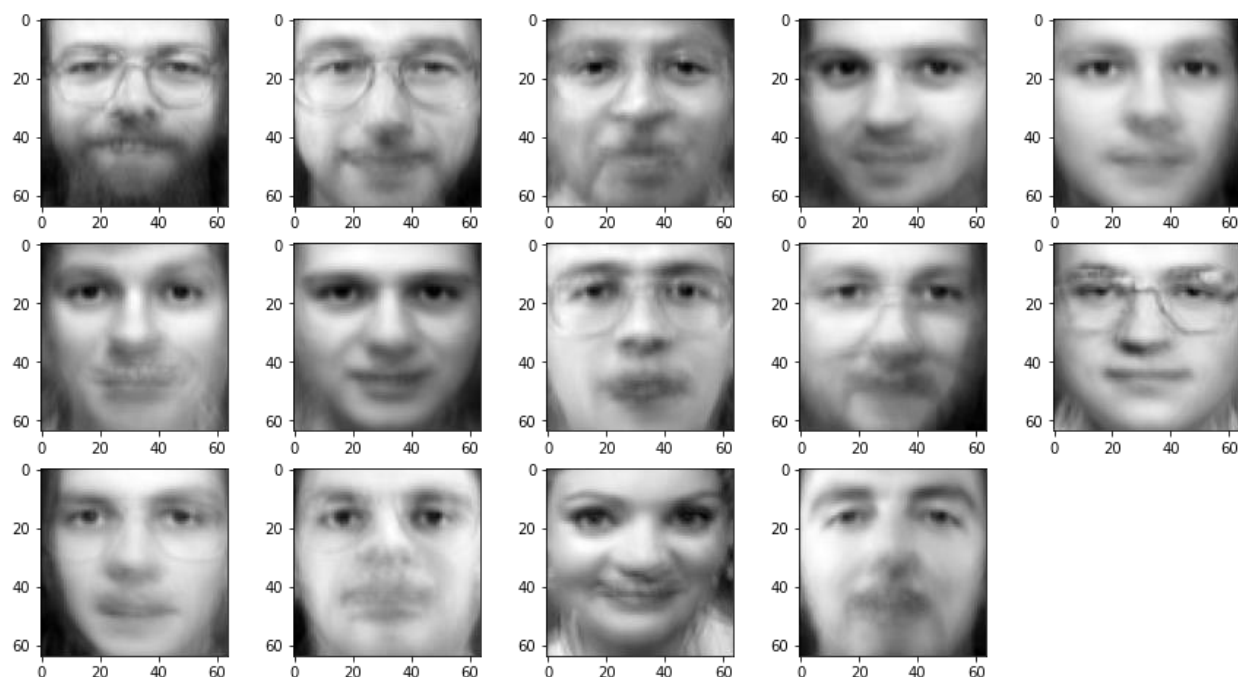
الف) در این نوع نمونه‌برداری به حفظ فراوانی گروه‌های مختلف اهمیت داده می‌شود و از هر دسته به تناسب فراوانی آن باید نمونه انتخاب شود. یعنی اگر به عنوان مثال مجموعه اولیه دارای ۱۰۰۰ آیتم در سه دسته ۵۰۰، ۳۰۰ و ۲۰۰ تایی باشد و قصد داشته باشیم تا یک نمونه ۱۰ تایی برداریم، باید ۵ تا از دسته اول ۳ تا از دسته دوم و ۲ تا از دسته سوم برداریم.

ب) نمودار elbow برای $k=2$ تا $k=61$ به شرح زیر است:

