دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

یادگیری ماشین

پاییز ۱۴۰۳

استاد: علی شریفی زارچی
مسئول تمرین: نیکی سپاسیان

مهلت ارسال امتیازی: ۱۱ آبان | **تمرین دوم** | مهلت ارسال نهایی: ۱۸ آبان

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روزهای مشخص‌شده است.
- در طول ترم، برای هر تمرین می‌توانید تا ۵ روز تأخیر مجاز داشته باشید و در مجموع حداکثر ۱۵ روز تأخیر مجاز خواهید داشت. توجه داشته باشید که تأخیر در تمرین‌های عملی و تئوری به صورت جداگانه محاسبه می‌شود و مجموع تأخیر هر دو نباید بیشتر از ۱۵ روز شود. پس از اتمام زمان مجاز، دو روز اضافی برای آپلود غیرمجاز در نظر گرفته شده است که در این بازه به ازای هر ساعت تأخیر، ۲ درصد از نمره تمرین کسر خواهد شد.
- اگر بخش عملی یا تئوری تمرین را قبل از مهلت ارسال امتیازی آپلود کنید، ۲۰ درصد نمره اضافی به آن بخش تعلق خواهد گرفت و پس از آن، ویدئویی تحت عنوان راهنمایی برای حل تمرین منتشر خواهد شد.
- حتماً تمرین‌ها را بر اساس موارد ذکرشده در صورت سوالات حل کنید. در صورت وجود هرگونه ابهام، آن را در صفحه تمرین در سایت کوئرا مطرح کنید و به پاسخ‌هایی که از سوی دستیار آموزشی مربوطه ارائه می‌شود، توجه کنید.
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- فایل پاسخ‌های سوالات نظری را در قالب یک فایل pdf به فرمت $HW۲\_T\_[STD\_ID].pdf$ آماده کنید و برای سوالات عملی، هریک را در یک فایل zip جداگانه قرار دهید و فایل zip اول را به فرمت $HW۲\_P۱\_[STD\_ID].zip$ و فایل zip دوم را به فرمت $HW۲\_P۲\_[STD\_ID].zip$ نامگذاری کرده و هرکدام را به صورت جداگانه آپلود کنید.
- گردآورندگان تمرین: محمدپارسا بشری، عرفان جعفری، مهدی طباطبایی، فاطمه السادات موسوی، محمد مولوی

سوالات نظری (۱۰۰ نمره)

۱. (۲۰ نمره) در یک مسئله‌ی طبقه‌بندی دو کلاسه با دو ویژگی، از هر کلاس دو داده داریم:

$$\omega_۱ : \begin{pmatrix}۴\\۱\end{pmatrix}, \begin{pmatrix}۱\\۰\end{pmatrix}, \omega_۲ : \begin{pmatrix}۲\\۵\end{pmatrix}, \begin{pmatrix}۳\\۴\end{pmatrix}$$

الف) با محاسبه‌ی بردار میانگین و ماتریس کوواریانس، داده‌ها را با استفاده از PCA به فضای یک بعدی برده و تمایزپذیری کلاس‌ها را بررسی کنید. برای هر دو راستا مسئله را حل کنید.

ب) در قسمت الف یک بردار پیشنهاد دهید که اگر به همه‌ی داده‌ها اضافه شود مولفه‌ی اساسی اول تغییر نکند.

حل.

الف)

برای محاسبه بردار میانگین باید میانگین تمام داده‌ها را مستقل از کلاس آن‌ها حساب کرد:

$$\bar{x} = \frac{۱}{۴}\left(\begin{pmatrix}۴\\۱\end{pmatrix} + \begin{pmatrix}۱\\۰\end{pmatrix} + \begin{pmatrix}۲\\۵\end{pmatrix} + \begin{pmatrix}۳\\۴\end{pmatrix}\right) = \begin{pmatrix}۲٫۵\\۲٫۵\end{pmatrix}$$

سپس باید ماتریس کوواریانس را حساب کنیم:

$$\tilde{X} = \begin{pmatrix} ۴ - ۲/۵ & ۱ - ۲/۵ \\ ۱ - ۲/۵ & ۰ - ۲/۵ \\ ۲ - ۲/۵ & ۵ - ۲/۵ \\ ۳ - ۲/۵ & ۴ - ۲/۵ \end{pmatrix} = \begin{pmatrix} ۱/۵ & -۱/۵ \\ -۱/۵ & -۲/۵ \\ -۱/۵ & ۲/۵ \\ ۰/۵ & ۱/۵ \end{pmatrix}$$

$$S = \frac{۱}{N}\tilde{X}^T\tilde{X} = \begin{pmatrix} ۱/۶۶ & ۰/۳۳ \\ ۰/۳۳ & ۵/۶۶ \end{pmatrix}$$

