

# تشخیص الگو

مدرس: دکتر اعرابی

پروژه نهایی

Feature Conditioning Assessment using LDA, PCA,  
Kernel PCA and LLE methods

اعضای گروه:

امیررضا فرنوش

منصور صفار مهرجردی

محمدهادی شادمهر

پاییز ۱۳۹۳

## چکیده

با پیشرفتهای اخیر تکنولوژی، داده‌های عظیمی در حوزه‌های مختلف به دست آمدند که نیاز به پردازش و دسته بندی داشتند و الگوریتم‌های موجود دسته بندی و پردازش قادر به تحلیل آن حجم از داده در زمان معقول نبودند. از طرفی همه این داده‌ها از ارزش اطلاعاتی خاصی برخوردار نبوده و نیاز به روش‌هایی برای بهینه سازی داده‌ها قبل از ورود به سیستم‌های طبقه بند احساس می‌شد.

این روش‌های بهینه سازی که به feature conditioning معروفند، مورد مطالعه دانشمندان متعددی قرار گرفتند و روش‌های مخلفی از جمله LLE, PCA, Kernel PCA و SVM ارایه شدند که هر یک مزایا و معایب خاص خود را داشتند.

در این نوشه به بررسی این چهار روش، مقایسه و ارزشیابی آن‌ها می‌پردازیم. به منظور ارزشیابی آن‌ها از دو دسته داده مختلف که ابعادی در حدود پنجاه دارند استفاده شده است. در آخر نیز با طبقه بند SVM داده‌های کاهش بعد یافته از این چهار روش را طبقه بندی و ارزیابی می‌شود.

## مقدمه

با پیشرفت‌های اخیر تکنولوژی، داده‌های عظیمی در حوزه‌های مختلف به دست آمدند که نیاز به پردازش و دسته بندی داشتند و الگوریتم‌های موجود دسته بندی و پردازش قادر به تحلیل آن حجم از داده در زمان معقول نبودند. از طرفی همه این داده‌ها از ارزش اطلاعاتی خاصی برخوردار نبوده و نیاز به روش‌هایی برای بهینه سازی داده‌ها قبل از ورود به سیستم‌های طبقه بند احساس می‌شد.

آنچه که واضح است استفاده از تمام ویژگیهای داده در بسیاری از موارد بسیار پرهزینه و وقت گیر است لذا نیاز به کاهش ابعاد داده‌ها ایجاد می‌گردد. در بسیاری از موارد ویژگیهای مختلف داده حاوی اطلاعات مشابه و بعضاً غیرمرتبط با طبقه بندی که قصد انجام آن روی داده‌ها را داریم هستند. این نکته قابل ذکر است که رو شهای کاهش بعد داده بعضاً میتوانند نظم داده‌ها و همسایگی آنها را تغییر دهند و روی طبقه بندی تاثیر بگذارند.

در این نوشتۀ به بررسی این چهار روش LLE, PCA, Kernel PCA و LDA، مقایسه و ارزشیابی آن‌ها می‌پردازیم. به منظور ارزشیابی آن‌ها از دو دسته داده مختلف استفاده شده است. در آخر نیز با طبقه بند SVM داده‌های کاهش بعد یافته از این چهار روش را طبقه بندی و ارزیابی می‌شود.

## روشهای خارج از درس

### LLE (Locally Linear Embedding) •

#### توضیح

LLE یک روش کاهش بعد unsupervised است که همسایگی داده ها هم مدنظر قرار داده و داده ی کاهش بعد یافته به این روش همسایگی داده ها (از نظر هندسی) حفظ میشود در حالیکه در روشهای دیگر کاهش بعد همانند PCA ممکن است دو داده که در بعد بالاتر از هم دور هستند، هنگامیکه به بعد پایینتر نگاشت میشوند کنار همدیگر قرار بگیرند. در روش LLE ما چند همسایه از هر داده را در نظر میگیریم و آنها را به عنوان یک linear patch در نظر میگیریم. سپس هر داده را بر اساس همسایه های آن بازسازی میکنیم و سپس داده های بعد پایین را بر اساس ضرایبی که در مرحله قبلی به دست آوردهیم بازسازی میکنیم.

#### الگوریتم

N داده با بعد D در نظر بگیرید که آنها را با  $\vec{X}$  نمایش میدهیم. بردار  $\vec{X}_i$  داده i ام را نشان میدهد. ما برای هر داده K تا از نزدیکترین داده های به آن را پیدا میکنیم و سپس آن داده را بر حسب داده های همسایه اش بازسازی میکنیم. ضرایب بازسازی را  $W$  مینامیم که  $j$  نشان دهنده i میزان مشارکت داده i در بازسازی داده i است. بدیهی است برای داده هایی که خارج از محدوده i K همسایه نزدیک یک داده باشند این ضرایب صفرند و این داده ها در بازسازی آن داده هیچ مشارکتی ندارند.

برای بازسازی یکتابع خطا تعريف میکنیم به شکل معادله زیر تعريف میکنیم که در واقع تمام خطاهای بازسازی داده ها با هم جمع میکند. هدف در اینجا مینیمم کردن این تابع است که به بررسی آن میپردازیم.

$$\mathcal{E}(W) = \sum_i \left| \vec{X}_i - \sum_j W_{ij} \vec{X}_j \right|^2$$

این تابع خطا بر اساس دو شرط بهینه میشود :

۱) داده‌ی  $X_i$  تنها بر اساس  $K$  تا از نزدیکترین همسایه هایش بازسازی می‌شود. این به این معنی است که  $W_{ij} = 0$  است برای داده‌هایی که جز مجموعه  $K$  تایی برای یک داده نیستند.

۲) مجموع ضرایب بازسازی یک برای یک داده ( $X_i$ ) ۱ است به این معنی که  $\sum_j W_{ij} = 1$  است.

برای به دست آوردن ضرایب  $j$  حل بسته وجود دارد که به روش لگرانژ می‌توان آنرا به دست آورد.

داده  $x$  را با  $K$  تا از نزدیکترین همسایه هایش بازسازی می‌کنیم که آنها را با  $\eta_j$  نشان میدهیم و ضرایب آن متناظر برای بازسازی  $x$  را با  $w_j$  نشان میدهیم که مجموع این ضرایب ۱ است. خطای بازسازی این داده را می‌توانیم به صورت زیر نشان دهیم.

$$\varepsilon = \left| \vec{x} - \sum_j w_j \vec{\eta}_j \right|^2 = \left| \sum_j w_j (\vec{x} - \vec{\eta}_j) \right|^2 = \sum_{jk} w_j w_k C_{jk}$$

که  $C$  ماتریس کوواریانس محلی (local covariance) است که طبق معادله زیر تعریف می‌شود.

$$C_{jk} = (\vec{x} - \vec{\eta}_j) \cdot (\vec{x} - \vec{\eta}_k)$$

با حل تابع خطای روش لگرانژین و با در نظر گرفتن شرایط بالا به جواب به فرم بسته زیر میرسیم.

$$w_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{lm} C_{lm}^{-1}}$$

در عمل روش دیگری برای به دست آوردن  $W$  به کار گرفته می‌شود به این صورت که ابتدا معادلات خطی به فرم  $\sum_j C_{jk} w_k$  را حل می‌کنیم و سپس ضرایب را rescale می‌کنیم تا مجموعشان ۱ شود.

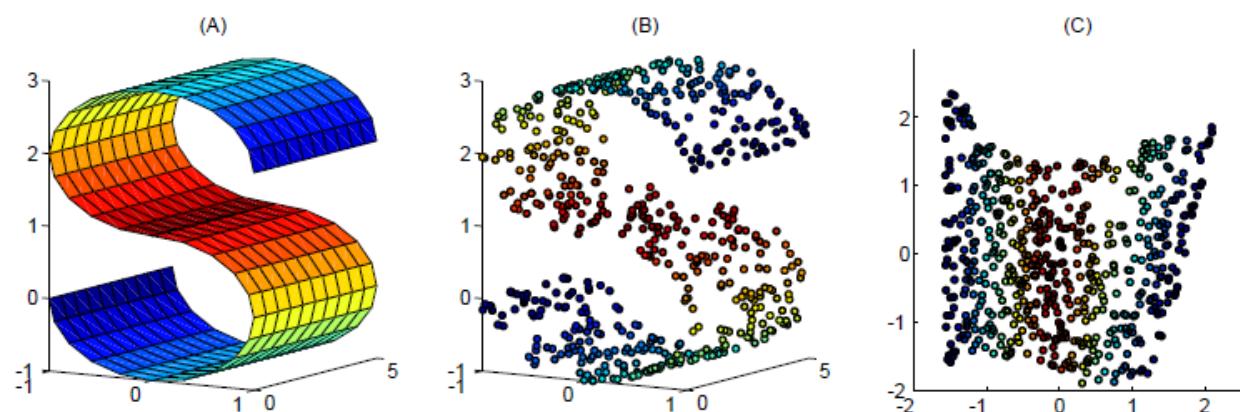
ماتریس کوواریانس محلی ( $C$ ) یک ماتریس متقارن و semi-positive definite است. در الگوریتم های LLE ممکن است حالاتی پیش بیاید که  $K$  از بعد داده‌ها ( $D$ ) بیشتر باشد که سبب می‌شود ماتریس  $C$  singular (ماتریس عناصر قطر اصلی ماتریس را در ماتریس  $I$  ضرب کرده و به آن اضافه می‌کنیم) شود. برای حل این مشکل ضرایبی از مجموع عناصر قطر اصلی ماتریس را در ماتریس  $I$  ضرب کرده و به آن اضافه می‌کنیم.

نکته کلیدی در مورد این ضرایب این است که برای هر یک از داده ها این ضرایب در اثر چرخش، جابجایی و scale کردن داده ها ، تغییری نخواهند کرد. حال یک نگاشت خطی را در نظر بگیرید که میتواند شامل چرخش (rotation)، جابجایی (translation) و scale کردن داده ها باشد. این نگاشت داده ها را از فضای  $D$  بعدی به فضای با بعد پایینتر  $d$  نگاشت میکند. از آنجا که این ضرایب  $W$  مستقل از چرخش و جابجایی و scale هستند، لذا انتظار داریم همین ضرایب که میتوانند در بعد بالا داده ها را بازسازی کنند، بتوانند در بعد پایینتر  $d$  هم اینکار را انجام دهند.

در مرحله آخر این الگوریتم هر داده  $X_i$  را به داده  $Y_i$  در بعد پایینتر نگاشت میکنیم. در این مرحله  $Y_i$  را به گونه ای نگاشت میکنیم تاتابع هزینه زیر مینیمم گردد.

$$\Phi(Y) = \sum_i \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2$$

تصویر زیر نمونه ای از کاربرد روش LLE را نشان میدهد. در اینجا داده ها در فضای سه بعدی قرار دارند (A) و ما به اندازه کافی از آنها نمونه برداری میکنیم تا نشان دهنده ای فضای  $S$  شکلی که در آن قرار دارند باشند(B) سپس آنها را به فضای دو بعدی نگاشت میکنیم (C). نکته حائز اهمیت در اینجا این است که همانطور که از مقایسه دو شکل A و C مشاهده میشود، LLE همسایگی داده ها را حفظ میکند و داده هایی که در فضای بعد بالا (۳) در کنار هم قرار داشتند، در فضای با بعد پایینتر هم در کنار هم بودند.



## Kernel PCA •

استاندارد کاهش ابعاد را تنها از طریق پیدا کردن روابط خطی ممکن می سازد. اگر داده ها ساختار های PCA پیچیده تری داشته باشند، به نحوی که نتوان آنها را به خوبی در یک زیر فضای خطی نمایش داد، استاندارد نمی تواند مفید واقع شود. خوشبختانه kernel PCA به ما این امکان را می دهد تا PCA استاندارد را به روشی برای کاهش ابعاد از طریق پیدا کردن روابط غیر خطی تعمیم دهیم. در ادامه روشی برای انجام تحلیل مولفه اصلی به صورت غیر خطی پیشنهاد شده است. با بکارگیری توابع کرنل می توان مولفه های اصلی را در فضاهایی با ابعاد ویژگی بالا به خوبی محسنه کرد که این فضاهای ویژگی توسط یک نگاشت غیرخطی به فضای ورودی مرتبط می شود.

فرض کنید که ابتدا داده ها را به صورت غیر خطی به فضای ویژگی  $F$  نگاشت دهیم:

$$\phi: R^N \rightarrow F, \quad x \rightarrow X$$

ما نشان می دهیم که حتی اگر  $F$  ابعاد بزرگ دلخواهی داشته باشد، به ازای انتخاب های مناسب  $\phi$  ما همچنان می توانیم PCA را در فضای  $F$  انجام دهیم. اینکار با استفاده از توابع کرنل که در مبحث SVM شناخته می شوند انجام می پذیرد.