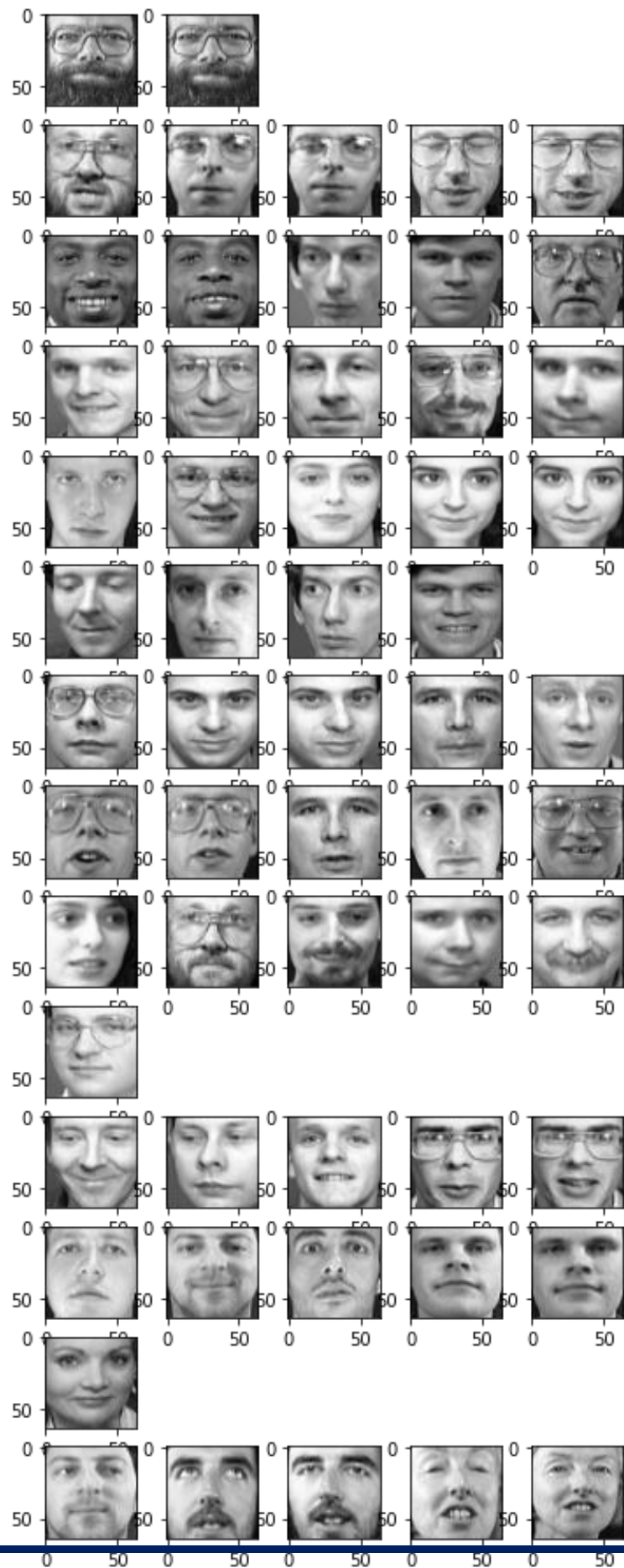


نمودار elbow چندان مشابه با نمودارهای elbow نرمال نشده است! به نظر $k=14$ مناسب می‌آید.

(ج) تصاویر مراکز چهارده خوشه به شرح زیر است:



برای هر خوشه حداکثر ۵ داده تست در تصویر زیر نمایش داده شده است. در این تصویر هر سطر مربوط به یک خوشه است. برای تعداد نسبتاً کمی از خوشه‌ها نظیر خوشه ۲ خروجی الگوریتم مطلوب است اما برای مابقی خوشه‌ها مانند خوشه ۳ یا ۴ نتایج چندان رضایت‌بخش نیست. البته باید توجه کرد که ما تنها ۱۴ خوشه در نظر گرفتیم و به ناچار برخی از چهره‌ها کنار هم می‌افتند. در برخی از موارد مانند آخرین خوشه با اینکه چهره‌ها متفاوت است ولی شباهت‌های منطقی‌ای وجود دارد.



سوال ۴

(۱) مقدار بهینه و سیاست بهینه به شرح زیر است:

6833	6755	5750	4869	4098	3479	###	###	766	765	750	707	501	316	178
6755	6701	5727	4865	4097	3479	###	###	1038	1030	1000	748	518	319	179
5750	5727	5628	4826	4090	3475	###	###	1382	1365	1053	771	522	321	179
4868	4865	4826	4701	4038	3437	###	###	1795	1412	1076	775	523	321	181
4094	4093	4086	4035	3900	3358	2840	2294	1827	1430	1078	776	524	324	185
3413	3413	3411	3400	3344	3205	2751	2302	1836	1431	1079	777	528	331	186
2815	2815	2814	2811	2797	2738	2601	2219	1830	1429	1079	783	539	332	186
2289	2289	2289	2288	2284	2268	2207	2077	1754	1419	1076	790	###	###	###
1828	1828	1828	1828	1827	1821	1803	1744	1622	1349	1062	785	###	###	###
1424	1424	1424	1424	1424	1423	1417	1398	1341	1229	999	763	533	327	183
1073	1073	1073	1074	1081	1090	1089	1072	1046	992	892	702	508	325	182
772	772	773	779	792	###	###	790	769	745	697	611	458	306	180
520	521	525	535	536	###	###	533	532	516	495	454	385	269	166
319	322	329	329	330	###	###	328	328	326	314	297	266	215	139
180	184	184	185	185	###	###	184	184	183	181	173	161	138	108

•	←	←	←	←	←	■	■	↓	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	■	■	↓	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	■	■	↗	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	↖	←	↗	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	↖	↖	←	←	←	←	↗	↗	↗
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	←	←
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	←	←
↗	↗	↗	↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↗	■	■	↖	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖

به نظر می‌رسد سیاست‌های اتخاذ شده مناسب هستند.