حال باید بردارهای ویژه و مقادیر ویژه متناظر با ماتریس کواریانس را پیدا کنیم:

$$Sv = \lambda v$$

$$\lambda_۱ = ۵/۷, \lambda_۲ = ۱/۶۳$$
$$v_۱ = \begin{pmatrix} -۰/۰۸ \\ -۰/۹۹ \end{pmatrix}, v_۲ = \begin{pmatrix} ۰/۹۹ \\ -۰/۰۸ \end{pmatrix}$$

حال داده‌ها را به راستای اول project می‌کنیم:

$$X' = XA = Xv_۱$$

$$X' = \begin{pmatrix} ۴ & ۱ \\ ۱ & ۰ \\ ۲ & ۵ \\ ۳ & ۴ \end{pmatrix} \begin{pmatrix} -۰/۰۸ \\ -۰/۹۹ \end{pmatrix} = \begin{pmatrix} -۱/۳۱ \\ -۰/۰۸ \\ -۵/۱۱ \\ -۴/۲۱ \end{pmatrix}$$

همانگونه که مشخص است، دو کلاس تمایزپذیر هستند. حال همین کار را برای راستای دوم انجام می‌دهیم:

$$X' = XA = Xv_۲$$

$$X' = \begin{pmatrix} ۴ & ۱ \\ ۱ & ۰ \\ ۲ & ۵ \\ ۳ & ۴ \end{pmatrix} \begin{pmatrix} ۰/۹۹ \\ -۰/۰۸ \end{pmatrix} = \begin{pmatrix} ۳/۸۸ \\ ۰/۹۹ \\ ۱/۵۸ \\ ۲/۶۵ \end{pmatrix}$$

همانگونه که می‌بینید، دو کلاس تمایزپذیر نیستند.

ب) نشان می‌دهیم که اضافه شدن هر بردار دلخواه a از فضای ویژگی‌ها به تمام داده‌ها مقدار مولفه اساسی اول را ثابت نگه می‌دارد.

$$\bar{X}_new = \frac{۱}{n}\sum(x_i + a)$$
$$= \frac{۱}{n}(\sum x_i + na)$$
$$= \bar{X} + a$$

بنابراین داریم:

$$Cov(X_new) = \frac{۱}{n-۱}\sum(x_i + a - \bar{X}_new)(x_i + a - \bar{X}_new)^T$$

$$= \frac{1}{n-1}\sum (x_i + a - \bar{X} - a)(x_i + a - \bar{X} - a)^T$$

$$= \frac{1}{n-1}\sum (x_i - \bar{X})(x_i - \bar{X})^T$$

$$= Cov(x)$$

ثابت ∑

بنابراین بردارهای ویژه تغییر نخواهند کرد و مقدار مولفه اساسی اول تغییر نخواهد کرد.

۲. (۲۰ نمره) قصد داریم مدلی بسازیم که با گرفتن ورودی X، متغیر خروجی Y را تخمین بزند. فرض کنید M مدل ضعیف به نام‌های $f_1(X), ..., f_M(X)$ داریم که روی نمونه‌های bootstrap شده‌ای از دیتاست اصلی آموزش داده شده‌اند و دارای بایاس و واریانس یکسان هستند. مدل نهایی به شکل زیر تعریف می‌شود:

$$f_{\text{ensemble}}(X) = \frac{1}{M}\sum_{i=1}^{M} f_i(X)$$

الف) بایاس و واریانس مدل $f_{\text{ensemble}}(X)$ را بر حسب بایاس و واریانس مدل‌های ضعیف محاسبه کنید. در این قسمت فرض کنید مدل‌های ضعیف از یکدیگر مستقل هستند. تحلیل کنید که افزایش تعداد مدل‌های ضعیف (M) چه تاثیری روی بایاس و واریانس مدل نهایی دارد.