فرض کنید داده هایی را که به فضای ویژگی  $\phi(x_1), \phi(x_2), \dots, \phi(x_l)$  نگاشت دادیم مرکزی شده باشند یعنی  $\sum_{k=1}^l \phi(x_k) = 0$  که  $l$  تعداد کل داده ها است. برای انجام PCA بر روی ماتریس کواریانس

$$C = \frac{1}{l} \sum_{j=1}^l \phi(x_j) \phi(x_j)^T \quad (1)$$

ما باید مقادیر ویژه  $\lambda \geq 0$  و بردارهای ویژه  $V \in F \setminus \{0\}$  را پیدا کنیم که رابطه  $\lambda V = CV$  را ارضاء کند. با جایگذاری (1) لازم به ذکر است که تمامی جواب های  $V$  در  $\text{span}(\phi(x_1), \dots, \phi(x_l))$  قرار می گیرد. بنابراین ما می توانیم سیستم معادل زیر را در نظر بگیریم:

$$\lambda(\phi(x_k) \cdot V) = (\phi(x_k) \cdot CV) \text{ for all } k = 1, \dots, l \quad (2)$$

در نتیجه ضرایب  $\alpha_1, \dots, \alpha_l$  وجود دارد که :

$$V = \sum_{i=1}^l \alpha_i \phi(x_i) \quad (3)$$

با جایگذاری (1) و (3) در (2) و تعریف ماتریس  $K_{l,l} = K$  به صورت زیر

$$K_{ij} = (\phi(x_i) \cdot \phi(x_j)) \quad (4)$$

به رابطه زیر می رسیم:

$$l\lambda K\alpha = K^2\alpha \quad (5)$$

که  $\alpha$  بردار ستونی با عناصر  $\alpha_1, \dots, \alpha_l$  است. برای پیدا کردن حل معادله (5) یک مسئله مقادیر ویژه را حل می کنیم:

$$l\lambda\alpha = K\alpha \quad (6)$$

و مقادیر ویژه غیر صفر را بدست می آوریم. به وضوح تمامی جواب های (6) معادله (5) را ارضا می کند. به علاوه می توان نشان داد که جواب های اضافی معادله (6) تاثیری در بسط (3) ندارد و در نتیجه برای ما مطلوب نیستند.

ما جواب های  $\alpha^k$  که متعلق به مقادیر ویژه غیر صفر هستند را نرمالیزه می کنیم که اینکار از طریق نرمالیزه کردن بردارهای مرتبط در فضای  $F$  صورت می گیرد  $V^k \cdot V^k = 1$ . با بکار گیری معادلات (3) و (4) و (6) رابطه زیر بدست می آید:

$$1 = \sum_{i,j=1}^l \alpha_i^k \alpha_j^k (\phi(x_i) \cdot \phi(x_j)) = (\alpha^k \cdot K \alpha^k) = \lambda_k (\alpha^k \cdot \alpha^k) \quad (7)$$

برای استخراج مولفه اصلی، ما تصویر های نقطه آزمون  $\phi(x)$  را بر روی بردار های ویژه  $V^k$  در فضای  $F$  محاسبه می کنیم:

$$(V^k \cdot \phi(x)) = \sum_{i=1}^l \alpha_i^k (\phi(x_i) \cdot \phi(x)) \quad (8)$$

توجه کنید که هیچکدام از معادلات (4) و (8) به  $\phi(x_i)$  نیاز ندارد و تنها به ضرب های داخلی نیاز دارند. بنابراین می توانیم توابع کرنل را برای محاسبه ضرب های داخلی استفاده کنیم بدون آنکه نگاشت  $\phi$  را انجام دهیم. برای برخی کرنل ها می توان نشان داد که یک نگاشت  $\phi$  به فضای ضرب داخلی  $F$  وجود دارد که ضرب داخلی را در فضای  $F$  محاسبه می کند. کرنل هایی که به خوبی در SVM استفاده شده اند شامل کرنل چندجمله ای و کرنل گوسی هستند:

$$k(x, y) = (x \cdot y)^d \quad (9)$$

$$k(x, y) = \exp(-\|x - y\|^2 / (2\sigma^2))$$

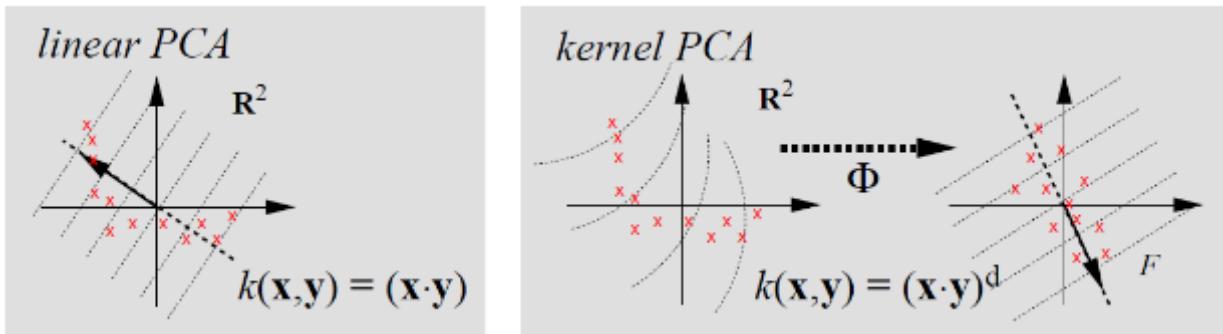
می توان نشان داد که کرنل های چندجمله ای درجه  $d$  نگاشت  $\phi$  را به فضای ویژگی ای که توسط همه  $N$  ضرب های  $d$  عنصر یک الگوی ورودی  $\text{span}$  شده است را انجام می دهند. مثلا برای حالت  $d = 2$  و  $N = 2$

$$(x \cdot y)^2 = (x_1^2, x_1 x_2, x_2 x_1, x_2^2)(y_1^2, y_1 y_2, y_2 y_1, y_2^2)^T \quad (10)$$

اگر داده های نگاشت یافته  $\phi(x)$  میانگین صفر نداشته باشند می توانیم ماتریس گرام  $\tilde{K}$  را جایگزین ماتریس کرنل  $K$  کنیم. ماتریس gram از رابطه زیر بدست می آید:

$$\tilde{K} = K - \mathbf{1}_N K - K \mathbf{1}_N + \mathbf{1}_N K \mathbf{1}_N \quad (11)$$

که  $\mathbf{1}_N$  یک ماتریس  $N \times N$  در درایه های آن  $1/N$  است.



شکل ۱ - خطوط نقطه چین مقادیر با ویژگی ثابت را نشان می دهند.

با جایگزین کردن  $\tilde{K}$  با توابع کرنل، ما الگوریتم زیر را برای Kernel PCA بدست می آوریم (عکس (1))

۱. ابتدا ماتریس ضرب داخلی  $K_{ij} = (k(x_i, x_j))$  را از داده های آموزش  $x_l$  محاسبه می کنیم
۲. ماتریس گرام  $\tilde{K}$  را با استفاده از معادله (11) حساب می کنیم
۳. معادله (6) را با قطری کردن  $K$  حل می کنیم و بردارهای  $\alpha^k$  را بدست می آوریم
۴. ضرایب بسط بردار های ویژه  $\alpha^k$  را با استفاده از معادله (7) نرمالیزه می کنیم
۵. مولفه های اصلی نقطه آزمون  $x$  (که مربوط به کرنل  $k$  هستند) را با محاسبه تصاویر بر روی بردارهای  $(kPC)_k = (V^k \cdot \phi(x)) = \sum_{i=1}^l \alpha_i^k k(x_i, x)$  ویژه از طریق معادله (8) بدست می آوریم:

## ویژگی های Kernel PCA

تمامی ویژگی های آماری و ریاضی PCA در مورد KPCA هم صدق می کند زیرا در واقع ما در PCA استاندارد را در فضای  $F$  انجام می دهیم. اگر تمام مقادیر ویژه را به صورت نزولی مرتب کنیم:

- ✓ اولین مولفه اصلی (تصاویر داده بر روی بردارهای ویژه متناظر آنها) بیشترین واریانس را دارد.
- ✓ خطای تقریب میانگین مربعات در نمایش مشاهدات به وسیله  $q$  مولفه اصلی اول مینیمم می شود.
- ✓ مولفه های اصلی ناهمبسته هستند.

## کاهش ابعاد و استخراج ویژگی

در آزمایشاتی که کاربرد ویژگی های استخراج شده از طریق Kernel PCA را در باز شناخت الگو و با استفاده از کلاسیند های خطی بررسی می کند، دو مزیت اصلی Kernel PCA غیر خطی چنین است: اول اینکه مولفه های اصلی غیرخطی در مقایسه با همان تعداد مولفه های خطی نتایج کلاسیندی بهتری بدست می دهند و دوم اینکه عملکرد مولفه های غیر خطی را می توان با بکارگیری مولفه های بیشتر از آنچه که در حالت خطی امکانش هست، بهبود بخشید. در واقع بر خلاف PCA خطی، روش پیشنهاد شده، می تواند تعداد مولفه های اصلی بیشتر از ابعاد داده ورودی استخراج کند.

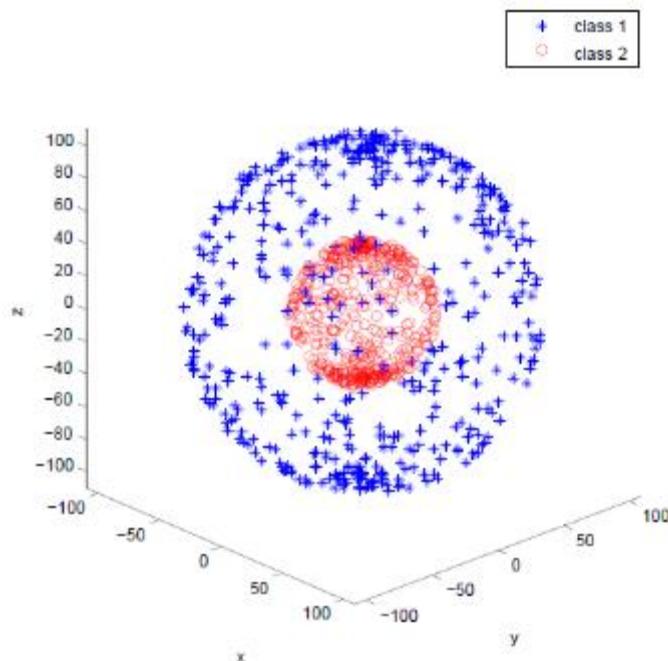
اگر ابعاد فضای ورودی کوچکتر از تعداد نمونه ها باشد استفاده از روش KPCA از نظر محسوباتی نسبت به روش PCA هزینه بیشتری دارد، زیرا نیاز به محاسبه مقادیر ویژه ماتریسی با ابعاد  $l$  در  $l$  هستیم، که  $l$  تعداد داده ها است. ولی این محاسبات اضافی می تواند مفید باشد زیرا در باز شناخت الگو نشان داده می شود که در صورتی که ویژگی های استخراج شده غیر خطی باشند استفاده از یک کلاسیند خطی کارایی خوبی خواهد داشت. البته PCA استاندارد امکان بازسازی الگوهای اصلی از مجموعه کامل مولفه های اصلی استخراج شده را از طریق بسط در پایه های بردار ویژه می دهد اما در KPCA چنین کاری ممکن نیست.

برای آنکه کارایی روش Kernel PCA در استخراج ویژگی مشخص شود، نمونه ای از نتایج آزمایش این روش بر روی داده های ساختگی و واقعی که در مقالات آورده شده است را بیان می کنیم.

## داده های ساختگی

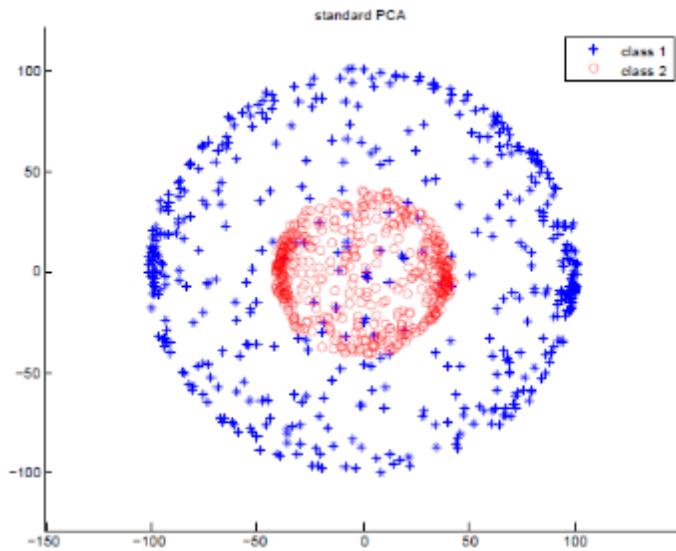
فرض می کنیم که تعداد داده های مساوی داریم که بر روی دو سطح کروی هم مرکز با شعاع های متفاوت پخش شده اند. در سیستم مختصات کروی زاویه  $\theta$  به صورت یکنواخت در بازه  $[0, \pi]$  و زاویه  $\varphi$  به صورت یکنواخت در بازه  $[0, 2\pi]$  برای داده های هر دو کلاس توزیع شده است. مشاهدات ما از داده ها مختصات آنها

در سیستم مختصات دکارتی است و هر سه مختصات آعشته به نویز گوسی شده اند. تعداد داده های هر کلاس را  $(2)$   $\sigma_{noise} = 1$  در نظر گرفتیم. نمایش سه بعدی داده ها در شکل  $(2)$  آمده است.