۲) مقدار بهینه و سیاست بهینه به شرح زیر است:

1467	1448	552	487	0	0	###	###	0	0	0	0	0	0	0
1412	1399	528	475	0	0	###	###	0	0	0	0	0	0	0
516	516	487	456	0	0	###	###	0	0	0	0	0	0	0
20	19	18	12	0	0	###	###	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	###	###	###
0	0	0	0	0	0	0	0	0	0	0	0	###	###	###
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	###	###	0	0	0	0	0	0	0
0	0	0	0	0	0	###	###	0	0	0	0	0	0	0
0	0	0	0	0	0	###	###	0	0	0	0	0	0	0
0	0	0	0	0	0	###	###	0	0	0	0	0	0	0

•	←	←	←	←	←	■	■	•	←	←	←	←	←	←
↑	↖	↖	↖	↖	↖	■	■	↑	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	↖	■	■	↑	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	↖	↖	↗	↑	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	←	←
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	■	■	↖	↖	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↑	↖	↖	↖	↖	■	■	↑	↖	↖	↖	↖	↖	↖	↖

برای این حالت همگرایی سیاست خیلی زود اتفاق می‌افتد و مقادیر وضعیت برای بسیاری از وضعیت‌ها برابر با صفر می‌شود. در این شرایط حرکت بالا چپ به عنوان پیش فرض (آیدی ۰) برای بسیاری از وضعیت‌ها انتخاب می‌شود و لذا به طور اتفاقی برای تعدادی زیادی از وضعیت‌ها پاسخ مطلوب حاصل می‌شود اما با این حال قسمت بالا سمت راست جدول مناسب نیست.

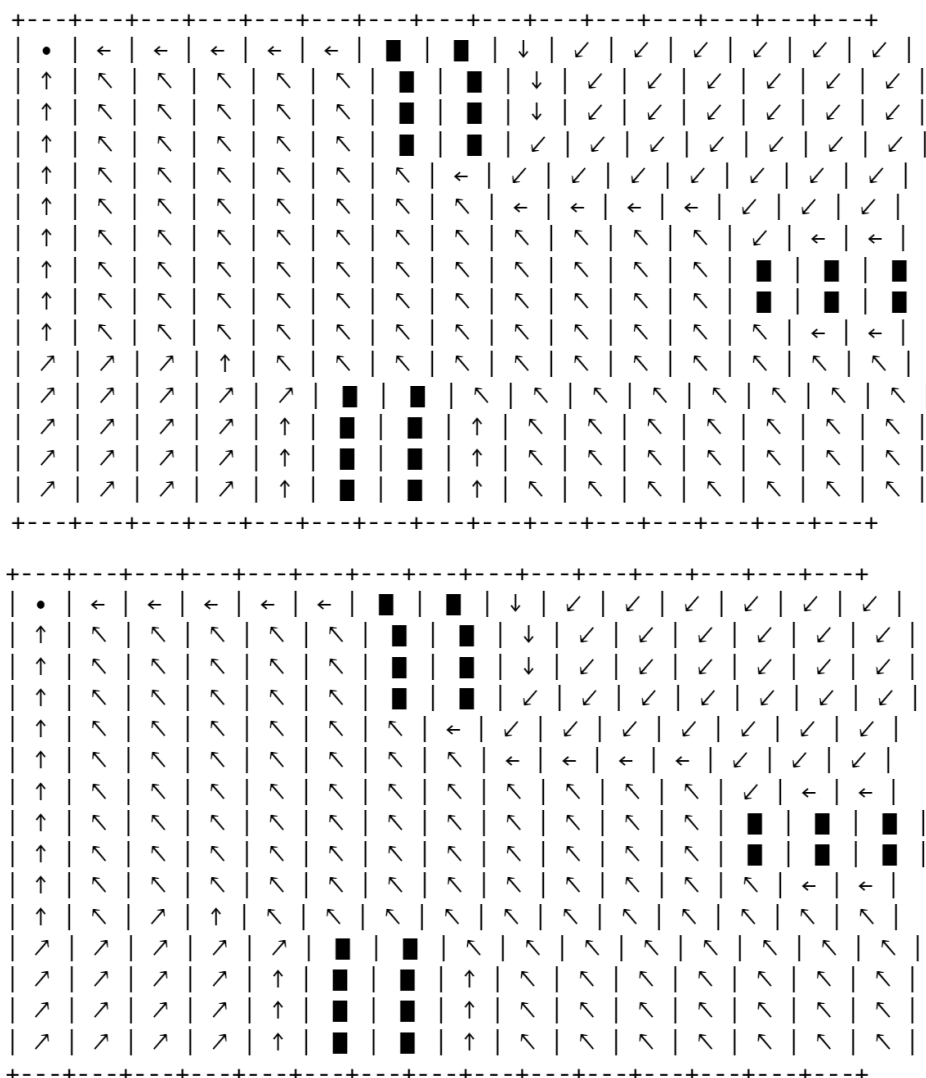
۳) مقدار بهینه و سیاسی بهینه به شرح زیر است:

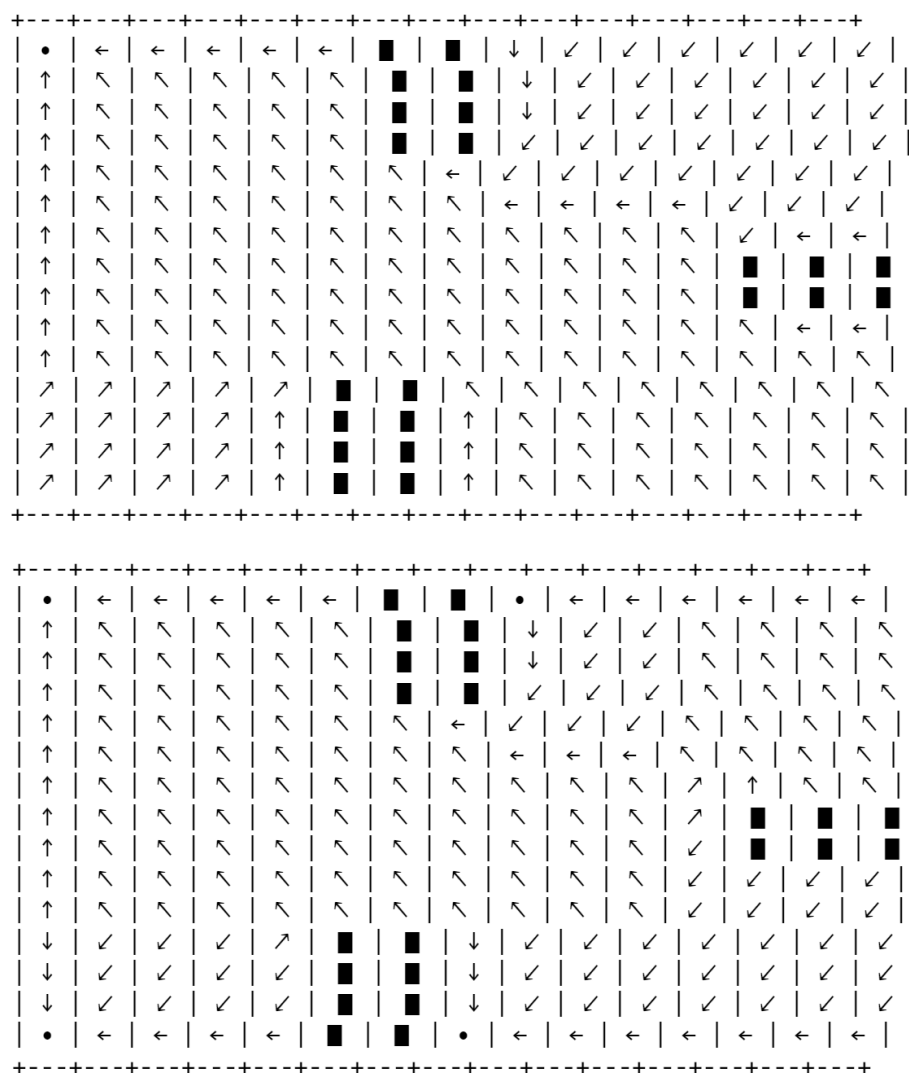
640	632	532	443	366	304	###	###	38	38	36	33	16	3	-3
632	627	529	443	366	304	###	###	63	62	59	36	17	3	-3
532	529	519	439	365	304	###	###	95	94	64	38	18	3	-3
443	443	439	426	360	300	###	###	136	98	66	39	18	3	-3
366	366	365	360	346	292	240	185	139	100	66	39	18	3	-3
297	297	297	296	290	277	231	186	140	100	66	39	18	4	-3
238	238	238	237	236	230	216	178	139	100	66	39	19	4	-3
185	185	185	185	184	183	177	164	132	99	66	40	###	###	###
139	139	139	139	139	138	137	131	119	92	65	39	###	###	###
99	99	99	99	99	99	99	97	91	81	59	37	18	4	-3
66	66	66	66	67	67	67	66	63	58	49	32	17	3	-3
38	38	38	39	40	###	###	40	38	36	32	25	13	3	-3
17	18	18	18	18	###	###	18	18	17	16	13	8	1	-4
3	3	4	4	4	###	###	4	4	4	3	2	1	-1	-4
-3	-3	-3	-3	-3	###	###	-3	-3	-3	-3	-3	-4	-4	-5

•	←	←	←	←	←	■	■	↓	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	■	■	↓	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	■	■	↗	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	↖	←	↗	↗	↗	↗	↗	↗	↗
↑	↖	↖	↖	↖	↖	↖	↖	←	←	←	←	↗	↗	↗
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	←	←
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	←	←
↑	↖	↗	↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↗	■	■	↖	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖

به نظر می‌رسد سیاست‌های اتخاذ شده مناسب هستند.