ب) حال فرض کنید مدل‌های ضعیف مستقل نیستند و correlation بین هر دو مدل $f_i(X)$ و $f_j(X)$ $(i \neq j)$ برابر ρ است. حال مانند قسمت الف، بایاس و واریانس مدل نهایی را محاسبه کرده و تاثیر تعداد مدل‌ها (M) و وابستگی بین آن‌ها (ρ) را روی این مقادیر بررسی کنید.

پ) به سوالات زیر به طور خلاصه پاسخ دهید.

- آیا یادگیرنده‌های ضعیف در adaboost نیاز به مشتق‌پذیر بودن دارند؟ چرا؟
- بین boosting و bagging ، کدام یک از نظر محاسباتی گران‌تر می‌باشد؟ چرا؟

حل.

الف) بایاس مدل جدید به شکل زیر محاسبه می‌شود:

$$\text{Bias}(f_{\text{ensembe}}(X)) = \frac{1}{M}\sum_{i=1}^{M}\text{Bias}(f_i(X)) = \text{Bias}(f_i(X))$$

واریانس مدل جدید نیز با توجه به مستقل بودن مدل‌های ضعیف به شکل زیر محاسبه می‌شود:

$$\text{Var}(f_{\text{ensembe}}(X)) = \frac{1}{M^2}\sum_{i=1}^{M}\text{Var}(f_i(X)) = \frac{1}{M}\text{Var}(f_i(X))$$

بنابراین می‌توان دید که با افزایش تعداد مدل‌ها (M) واریانس مدل نهایی کاهش می‌یابد در حالیکه بایاس ثابت می‌ماند. برای مدل‌های ضعیف که معمولا واریانس بالا و بایاس پایین دارند، این کار منجر به کاهش چشمگیری در خطای مدل نهایی می‌شود.

ب) بایاس مدل نهایی در حالت مستقل نبودن مدل‌های ضعیف فرقی با قسمت قبل ندارد. واریانس مدل جدید

به شکل زیر محاسبه خواهد شد:

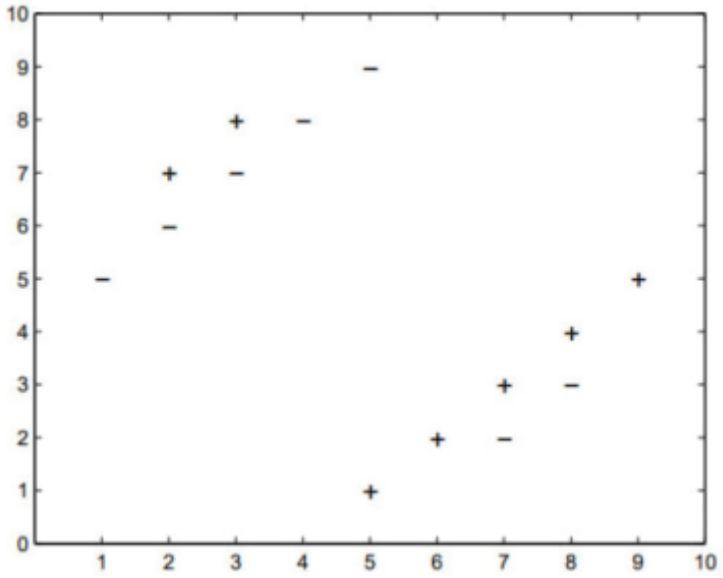
$$\begin{aligned}
\mathrm{Var}(f_{\text{ensembe}}(X)) &= \mathrm{Var}(\frac{1}{M}\sum_{i=1}^{M} f_i(X)) \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\mathrm{Cov}(f_i(X), f_j(X)) \\
&= \frac{1}{M^2}\sum_{i=j}\mathrm{Cov}(f_i(X), f_i(X)) + \frac{1}{M^2}\sum_{i\neq j}\mathrm{Cov}(f_i(X), f_j(X)) \\
&= \frac{1}{M^2}\sum_{i=j}\mathrm{Var}(f_i(X)) + \frac{1}{M^2}\sum_{i\neq j}\rho\mathrm{Var}(f_i(X)) \\
&= \frac{1}{M}\mathrm{Var}(f_i(X)) + \frac{M-1}{M}\rho\mathrm{Var}(f_i(X))
\end{aligned}$$

بنابراین می‌توان دید که با افزایش ρ تاثیر تعداد مدل‌های ضعیف (M) کمتر می‌شود. به عبارتی اضافه کردن تعداد بیشتری مدل ضعیف، واریانس مدل را به قدر کافی کاهش نمی‌دهد؛ زیرا به دلیل وابستگی زیاد بین این مدل‌ها، در واقع مدل‌های اضافه شده دارند پیش‌بینی‌های مشابهی با مدل‌های موجود ارائه می‌دهند.