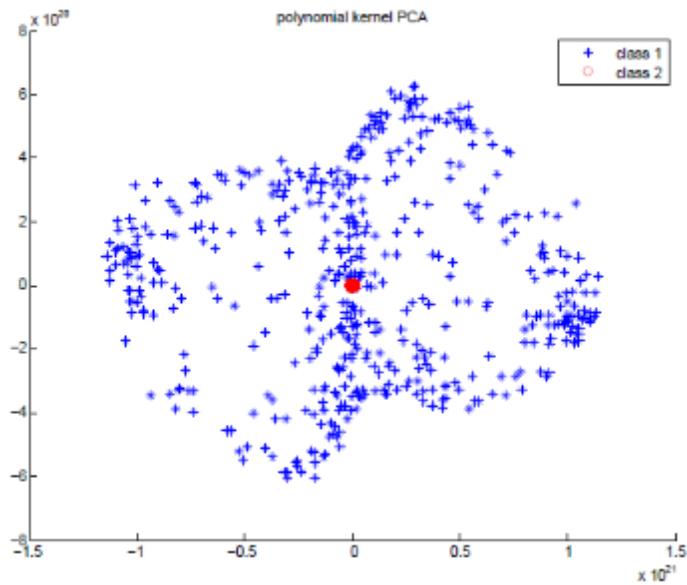
شکل ۲- نمایش سه بعدی داده ها

داده ها را با استفاده از دو روش PCA و KPCA به فضای ویژگی دو بعدی تصویر می کنیم. نتایج برای PCA در شکل  $(3)$  و برای KPCA با کرنل چندجمله ای در شکل  $(4)$  و برای KPCA با کرنل گوسی در شکل  $(5)$  آمده است.

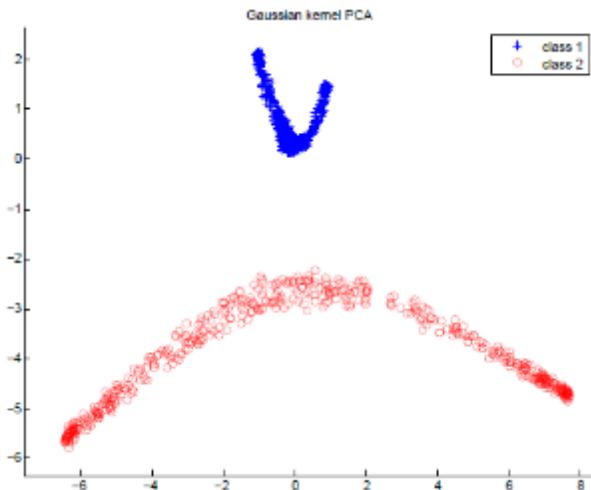


شکل ۳- نتایج برای PCA استاندارد

همانطور که مشاهده می کنید PCA استاندارد نتوانسته است داده ها را از هم جدا کند. KPCA با کرنل چند جمله ای داده های کلاس دو را متمرکز کرده است و داده های کلاس ۱ را پراکنده کرده است. هرچند داده ها از هم قابل تفکیک هستند ولی با یک خط نمی توان آنها را جدا کرد. در KPCA با کرنل گوسی داده ها را می توان به خوبی با یک خط از هم جدا کرد!



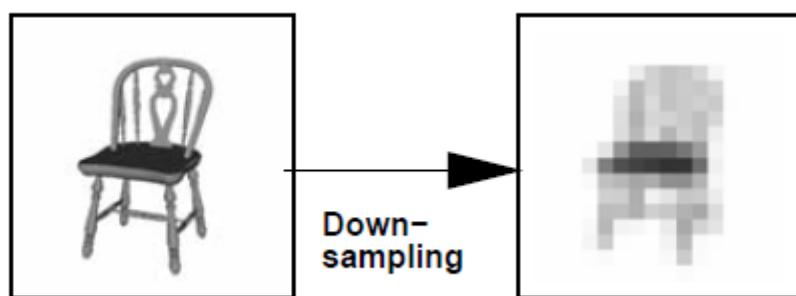
شکل ۴- نتایج KPCA چند جمله ای برای  $d=5$



شکل ۵- نتایج KPCA با کرnel گوسی برای  $\sigma = 27.8$

### داده های واقعی

در آزمایش دیگری، از عکس های پایگاه داده MPI استفاده شده است که حاوی مدل های سه بعدی صندلی است. این مجموعه داده شامل عکس های down-sample شده ۱۶\*۱۶ از ۲۵ صندلی است که از نیمکره بالایی محدوده مشاهده گرفته شده اند. برای مثال شکل (6) را ببینید. داده های آموزش حاوی ۸۹ مشاهده با فاصله های منظم از هر صندلی است. داده های آزمون حاوی ۱۰۰ مشاهده تصادفی از هر صندلی است. در این آزمایش ماتریس ضرب داخلی  $K$  برای همه ۲۲۲۵ داده آموزش محاسبه شد و از کرnel چندجمله ای برای بدست آوردن مولفه های اصلی غیر خطی استفاده شده است. برای کلاس بندی داده های آزمون از کلاس بند SVM استفاده شده است. نتایج کلاس بندی در جدول (1) آمده است.



شکل ۶- یک عکس نمونه از مجموعه داده MPI که به عکس ۱۶\*۱۶ تبدیل شده است

# of components	Test Error Rate for degree						
	1	2	3	4	5	6	7
64	23.0	21.0	17.6	16.8	16.5	16.7	16.6
128	17.6	9.9	7.9	7.1	6.2	6.0	5.8
256	16.8	6.0	4.4	3.8	3.4	3.2	3.3
512	n.a.	4.4	3.6	3.9	2.8	2.8	2.6
1024	n.a.	4.1	3.0	2.8	2.6	2.6	2.4
2048	n.a.	4.1	2.9	2.6	2.5	2.4	2.2

جدول ۱- نرخ خطا در کلاسیندی داده های آزمون SVM، برای کلاسیند MPI که با استفاده از مولفه های اصلی غیرخطی که توسط KPCA با کرنل چندجمله ای برای  $d=1$  تا ۷ استخراج شده است، آموزش دیده است.

در حالت با درجه ۱ در واقع همان PCA استاندارد را انجام می دهیم که تعداد مقادیر ویژه غیر صفر حداقل برابر ابعاد ورودی  $16^*16=256$  است. همانطور که از نتایج مشخص است در تمامی حالت ها مولفه های غیر خطی KPCA نتایج بهتری در کلاسیندی نسبت به PCA استاندارد به دست می دهند. نرخ خطا برای بهترین نتایج بین ۲ تا ۴ درصد است در حالی که برای PCA استاندارد حدود ۱۷ درصد است.

### کاهش ابعاد داده ها

روش Kernel PCA را بر روی داده های (خودمان) انجام دادیم. برای این منظور از میان کل داده ها، ۵۰۰۰ داده را به صورت تصادفی انتخاب کردیم تا از آنها در مرحله آموزش برای بدست آوردن مقادیر ویژه و بردارهای ویژه ماتریس ضرب داخلی  $K$  در روش Kernel PCA استفاده کنیم. علت اینکه از کل داده ها در مرحله آموزش استفاده نکردیم بزرگ شدن ابعاد ماتریس  $K$  و بالا رفتن پیچیدگی محاسباتی و ملاحظات حافظه بود. سپس همه داده ها را با استفاده از بردارهای ویژه بدست آمده کاهش بُعد دادیم و تعداد ویژگی ها را از ۵۲ به ۴ کاهش دادیم. از کرنل های گوسی و چندجمله ای برای این منظور استفاده کردیم.

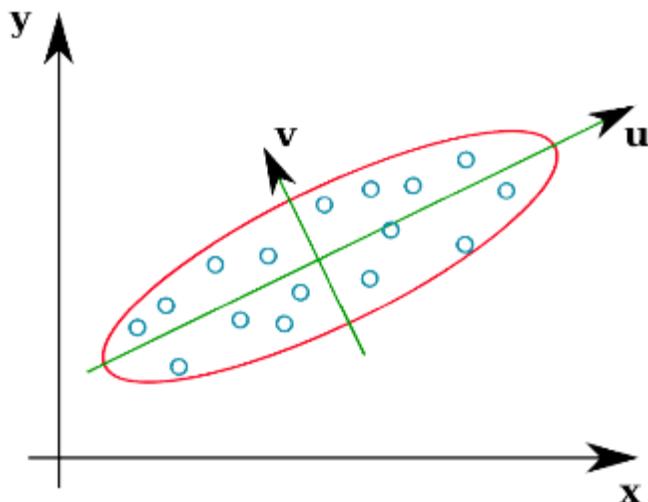
## روشهای مطرح شده در کلاس درس

### PCA (Principal Component Analysis) •

مقدمه:

آنالیز اجزای اصلی<sup>1</sup> (PCA) یک پروسه چند متغیری آماریست که قصد نشان دادن ساختار کواریانسی دسته‌ای از متغیرها را توسط یک مجموعه متغیر کوچک دارد که این مجموعه جدید ترکیب خطی ای از دسته اولیه است.

به طور خلاصه می‌توان گفت PCA جهاتی که تغییرات داده‌ها در آن بیشینه است را پیدا می‌کند و به عنوان دستگاه مختصات جدید در نظر می‌گیرد.



در این نمودار ابتدا جهت  $u$  به عنوان راستای بیشترین تغییر کل داده‌ها و سپس جهت  $v$  به عنوان تغییرات باقیمانده بیشینه است مشخص شده‌اند.

از دیدگاه جبری، این اجزای اصلی پیدا شده (PCs)، ترکیب خطی ای از متغیرهای اصلی هستند که از دیدگاه کمینه مربعات بهترین طرح را ارایه می‌کنند. از دیدگاه هندسی نیز دستگاه مختصات جدیدی ایجاد شده که داده‌ها در هر جهت از آن، بیشترین تغییرات ممکن را داشته باشند. از دید محاسباتی نیز اجزای اصلی (PC) با محاسبه مقادیر ویژه و بردارهای ویژه ماتریس کواریانس داده‌ها به دست می‌آیند. به طوریکه برداری ویژه با بیشترین مقدار ویژه متناظر با همان جهت با بیشترین تغییرات است.

<sup>1</sup> Principals Component Analyses

مقادیر ویژه به طور خلاصه به شکل زیر تعریف می‌شوند:

$$\text{Determinant}(A - \lambda I) = |(A - \lambda I)| = 0$$

که در آن  $A$  یک ماتریس مربعی و  $\lambda$  مقدار ویژه آن است. این مقادیر ویژه را در ماتریس قطری‌ای به نام  $\Lambda$  نمایش می‌دهیم.

بردارهای ویژه نیز بردارهای متناظر با هر  $\lambda$  است که در رابطه  $Ax = \lambda x$  (که  $x$  همان بردار ویژه است) صدق کنند. بردارهای ویژه که متعامد هستند را در ماتریسی هم بعد با  $A$  به نام  $\Phi$  نشان می‌دهیم.

### مراحل اجرایی PCA

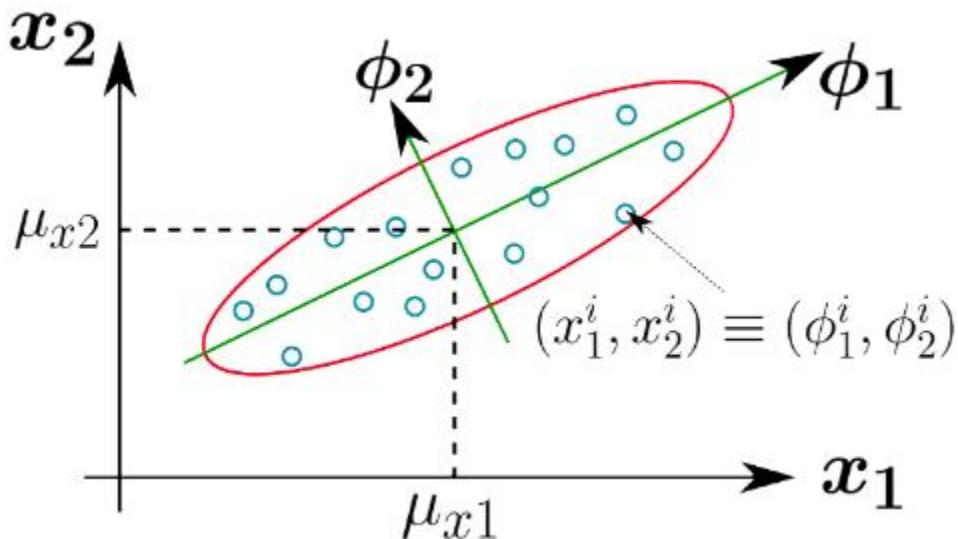
ابتدا میانگین داده‌ها در هر بعد ( $\mu_1, \mu_2, \dots$ ) محاسبه می‌شود و سپس ماتریس کواریانس ( $\Sigma$ ) را بر اساس آن بدست می‌آوریم.