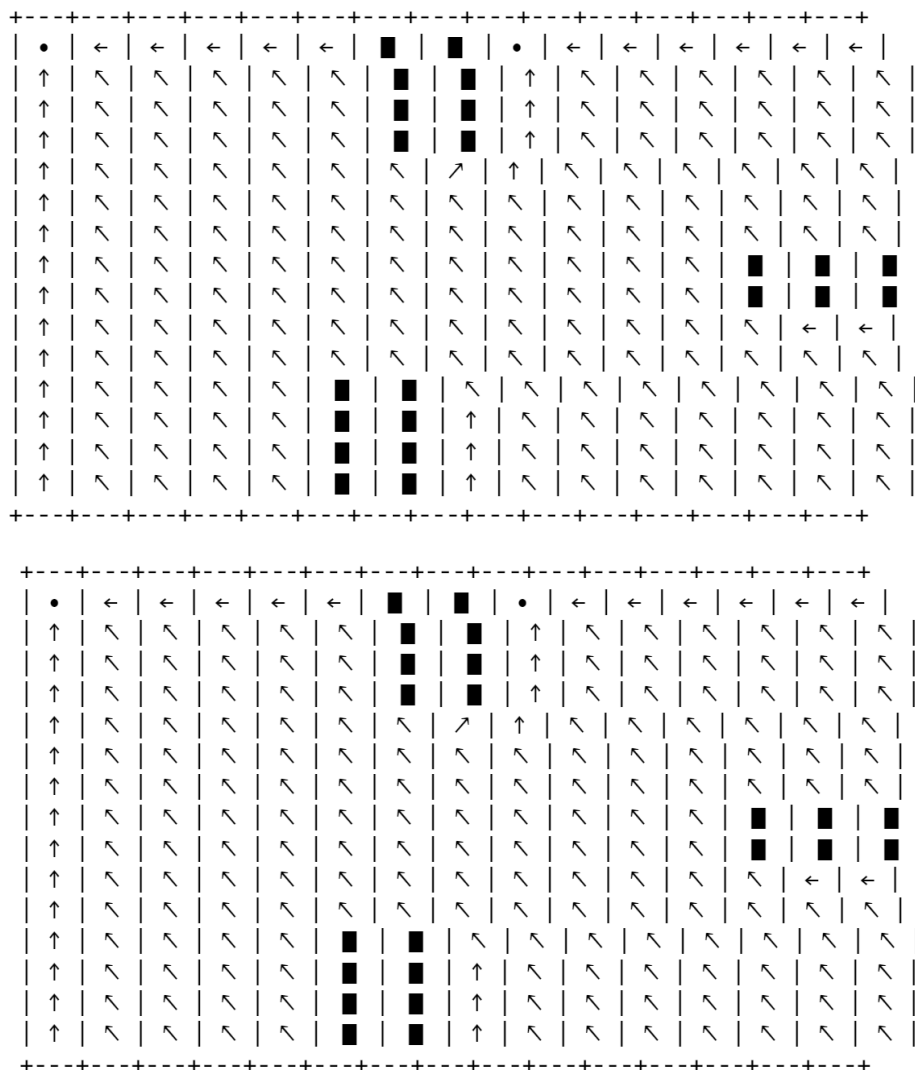
۴) برای تنظیمات قسمت ۱ به ترتیب چهار مقدار ۰/۹، ۰/۷۵، ۰/۵ و ۰/۱ را امتحان کردیم:

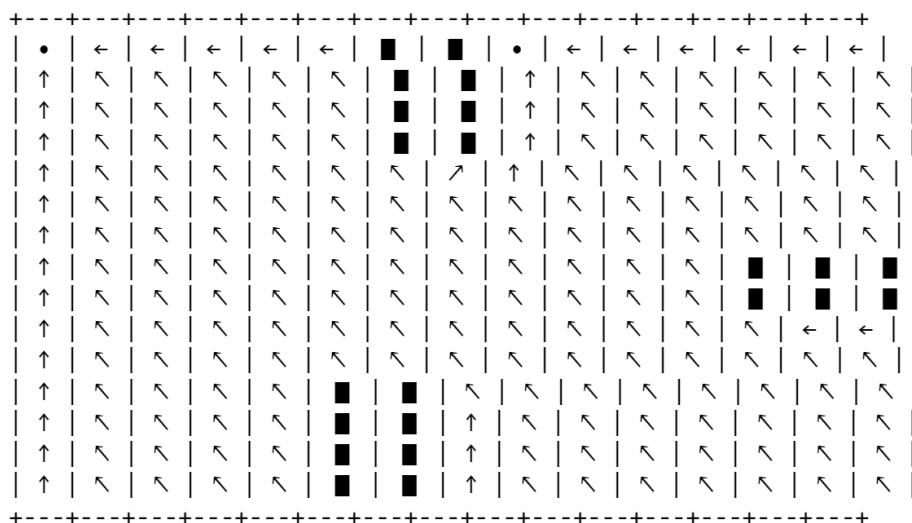
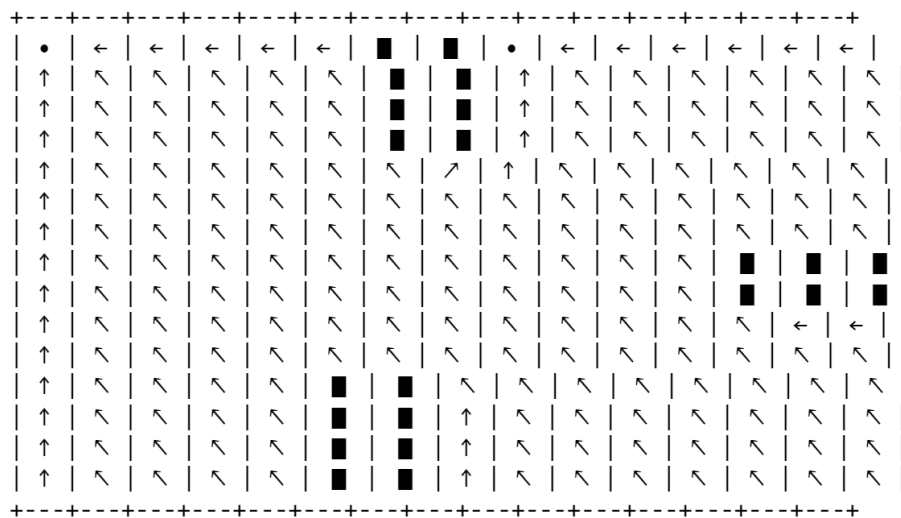
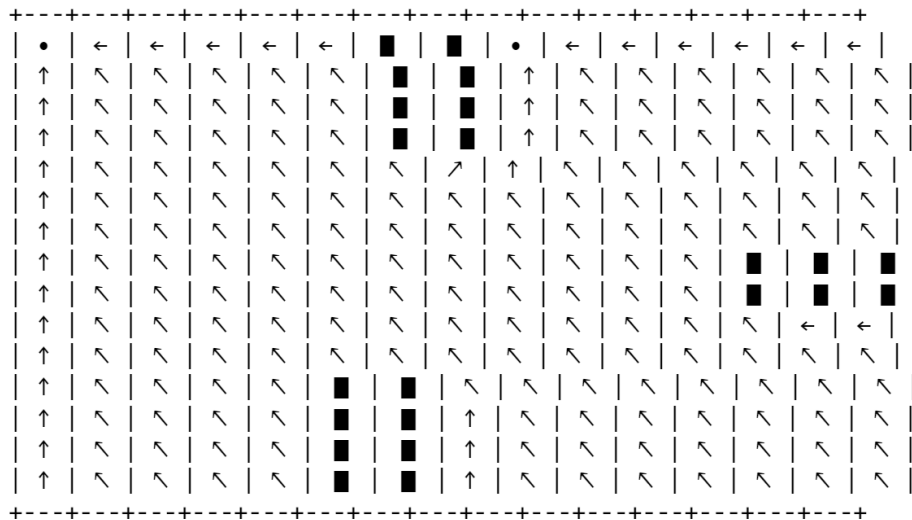