پ)

- خیر، adaboost نیازی به مشتق‌پذیر بودن یادگیرنده‌های ضعیف ندارد زیرا بر وزن‌دهی مجدد و رای‌گیری متکی است و نه بر روش‌های بهینه‌سازی مبتنی بر گرادیان.
- Boosting ، زیرا ماهیت دنباله‌دار boosting و وابسته بودن هر مدل به نتایج مدل قبلی اجازه موازی‌سازی را نمی‌دهد،. درحالی که در bagging یادگیرنده‌های پایه به دلیل استقلال می‌توانند به صورت موازی آموزش داده شوند.

۳. (۲۰ نمره) در این سوال یک دسته‌بند KNN با متریک فاصله L_2 در نظر بگیرید. کلاس‌ها را تماما دوحالته (+/-) در نظر خواهیم گرفت. به سوالات زیر با توجه به مجموعه‌دادهٔ مشخص شده در تصویر پاسخ دهید.



الف) به ازای چه مقدار k خطای این دسته بند کمینه می‌شود؟ مقدار این خطا چقدر است؟
ب) چرا استفاده از مقادیر بسیار زیاد یا بسیار کم برای k می‌تواند منجر به خطا شود؟

پ) فرض کنید از روش Leave One Out Cross Validation استفاده کنیم. به ازای چه مقدار k خطای دسته‌بندی کمینه می‌شود؟ مقدار این خطا چقدر است؟(استفاده از ابزارهای sklearn برای به دست آوردن k بهینه مجاز است.).

ت) مرز تصمیم برای دسته‌بند 1-NN را برای این مجموعه‌داده در تصویر نشان دهید.

حل.

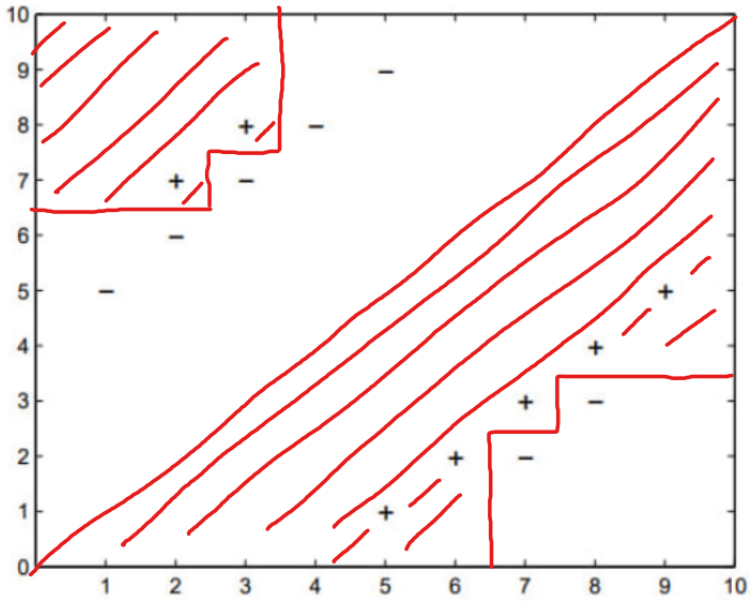
الف) چون هر نقطه‌ای همسایهٔ خودش (نزدیکترین همسایه خودش) به شمار می‌رود، 1-NN کمترین خطا را بدست می‌دهد که همان صفر است.

ب) **مقادیر زیاد** k : خط پایین سمت راست تصویر که دو دادهٔ منفی روی آن قرار دارند را درنظر بگیرید؛ این خط بوسیله خطی متشکل از داده‌های مثبت، از سایر داده‌های منفی جدا می‌شود. در صورت افزایش k ، دادگان مثبت نیز در دسته‌بندی درنظر گرفته می‌شوند که موجب افزایش خطا خواهند شد.

مقادیر اندک k : در دو سمت مجموعه‌داده که دادگان dense هستند، می‌توان مشاهده کرد که اکثر داده‌هایی که فاصله اقلیدسی کمی از یکدیگر دارند، عضو کلاس‌های متفاوت هستند، که در صورت کوچک گرفتن اندازه همسایگی این امر موجب افزایش خطا خواهد شد.