در مرحله بعد مقادیر ویژه و بردارهای ویژه ماتریس کواریانس محاسبه می‌شوند.

در آخر هر نقطه از مختصات اولیه طبق فرمول زیر به فضای مختصاتی جدید انتقال پیدا می‌کند.

$$p_\phi = (p_x - \mu_x) \cdot \phi$$

همانطور که در فرمول مشخص است PCA میانگین داده‌ها را صفر نموده و آن‌ها را در دستگاه مختصاتی که هر بعدش مناسب با مقدار ویژه متناظرش، پراکندگی بیشتری را نشان می‌دهد.



## کاهش بعد در PCA

همانطور که اشاره شد در مختصات جدید هر بعد متناظر با مقدار ویژهایست که میزان پراکندگی داده ها را نشان می دهد. در نتیجه با فرض آنکه بعد با پراکندگی بیشتر اطلاعات بیشتری را در بر دارند می توانیم بعد را بر اساس مقادیر ویژه مرتب کنیم و بعد کم ارزش و بی ارزش(مقادیر ویژه صفر) را حذف کنیم و فقط از بعد ارزشمند در الگوریتم های طبقه بند استفاده کنیم.

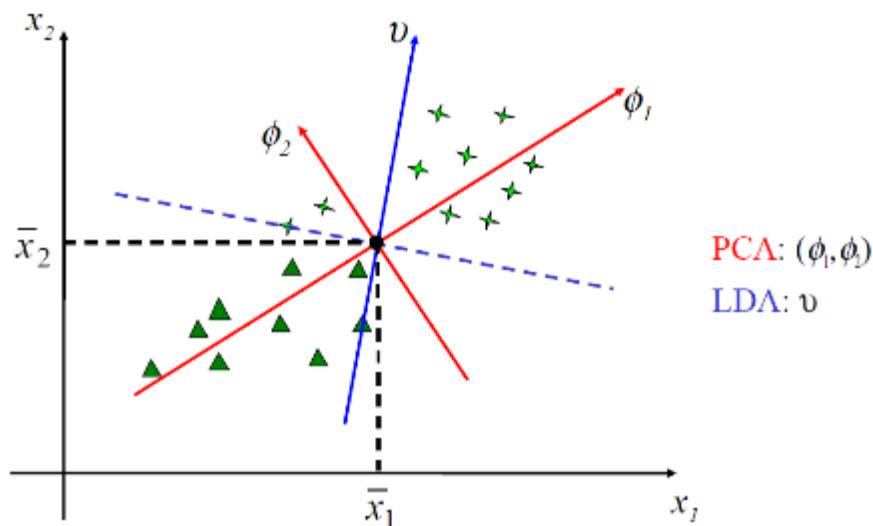
در بسیاری از موارد مقادیر ویژه صفر دیده می شوند که خبر از بی ارزش بودن بعد متناظر می دهند. البته در محاسبات، این مقادیر دقیقا صفر نیستند و باید مقادیر اپسیلون برای تشخیص صفر در نظر گرفت. همچنین برای حذف بعد با مقادیر ویژه غیر صفر نیز می توانیم مرزهای قطع را با توجه به سیر مقادیر ویژه پیدا کرده و با اضافه کردن خطایی متناظر با آنها، کاهش بعد بیشتری را لحاظ کنیم.

## LDA (Linear Discriminant Analysis) •

مقدمه:

آنالیز جدا کننده خطی<sup>۲</sup> (LDA) در حوزه استخراج ویژگی‌ها تکنیک شناخته شده ایست. این تکنیک در بسیاری از مسائل شنا سایی الگو با نتیجه موفقیت آمیزی استفاده شده است. این تکنیک به دلیل خدماتی که رونالد فیشر<sup>۳</sup> برای آن انجام داد، به آنالیز جدا کننده فیشر (FDA) نیز مشهور است.

هدف اصلی LDA نگاشت داده‌ها به فضایی است که در آن نمونه‌ها را در گروه‌های مجزا، جدا کند. LDA برای رسیدن به این هدف به دنبال افزایش جداپذیری بین گروهی و در عین حال کاهش تغییرات داخلی کلاس‌ها است. ضمناً فرض می‌شود کواریانس ماتریس همه کلاس‌ها با هم برابرند زیرا ماتریس پراکندگی<sup>۴</sup> داخلی یکسانی برای همه کلاس‌ها در نظر گرفته شده است.



مقایسه ای از دستگاه مختصات LDA, PCA

روش:

در ابتدا به چند تعریف اولیه می‌پردازیم.

میانگین هر کلاس و کواریانس هر کلاس را به شکل زیر تعریف می‌کنیم:

<sup>2</sup> Linear Discriminant Analysis

<sup>3</sup> Ronald A. Fisher

<sup>4</sup> Scatter Matrix

$$\bar{x}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} x_{i,j}$$

$$S_i = \frac{1}{(N_i - 1)} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^g N_i \bar{x}_i = \frac{1}{N} \sum_{i=1}^g \sum_{j=1}^{N_i} x_{i,j} \quad N = N_1 + N_2 + \dots + N_g$$

همچنین ماتریس پراکندگی بین کلاسی،  $S_b$  و ماتریس پراکندگی درونی هر کلاس،  $S_w$  را به شکل زیر تعریف می‌کنیم:

$$S_b = \sum_{i=1}^g N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

$$S_w = \sum_{i=1}^g (N_i - 1) S_i = \sum_{i=1}^g \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

که در آن  $x_{ij}$ ،  $j$ امین نقطه  $n$  بعدی از کلاس  $i$ ،  $\omega_i$  تعداد نقاط داده آموزش در کلاس  $i$  و  $g$  تعداد کل کلاس‌ها است.

هدف اصلی LDA پیدا کردن ماتریس نگاشتی<sup>۵</sup> است که نسبت دترمینانی  $S_b$  به  $S_w$  بیشینه شود. اصطلاحاً به این ماتریس مقیاس فیشر نیز می‌گویند.

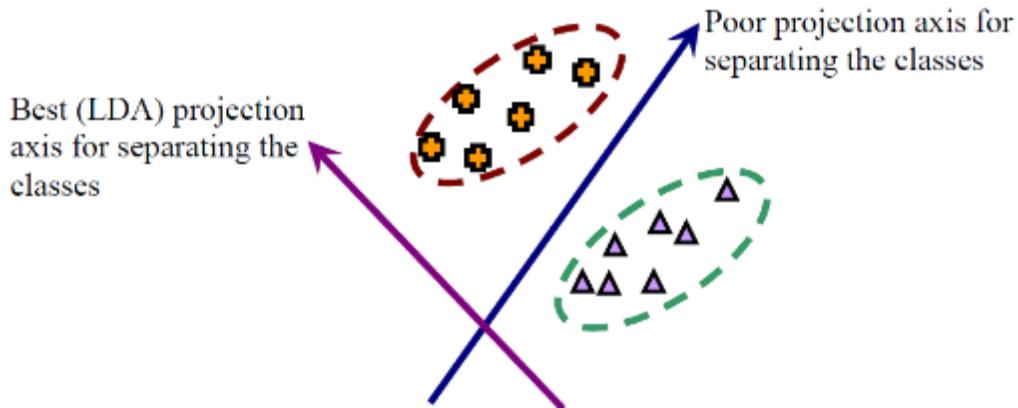
$$\Phi_{lدا} = \arg \max_{\Phi} \frac{|\Phi^T S_b \Phi|}{|\Phi^T S_w \Phi|}$$

برای آنکه شهود بیشتری نسبت به این روش پیدا کنیم به توضیحات بیشتر می‌پردازیم. دترمینان ماتریس کواریانس میزان پراکندگی یک کلاس را بیان می‌کند. ماتریس کواریانس در PCA را در نظر بگیرید. دترمینان تنها ضرب مقادیر قطر اصلی بود که مقادیر مستقل واریانس‌ها بودند. دترمینان تحت هر نگاشت<sup>۶</sup> متعامد نرمایی مقدار برابر دارد.

<sup>5</sup> Projection Matrix

<sup>6</sup> Projection

در نتیجه مقیاس فیشر، سعی می کند تا واریانس میانگین کلاس ها بیشینه و واریانس داده های هر کلاس کمینه شوند.



نشان دادیم که  $\varphi_{lda}$  در واقع پاسخ سیستم ویژه زیر است:

$$S_b\varphi - S_w\varphi A = 0$$

با ضرب کردن معکوس  $S_w$  داریم:

$$S_w^{-1} S_b\varphi - S_w^{-1} S_w\varphi A = 0$$

$$S_w^{-1} S_b\varphi - \varphi A = 0$$

$$S_w^{-1} S_b\varphi = \varphi A$$

اگر  $S_w$  ماتریس غیر تکینه‌ای<sup>7</sup> باشد در نتیجه مقیاس فیشر بیشینه می شود، هر وقت ماتریس  $\varphi_{lda}$  از بردار ویژه‌های  $S_w^{-1} S_b$  تشکیل شده باشد. این در حالی است که حداکثر از  $g-1$  مقدار ویژه غیر صفر برخوردار است. در نتیجه تنها  $g$  نقطه برای تخمین  $S_b$  موجود است.

$LDA$  یک نگاشت برای دستگاه مختصات است. در نتیجه بعد از محاسبه این نگاشت تمامی نقاط را انتقال می‌دهیم و سپس با یک دسته بند ساده به دسته بندی آن‌ها می‌پردازیم.

---

<sup>7</sup> Non-singular

## مقایسه $LDA$ و $PCA$

در یک جمله  $LDA$  به دنبال جهاتی(برای دستگاه مختصات) است که برای جدا کردن<sup>۸</sup> داده‌ها موثرترند، در حالی که  $PCA$  به دنبال جهاتی است که داده‌ها بهتر به نمایش<sup>۹</sup> بگذارد. جهاتی که در الگوریتم  $PCA$  کنار گذاشته می‌شوند ممکن است دقیقاً جهات لازم برای جدا سازی داده‌ها باشند.

---

<sup>8</sup> Discriminating

<sup>9</sup> Representing

## پیاده سازی روشها

توضیحات در مورد داده:

در این پروژه از دو دسته داده متفاوت استفاده نمودیم. داده اول همان داده درس بود که حاوی ۵ کلاس بود و هر کلاس حاوی ۲۰۰۰ داده تست و ۴۰۰۰ داده آموزش بود و ابعاد داده ها ۵۲ بود.

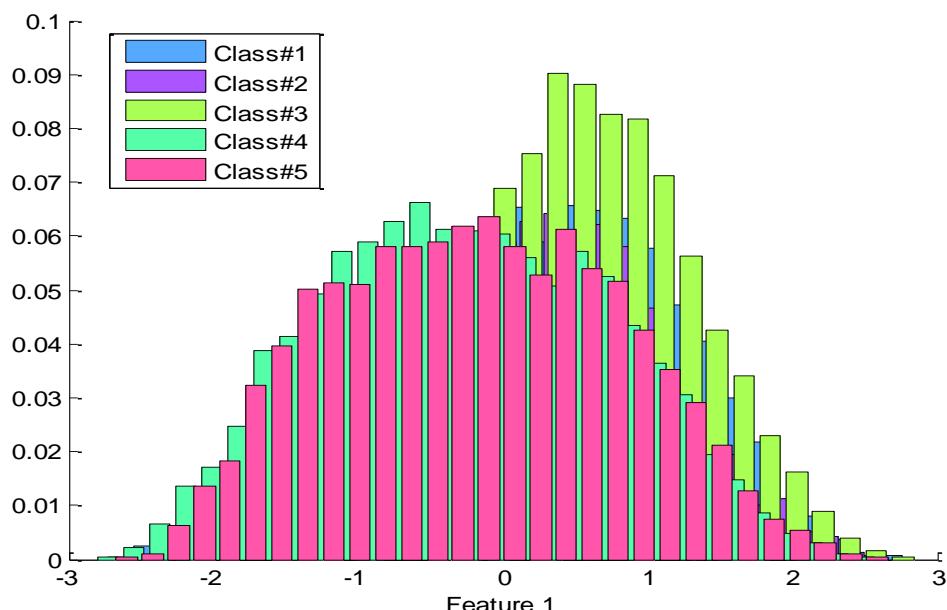
داده دوم داده ترمهای پیشین این درس بود که حاوی ۱۰ کلاس، و هر کلاس شامل ۵۰۰ داده تست و ۱۵۰۰ داده آموزش بود. هم چنین ابعاد داده ها ۵۰ بود.

در ادامه feature conditioning های بعد اول، دو بعد اول و سه بعد اول برای هر ۴ روش Scatter Plot ترسیم میگردد.