با بررسی حالت‌های مختلف در می‌یابیم که به جز برای مقدار $0/1$ برای سه حالت دیگر سیاست بهینه یکسان و در یک حالت مناسب قرار دارد. اما برای $0/1$ برای بخش بالا چپ جدول به نقطه نهایی می‌رسیم و برای قسمت‌های دیگر به وضعیت‌های نامطلوبی خواهیم رفت؛ این نشان می‌دهد که وقتی آینده‌نگری مدل شدیداً کم باشد، مدل نمی‌تواند به جواب مناسبی دست پیدا کند و ترجیح می‌دهد عامل را در نزدیکی جایی که هست نگه دارد.

حال چهار حالت discount factor قسمت قبل به علاوہی حالت ۰/۹۹ را برای تنظیم قسمت ۲ بدست می‌آوریم. نتایج سیاست بهینه به ترتیب برای ۰/۹۹، ۰/۹، ۰/۷۵، ۰/۵ و ۰/۱ به شرح زیر است:



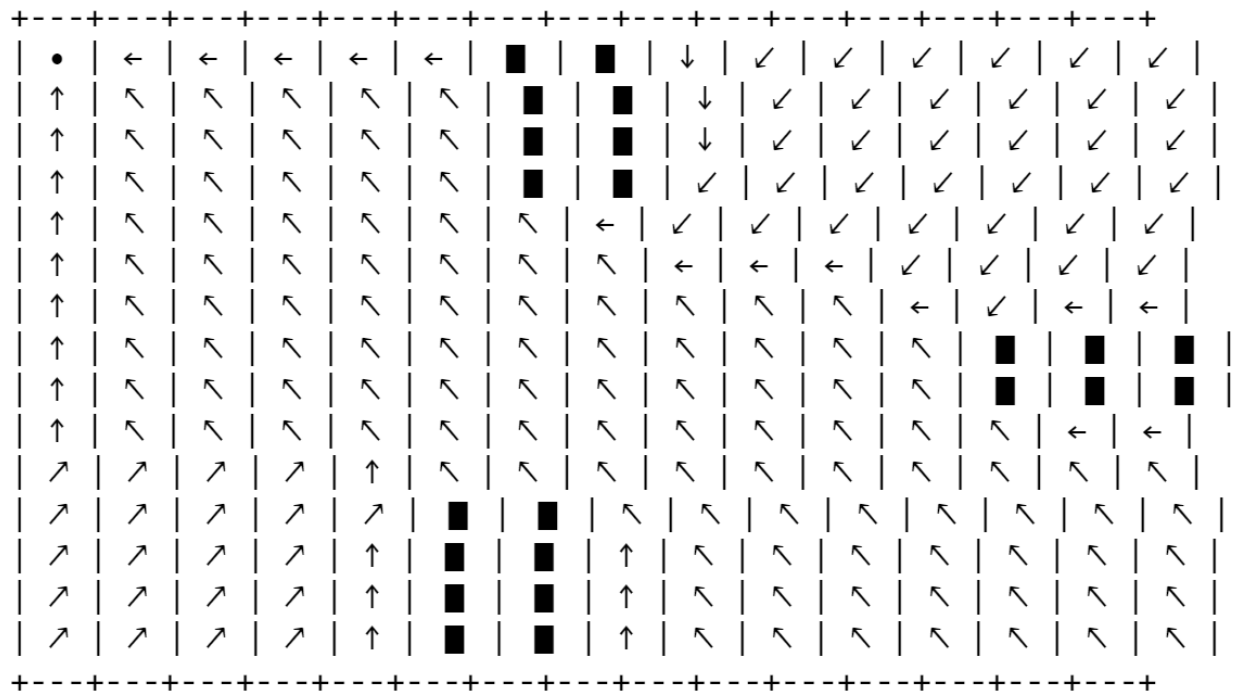


در تمامی این حالات، جدول مقادیر غالباً شامل مقدار صفر است و تنها چندین وضعیت بالای جدول مقدار دارد. با توجه به آنکه حرکت به سمت بالا چپ با آیدی ۰ به نوعی حرکت پیش فرض محسوب می‌شود، این باعث می‌شود تا در ابتدا جدول سیاست در یک وضعیت معقول قرار داشته باشد و همگرایی سیاست در چندین گام اول رخ دهد. اما همانطور که مشخص است برای خانه‌های بالا راست سیاست‌های پیشنهادی مناسب نیستند. تغییر discount factor برای حل این مشکل موثر نیست. شاید تغییر پیش‌فرض بتواند مشکل را حل کند و یا آنکه شاید بتوان پیاده‌سازی را به گونه‌ای تغییر داد که در چند گام اول امکان خاتمه الگوریتم به دلیل همگرایی سیاست وجود نداشته باشد.

(۵) با مقدار $\text{discount} = 0.9$ برای سه محیط پایه، بدون اصطکاک و با اصطکاک زیاد به ترتیب نتایج زیر حاصل شد.