پ) k = 5, error = 0.28571 برای بدست آوردن k بهینه می‌توانید از ابزارهای sklearn استفاده کنید (کد ضمیمه پاسخنامه شده است.).

ت)



۴. (۲۰ نمره)

تابع هزینه الگوریتم خوشه‌بندی k-means به شکل زیر تعریف می‌شود:

$$L = \sum_{j=1}^{k} \sum_{x_i \in S_j} \|x_i - \mu_j\|^2$$

که در آن $x_1, x_2, \ldots, x_n$ نمونه‌ها و $\mu_1, \mu_2, \ldots, \mu_k$ مراکز خوشه‌ها هستند. منظور از S_j نیز مجموعه‌ای از نمونه‌هاست که به مرکز μ_j نزدیک‌تر از هر خوشه‌ی دیگر هستند.

الف) یک مرحله از الگوریتم را در نظر بگیرید که برچسب داده‌ها y_j ثابت است و مراکز خوشه‌ها μ_i به‌روزرسانی می‌شوند. نشان دهید که برای کمینه کردن تابع هزینه در این مرحله، کافی است میانگین هر خوشه به‌عنوان مرکز آن خوشه قرار گیرد.

ب) آیا الگوریتم k-means نسبت به مقداردهی اولیه مراکز خوشه‌ها حساس است؟ آیا این الگوریتم به طور تضمینی همگرا می‌شود؟

پ) در مرحله‌ای از الگوریتم k-means که در آن میانگین خوشه‌ها μ_i ثابت‌اند و برچسب‌های نقاط داده y_j به‌روزرسانی می‌شوند، گاهی ممکن است یک نقطه X_j به چندین مرکز خوشه با فاصله مساوی نزدیک باشد. اگر یکی از گزینه‌ها این باشد که X_j در همان خوشه‌ای که در تکرار قبلی بوده باقی بماند، توضیح دهید چرا بهتر است این گزینه انتخاب شود؟ اگر این اصل رعایت نشود، چه مشکلی ممکن است پیش بیاید؟

حل.

الف) اگر n_j تعداد نقاط نمونه اختصاص داده شده به خوشه j باشد، و از تابع هزینه L استفاده کنیم، مشتق تابع هزینه نسبت به μ_i به صورت زیر خواهد بود:

$$\frac{\partial L}{\partial \mu_i} = \sum_{y_j=i} (\mu_i - X_j)$$

با برابر صفر قرار دادن این مشتق، خواهیم داشت:

$$\sum_{y_j=i} \mu_i = \sum_{y_j=i} X_j$$

که معادل است با:

$$n_j \mu_i = \sum_{y_j=i} X_j$$

بنابراین:

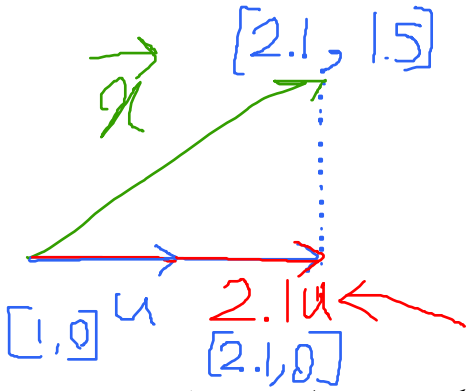
$$\mu_i = \frac{1}{n_j} \sum_{y_j=i} X_j$$

که همان میانگین نقاط نمونه اختصاص داده شده به خوشه i است.

ب) الگوریتم k-means به صورت تضمینی به حداقل محل (نه لزوماً سراسری) همگرا می‌شود و به انتخاب مقدار k و محل مراکز اولیه‌ی خوشه‌ها حساس است زیرا این دو مقدار نقش تعیین کننده‌ای در همگرایی به یک نقطه بهینه سراسری دارند.

پ) اگر یک نقطه نمونه بین دو یا چند میانگین خوشه که فاصله‌ای مساوی دارند، دائماً جابه‌جا شود، الگوریتم k-means ممکن است به یک چرخه بی‌پایان از تغییرات برسد. در این حالت، هر بار که برچسب آن نقطه به‌روزرسانی می‌شود، بدون اینکه هیچ بهبودی در کاهش تابع هزینه حاصل شود، نقطه به خوشه دیگری منتقل می‌شود. این باعث می‌شود الگوریتم نتواند به حالت پایدار برسد. در نتیجه، ممکن است الگوریتم به پایان نرسد، حتی با وجود اینکه تابع هزینه دیگر کاهش پیدا نمی‌کند و پیشرفتی در خوشه‌بندی حاصل نمی‌شود.