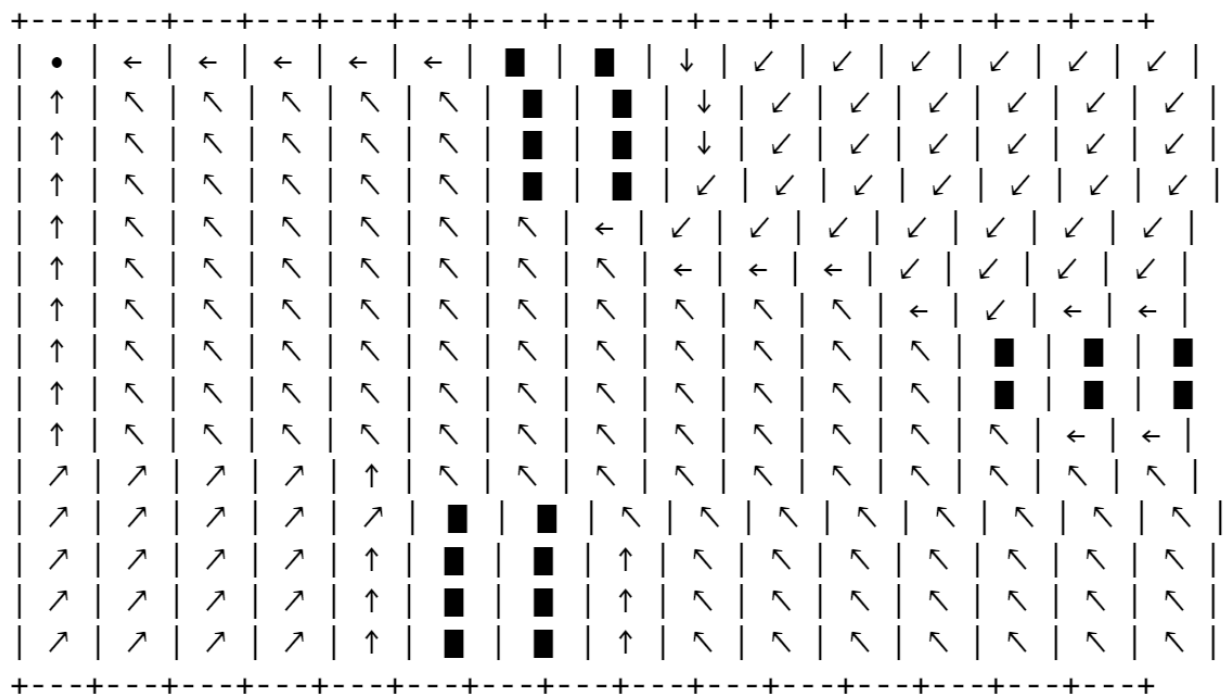
محیط پایه

8327	8249	7244	6362	5592	4972	###	###	2236	2235	2218	2169	1943	1721	1531
8249	8194	7220	6359	5591	4972	###	###	2519	2511	2478	2215	1963	1726	1533
7244	7220	7122	6319	5583	4968	###	###	2870	2852	2535	2241	1968	1728	1535
6362	6359	6319	6195	5531	4931	###	###	3286	2901	2558	2246	1970	1729	1539
5587	5586	5579	5529	5393	4852	4334	3787	3318	2919	2561	2247	1971	1734	1547
4907	4906	4904	4893	4837	4698	4244	3794	3328	2920	2562	2248	1977	1745	1549
4308	4308	4308	4304	4290	4231	4094	3711	3322	2918	2562	2255	1991	1746	1549
3782	3782	3782	3781	3776	3760	3700	3569	3245	2908	2559	2263	###	###	###
3320	3320	3320	3319	3318	3313	3295	3235	3112	2837	2545	2258	###	###	###
2913	2913	2913	2913	2913	2912	2906	2886	2828	2714	2479	2233	1983	1738	1542
2556	2556	2556	2557	2565	2574	2572	2555	2528	2472	2367	2167	1953	1736	1541
2242	2242	2243	2250	2265	###	###	2262	2239	2213	2160	2065	1893	1708	1536
1966	1967	1973	1985	1986	###	###	1983	1982	1962	1936	1887	1801	1654	1509
1726	1731	1740	1741	1742	###	###	1739	1739	1736	1719	1695	1649	1572	1458
1536	1543	1545	1545	1545	###	###	1543	1543	1542	1538	1522	1497	1453	1392



محیط بدون اصطکاک

8327	8249	7244	6362	5592	4972	###	###	2237	2235	2218	2170	1943	1721	1531
8249	8195	7221	6359	5591	4972	###	###	2519	2511	2478	2216	1963	1726	1533
7244	7221	7122	6319	5583	4968	###	###	2870	2852	2535	2241	1968	1728	1535
6362	6359	6319	6195	5532	4931	###	###	3286	2901	2559	2246	1970	1730	1539
5587	5586	5579	5529	5393	4852	4334	3787	3318	2919	2561	2247	1971	1735	1547
4907	4906	4904	4894	4837	4698	4244	3795	3328	2920	2562	2248	1977	1745	1549
4308	4308	4308	4304	4290	4231	4094	3711	3322	2918	2562	2255	1991	1746	1549
3782	3782	3782	3781	3777	3760	3700	3569	3245	2908	2559	2263	###	###	###
3320	3320	3320	3319	3318	3313	3295	3235	3112	2837	2545	2258	###	###	###
2913	2913	2913	2913	2913	2912	2906	2886	2828	2714	2479	2233	1983	1738	1542
2556	2556	2556	2557	2565	2574	2572	2555	2528	2472	2367	2167	1953	1736	1541
2242	2242	2243	2250	2265	###	###	2262	2239	2213	2160	2065	1893	1708	1536
1966	1967	1973	1985	1986	###	###	1983	1982	1962	1936	1887	1801	1654	1509
1726	1731	1740	1741	1742	###	###	1739	1739	1736	1719	1695	1649	1572	1458
1536	1543	1545	1545	1546	###	###	1543	1543	1542	1538	1522	1497	1454	1392



محیط با اصطکای زیاد

803	795	694	606	529	466	###	###	192	192	191	186	163	141	122
795	789	692	605	528	466	###	###	221	220	217	190	165	141	122
694	692	682	601	528	466	###	###	256	254	222	193	166	141	122
606	605	601	589	522	462	###	###	298	259	225	193	166	142	123
528	528	527	522	509	454	402	348	301	261	225	193	166	142	123
460	460	460	459	453	439	393	348	302	261	225	194	166	143	123
400	400	400	400	398	392	379	340	301	261	225	194	168	143	124
347	347	347	347	347	345	339	326	293	260	225	195	###	###	###
301	301	301	301	301	300	298	292	280	253	223	195	###	###	###
260	260	260	260	260	260	259	257	252	240	217	192	167	142	123
224	224	224	224	225	226	226	224	222	216	205	185	164	142	123
193	193	193	194	195	###	###	195	193	190	185	175	158	139	122
165	165	166	167	167	###	###	167	167	165	162	157	149	134	119
141	142	143	143	143	###	###	143	143	142	141	138	134	126	114
122	123	123	123	123	###	###	123	123	123	122	121	118	114	108

•	←	←	←	←	←	■	■	↓	↙	↙	↙	↙	↙	↙
↑	↖	↖	↖	↖	↖	■	■	↓	↙	↙	↙	↙	↙	↙
↑	↖	↖	↖	↖	↖	■	■	↙	↙	↙	↙	↙	↙	↙
↑	↖	↖	↖	↖	↖	↖	←	↙	↙	↙	↙	↙	↙	↙
↑	↖	↖	↖	↖	↖	↖	↖	←	←	←	↙	↙	↙	↙
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	←	↙	←	←
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	■	■	■
↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖	←	←
↗	↗	↗	↗	↑	↖	↖	↖	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↗	■	■	↖	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖
↗	↗	↗	↗	↑	■	■	↑	↖	↖	↖	↖	↖	↖	↖

در این حالت اجرا، برای هر سه محیط به یک جواب مناسب می‌رسیم این در حالی است که برای محیط بدون اصطکاک در حالت اجرای policy iteration امکان دستیابی به جواب مناسب وجود نداشت. در الگوریتم value iteration همگرایی دیرتر رخ می‌دهد و این باعث می‌شود که زمان اجرا نسبت به policy iteration خیلی بیشتر باشد؛ اما از طرفی وقتی همگرایی رخ دهد با اطمینان بیشتری می‌توان الگوریتم را خاتمه داد و امکان دستیابی به جواب با کیفیت‌تر بیشتر است.