۵. (۲۰ نمره) فرض کنید مجموعه‌ای از نقاط $x^{(1)}, \ldots, x^{(m)}$ داده شده‌اند. فرض کنید داده‌ها نرمال شده‌اند و دارای میانگین صفر و واریانس یک در هر بعد هستند. همچنین فرض کنید $f_u(x)$ تصویر نقطه x در جهت بردار یکه u باشد. به عبارتی اگر داشته باشیم:



$$V = \{au : a \in \mathbb{R}\}$$

آنگاه:

$$f_u(x) = \arg\min_{v \in V} \|x - v\|^2$$

نشان دهید بردار u که خطای MSE بین نقاط و تصویر آن‌ها را کمینه می‌کند، همان مولفه اصلی اول (PC_1) است. به عبارتی نشان دهید که:

$$\arg\min_{u:u^\top u=1} \frac{1}{m} \sum_{i=1}^{m} \left\|x^{(i)} - f_u(x^{(i)})\right\|^2$$

برابر اولین مولفه اصلی است.

حل.
ابتدا داریم:

$$\begin{aligned}
f_u(x) &= \arg\min_{v=au} \|x - au\|^2 \\
\frac{\partial}{\partial a}\|x - au\|^2 &= \frac{\partial}{\partial a}(x - au)^T(x - au) \\
&= \frac{\partial}{\partial a}(x^T x - 2au^T x + a^2 u^T u) \\
&= -2u^T x + 2au^T u = 0 \quad \Rightarrow \quad a = \frac{u^T x}{u^T u} \\
f_u(x) &= au = \frac{u^T x}{u^T u} u
\end{aligned}$$

حال عبارت را بازنویسی می‌کنیم:

$$
\begin{aligned}
\arg\min_{\|u\|=۱} \sum_{i=۱}^{m} \|x^{(i)} - f_u(x^{(i)})\|^۲ &= \arg\min_{\|u\|=۱} \sum_{i=۱}^{m} \|x^{(i)} - uu^T x^{(i)}\|^۲ \\
&= \arg\min_{\|u\|=۱} \sum_{i=۱}^{m} (x^{(i)} - uu^T x^{(i)})^T (x^{(i)} - uu^T x^{(i)}) \\
&= \arg\min_{\|u\|=۱} \sum_{i=۱}^{m} \left(x^{(i)T} x^{(i)} - u^T x^{(i)} x^{(i)T} u \right) \\
&= \arg\min_{\|u\|=۱} \sum_{i=۱}^{m} \left(x^{(i)T} x^{(i)} - ۲(u^T x^{(i)})^۲ + (u^T x^{(i)})^۲ \right) \\
&= \arg\min_{\|u\|=۱} \sum_{i=۱}^{m} -(u^T x^{(i)})^۲ \\
&= \arg\max_{\|u\|=۱} \sum_{i=۱}^{m} (u^T x^{(i)})^۲ \\
&= \arg\max_{\|u\|=۱} u^T \left(\sum_{i=۱}^{m} x^{(i)} x^{(i)T} \right) u
\end{aligned}
$$

حل معادله زیر را با استفاده از روش لاگرانژ انجام می‌دهیم:

$$\arg\max_{\|u\|=۱} u^T \Sigma u$$

conditional optimization

حال برای حل این مسئله از ضرایب لاگرانژ استفاده می‌کنیم:

$$\mathcal{L}(u, \lambda) = u^T \Sigma u - \lambda(u^T u - ۱)$$

حال برای حل این معادله مشتق لاگرانژین را محاسبه می‌کنیم:

$$\frac{\partial}{\partial u} \mathcal{L}(u, \lambda) = ۲\Sigma u - ۲\lambda u = ۰ \quad \Rightarrow \quad \Sigma u = \lambda u$$

پس می‌توان نتیجه گرفت که جواب این مسئله همان بردارهای ویژه‌ی ماتریس Σ هستند.

$$u^T \Sigma u = u^T \lambda u = \lambda u^T u = \lambda$$

در نتیجه این مقدار جواب بهینه‌ای است که معادل با بردارهای ویژه ماتریس Σ می‌باشد که همان اولین مولفه PCA است.