## PCA<sup>۱۰</sup> •

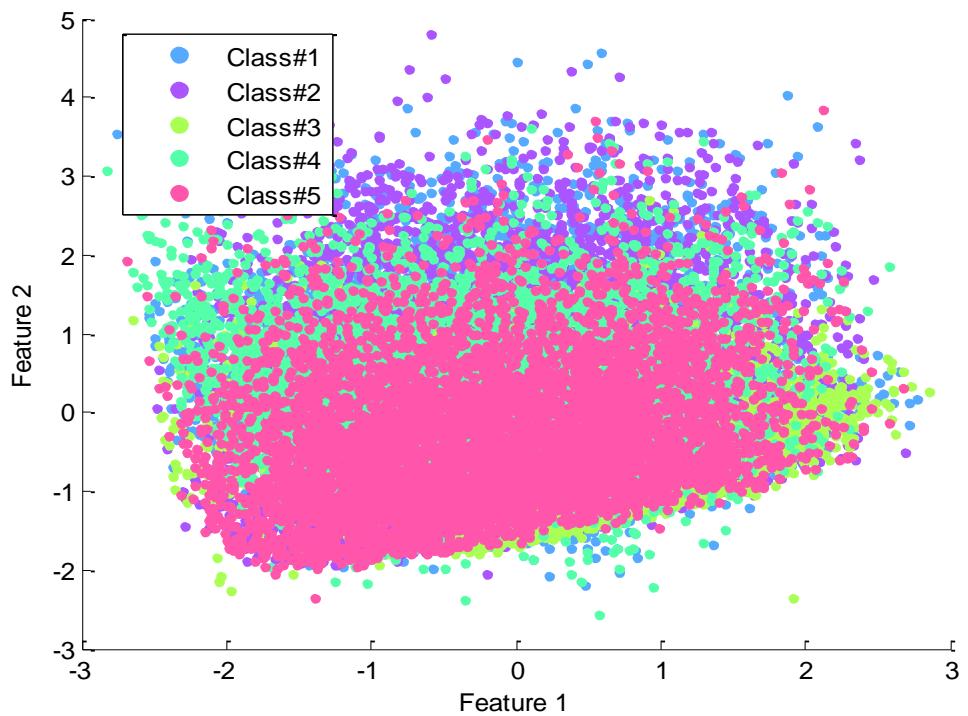
روش PCA را برای هر دو دسته داده پیاده سازی کردیم و ابعاد داده ها را در هر دو دسته به ۴ کاهش دادیم.  
Scatter plots for first dataset (dataset used in class)

First dimension:

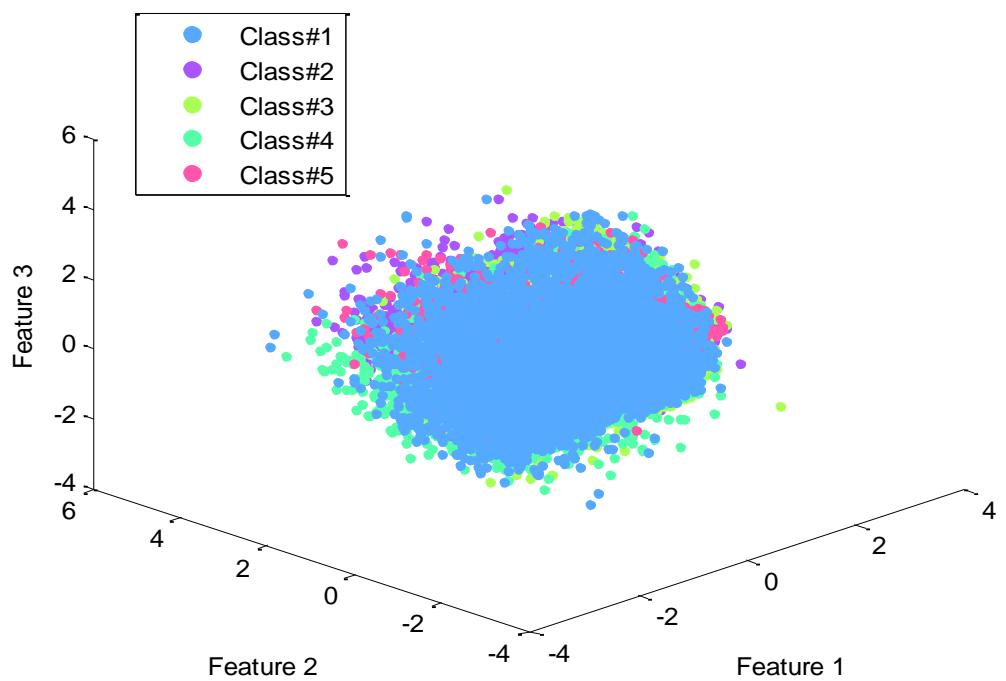


<sup>10</sup> PCA.m

First 2 dimensions:

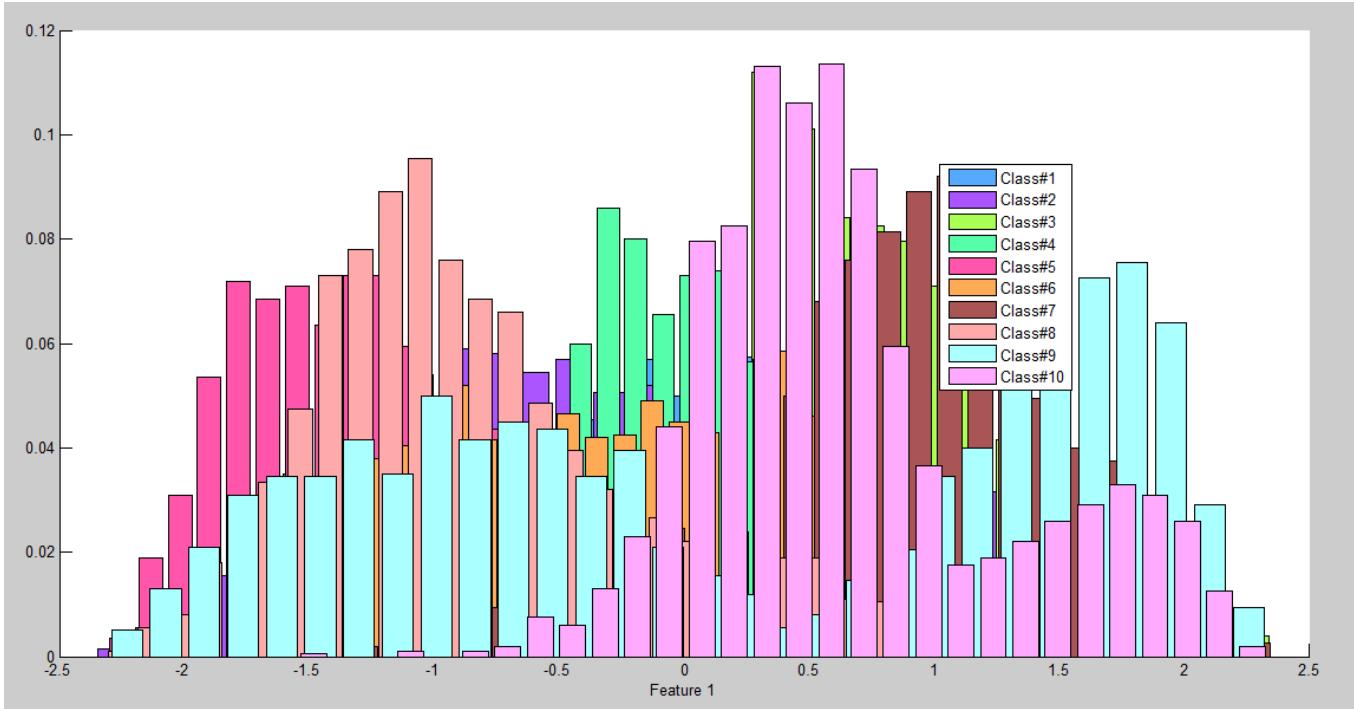


First 3 dimensions:

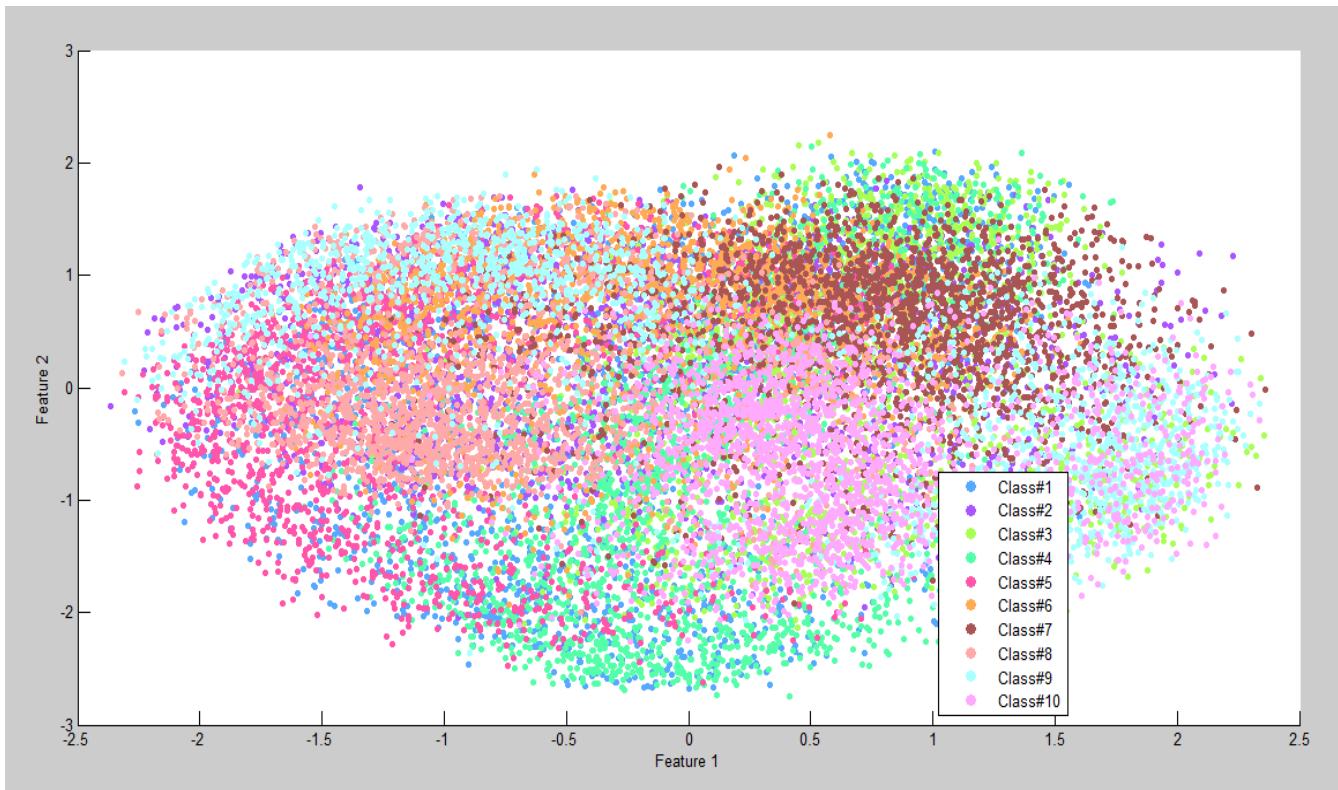


Scatter plot for second dataset (dataset used from previous terms)

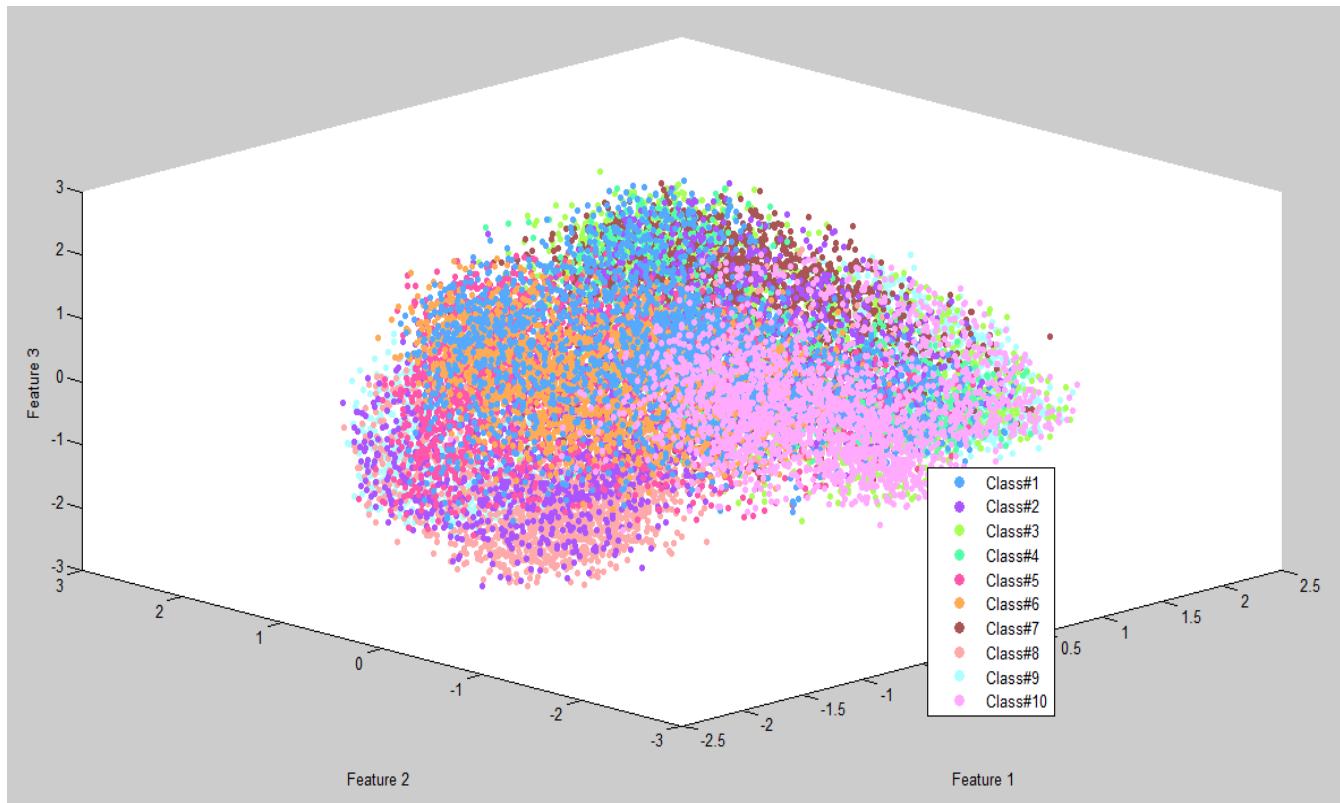
## First dimension:



## First 2 dimensions:



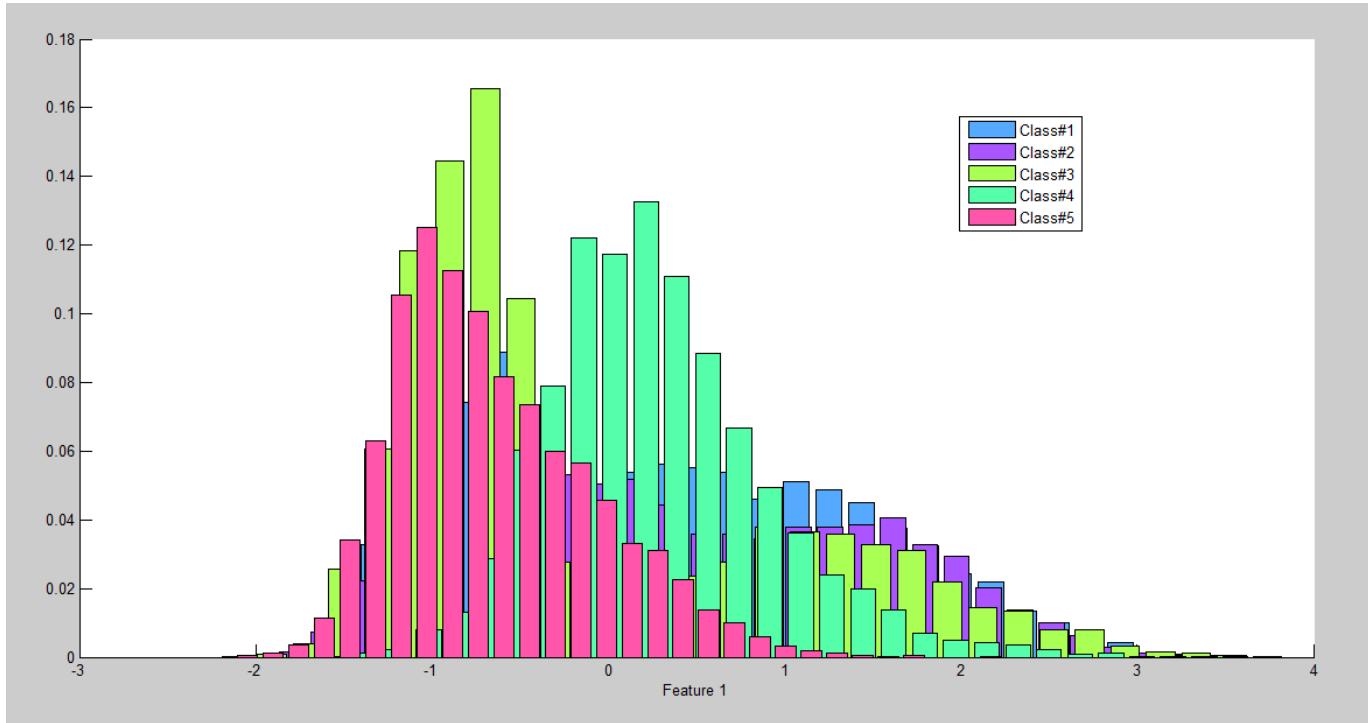
First 3 dimensions:



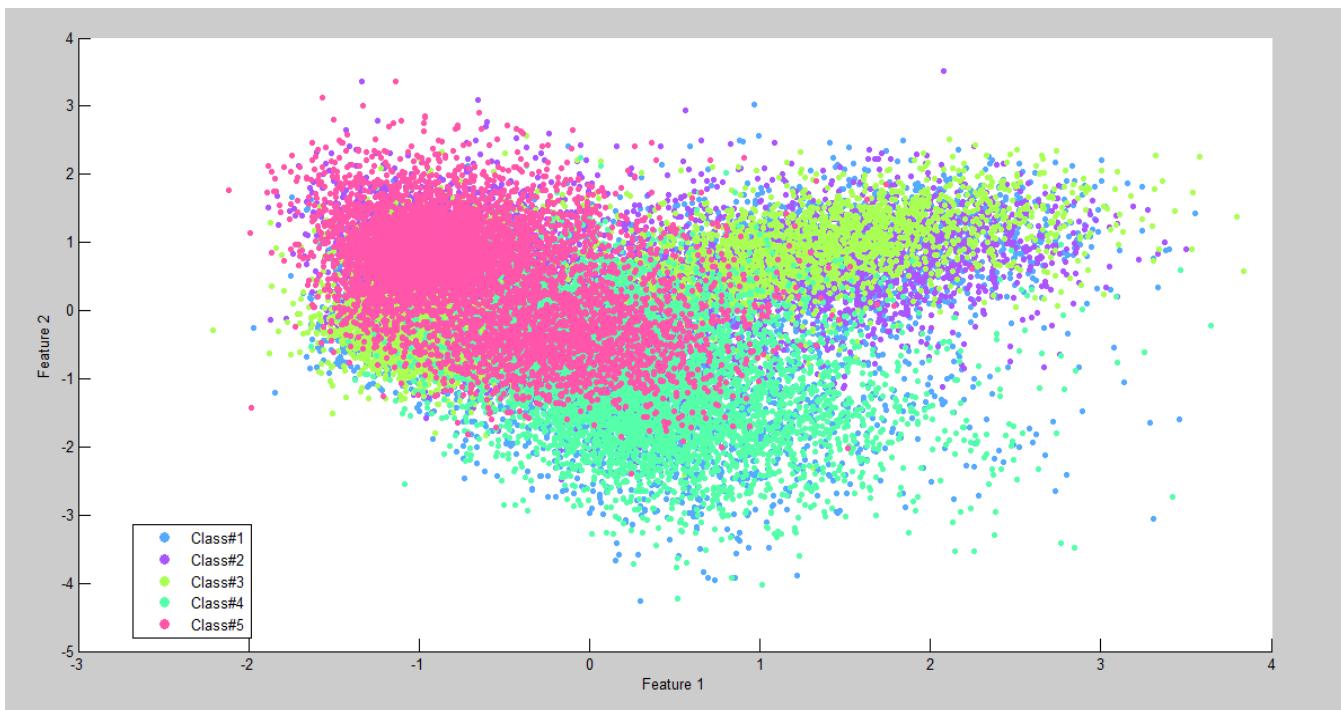
LDA •

## Scatter plot for first dataset

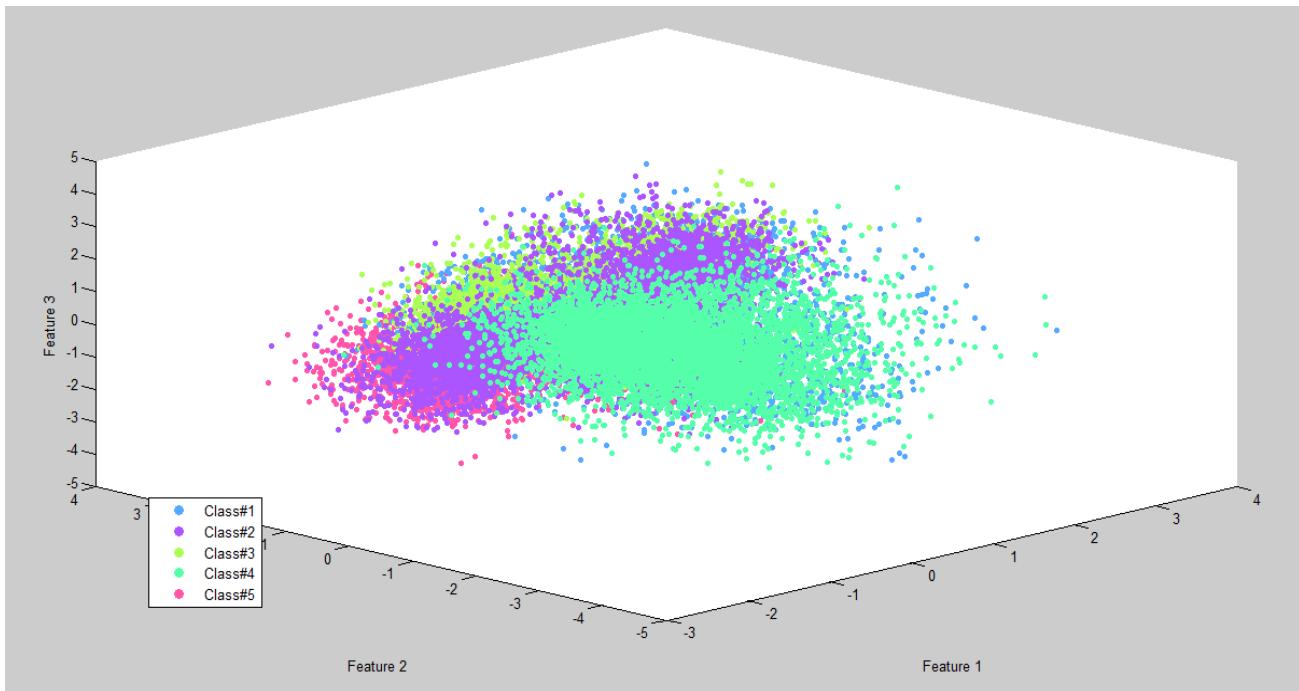
First dimension:



First 2 dimensions:

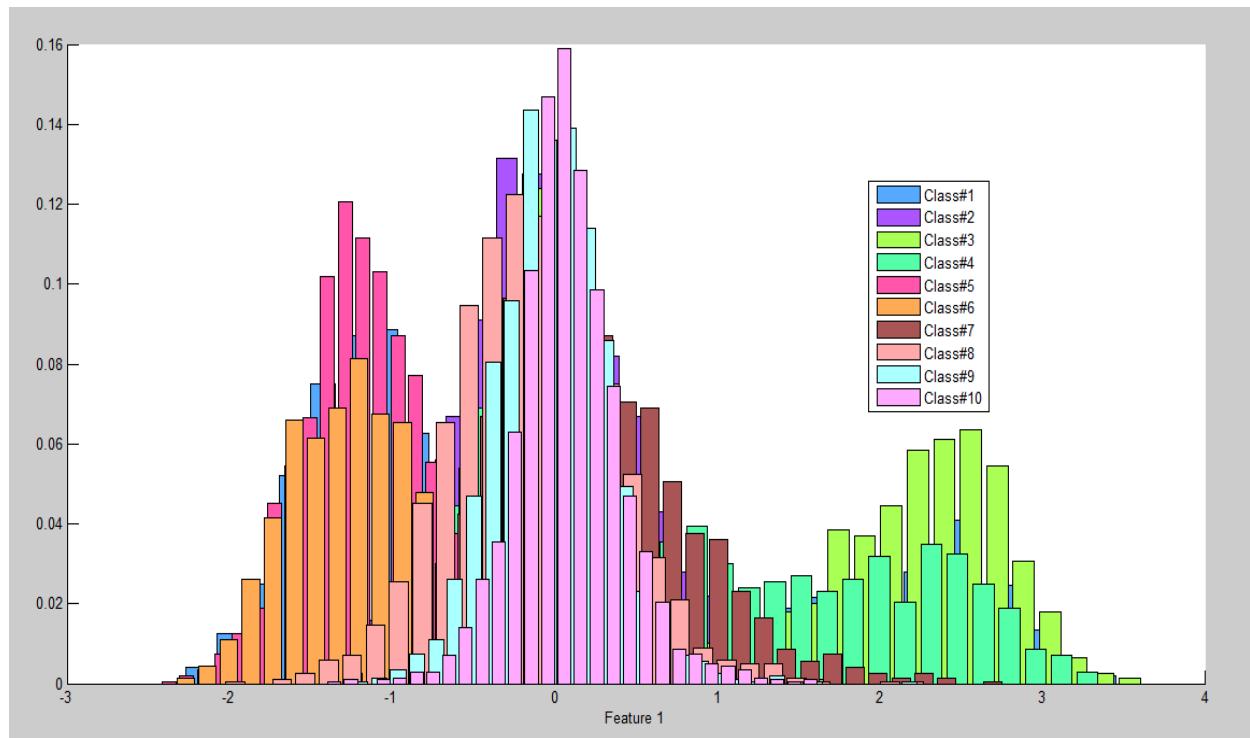


First 3 dimensions:



Scatter plots for second dataset

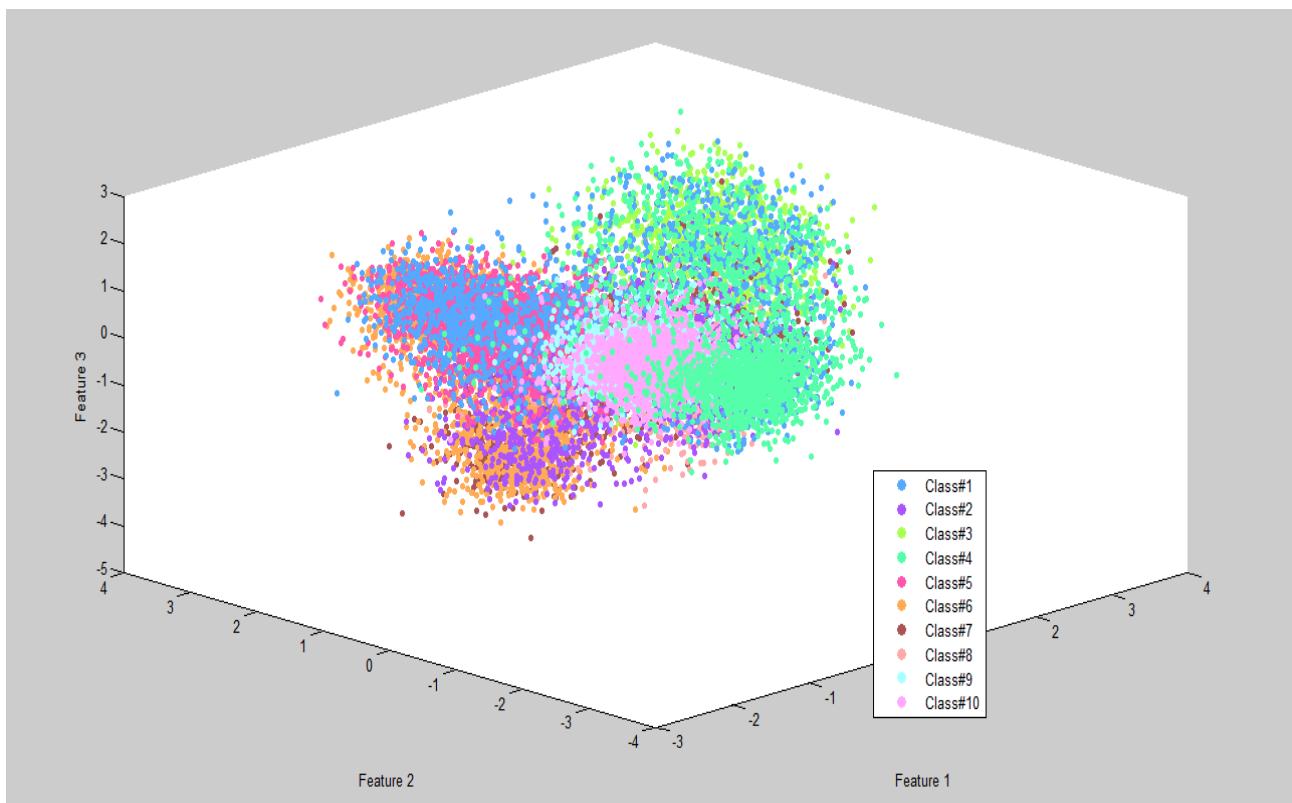
First dimension:



First 2 dimensions:



First 3 dimensions:



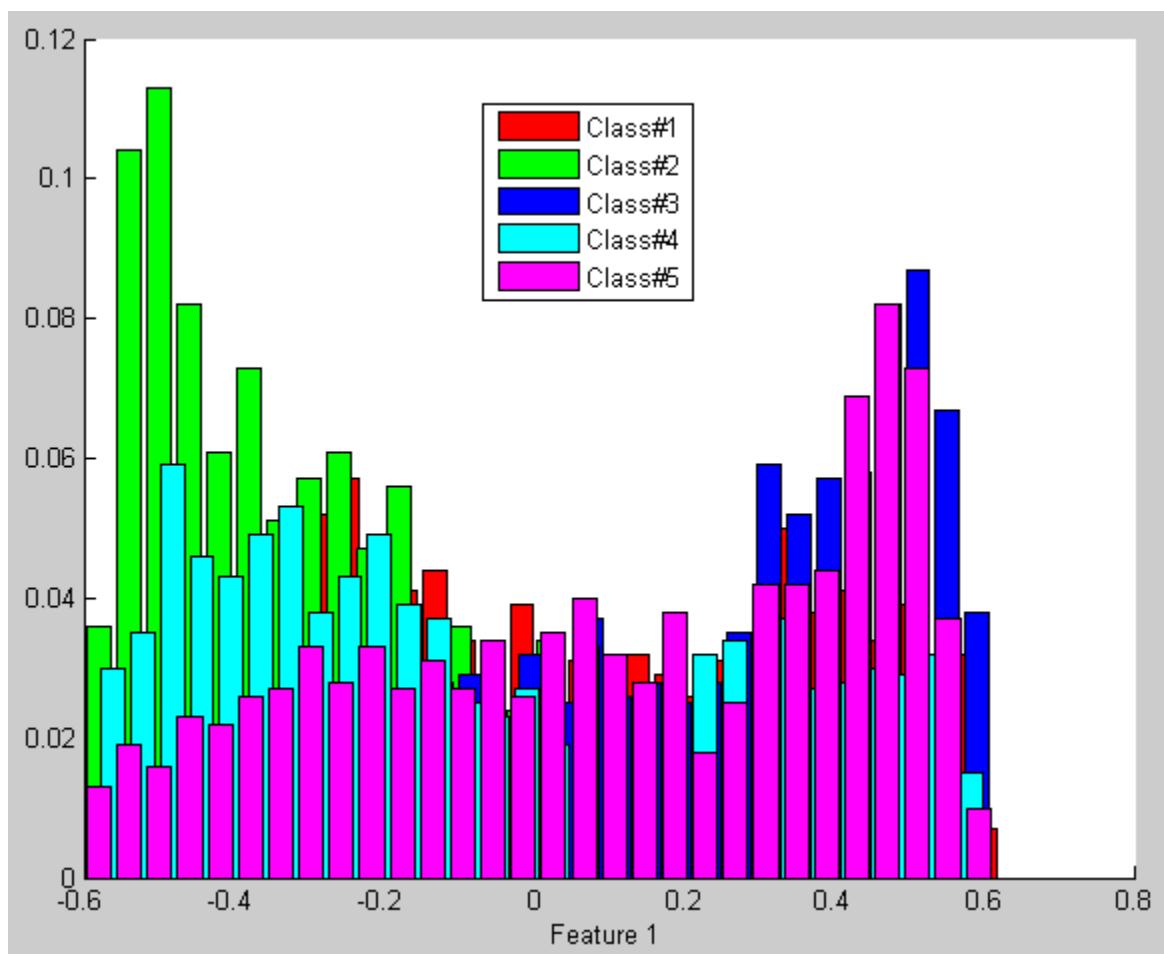
## Kernel PCA •

در روش Kernel PCA ما برای هر دسته داده دو کرنل گوسی و چندجمله ای را در اعمال کردیم که نتایج آن به صورت زیر است.

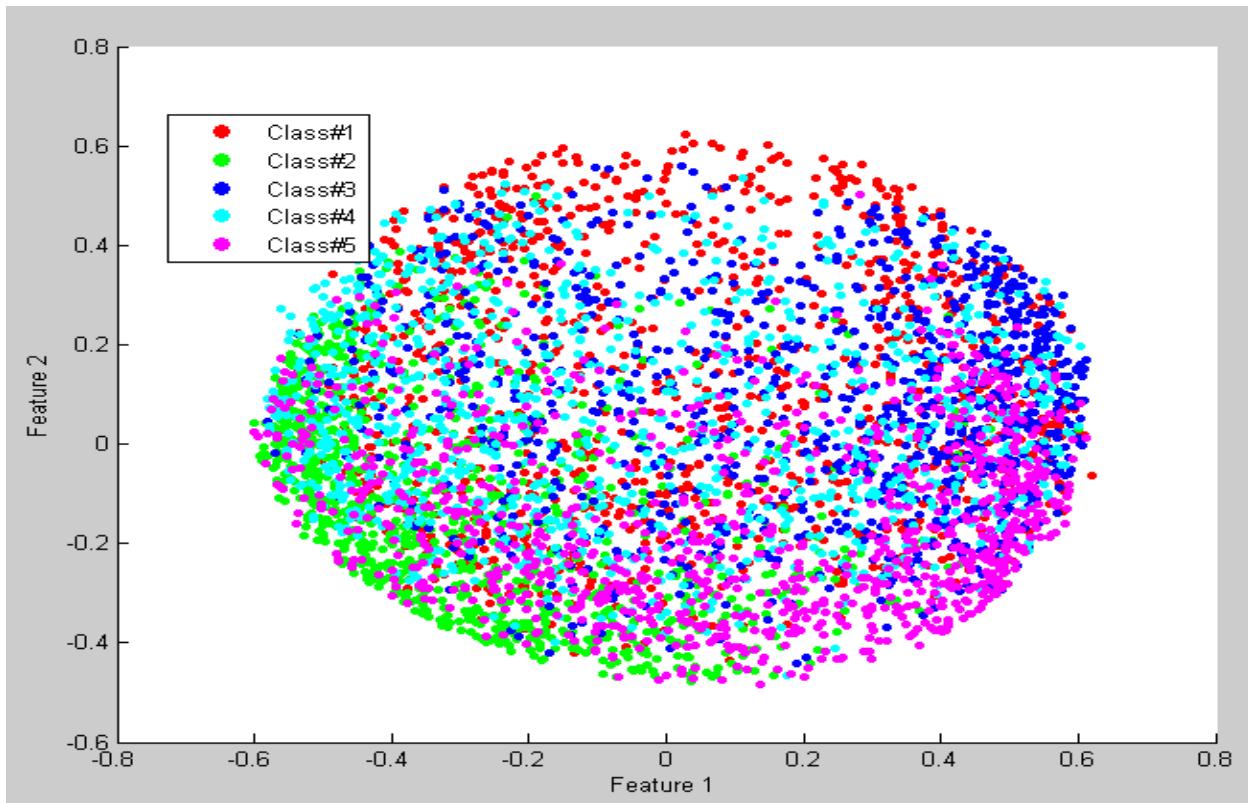
### ۱۰ با هسته گوسی و پارامتر Kernel PCA

Scatter plots for first dataset

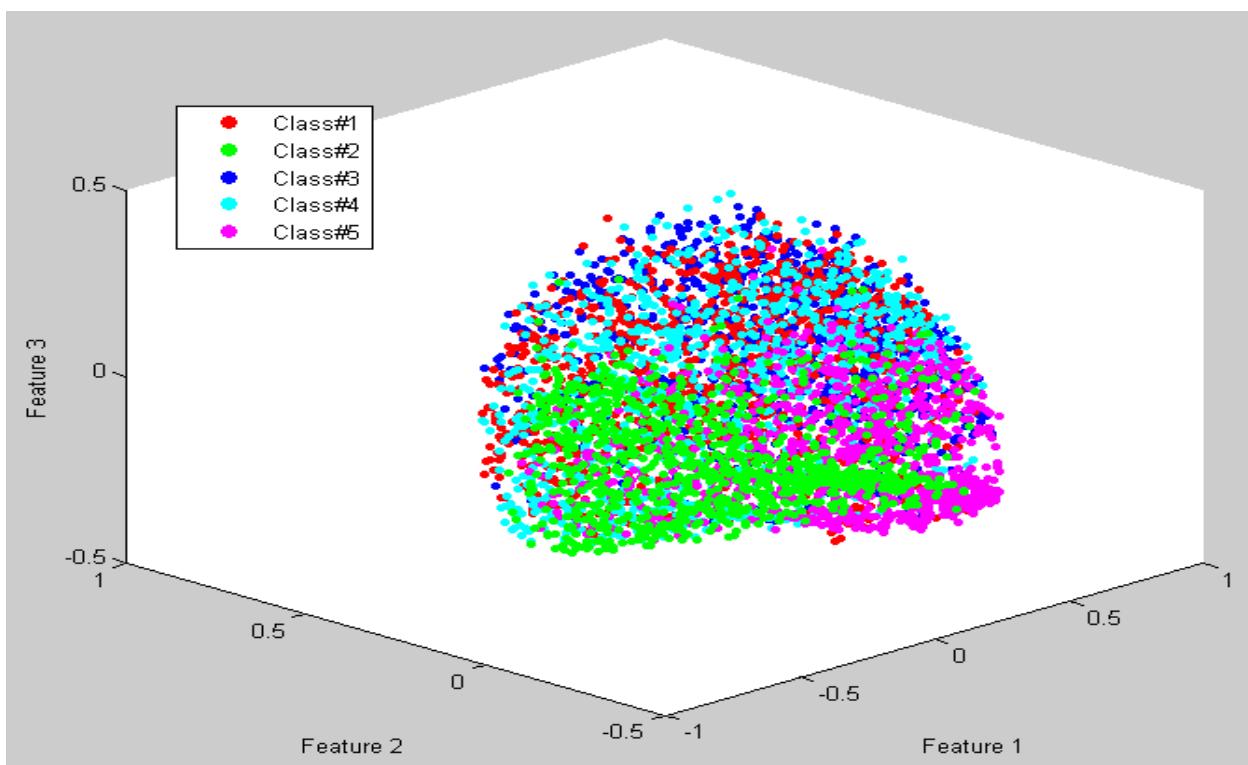
First dimension:



First 2 dimensions:



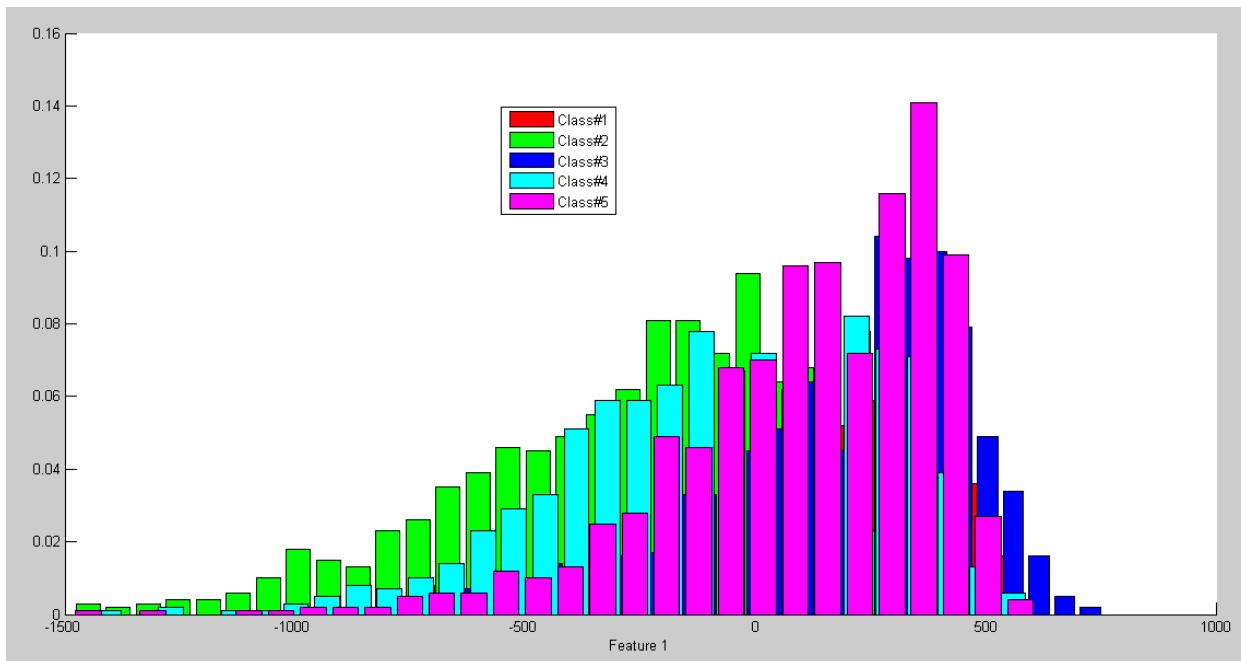
First 3 dimensions:



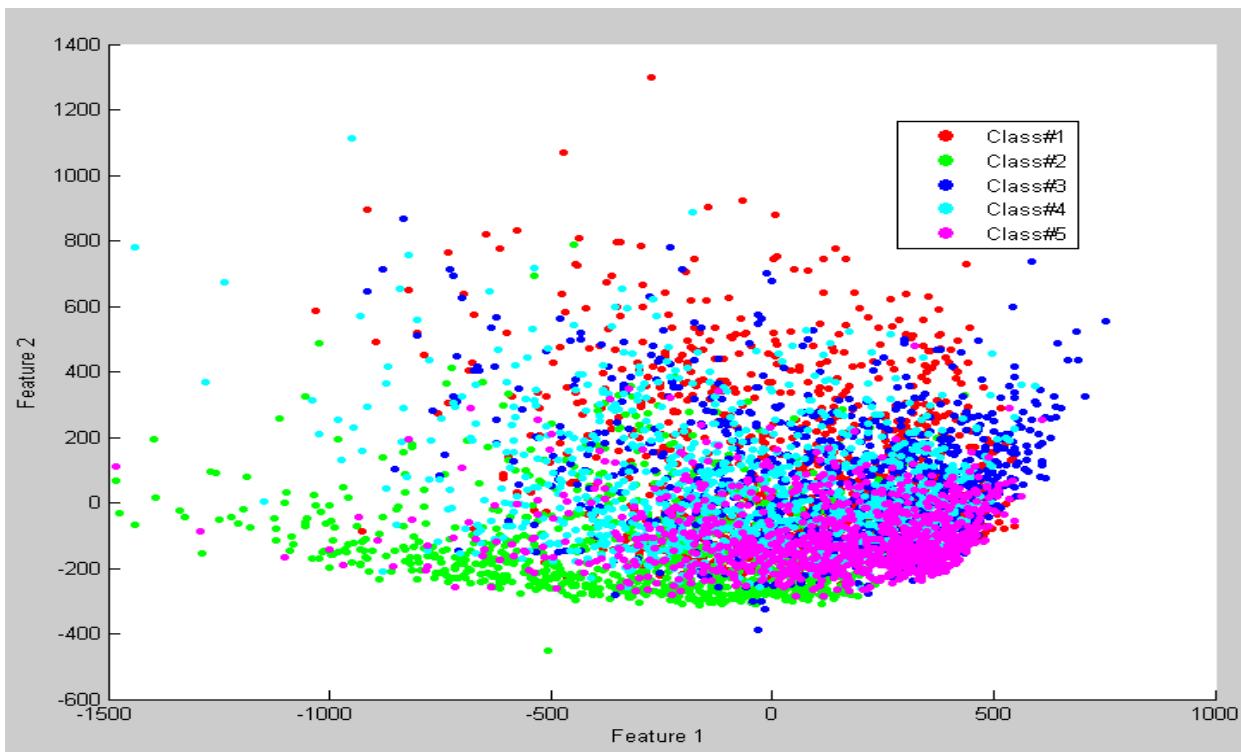
## با هسته چندجمله‌ای و پارامتر ۳ Kernel PCA

Scatter plots for first dataset

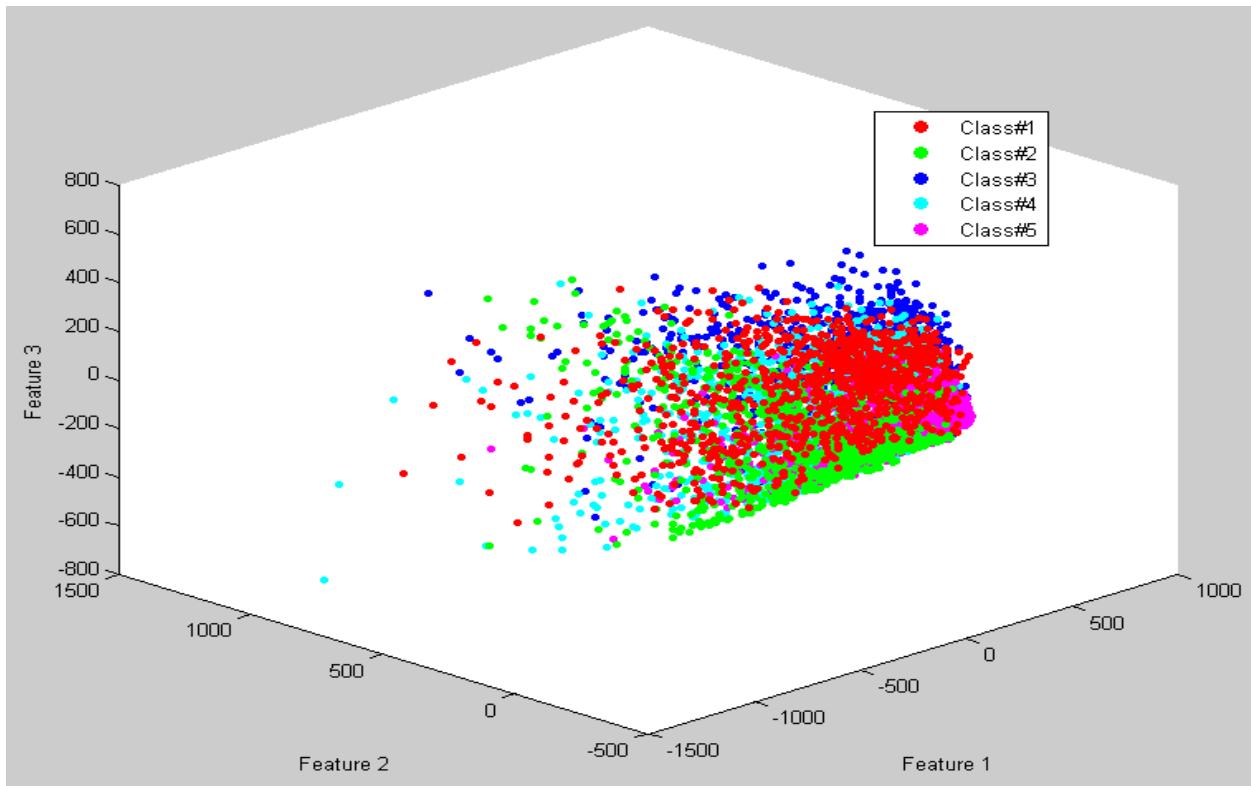
First dimension:



First 2 dimensions:



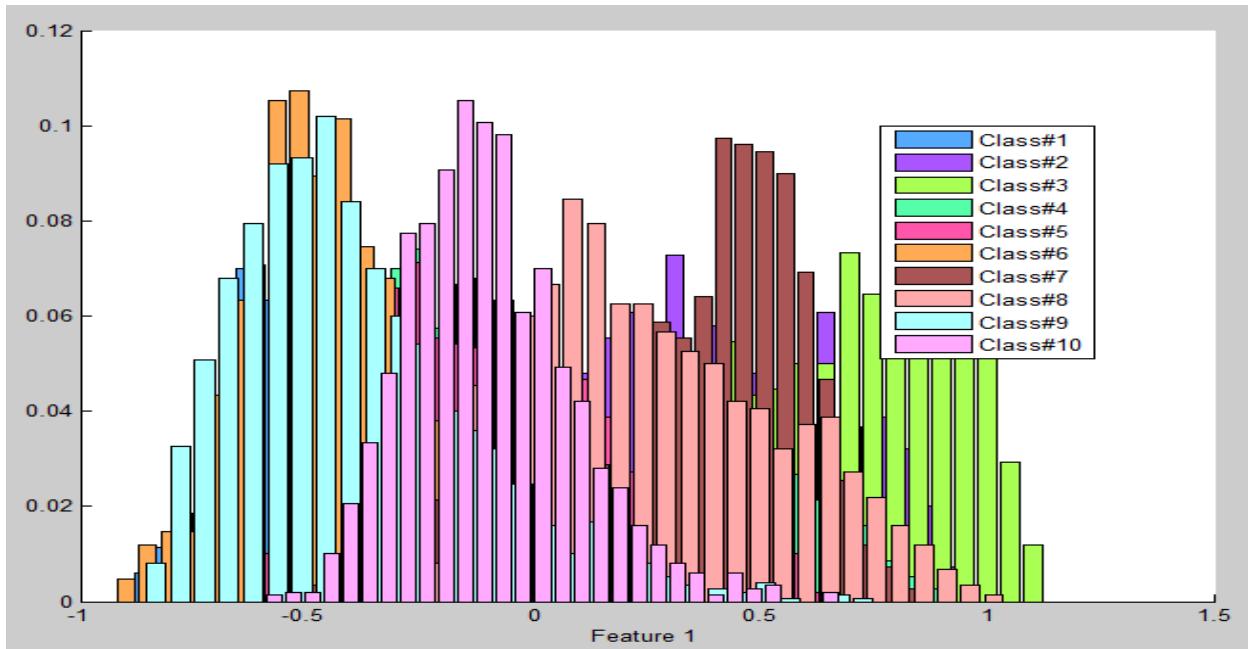
First 3 dimensions:



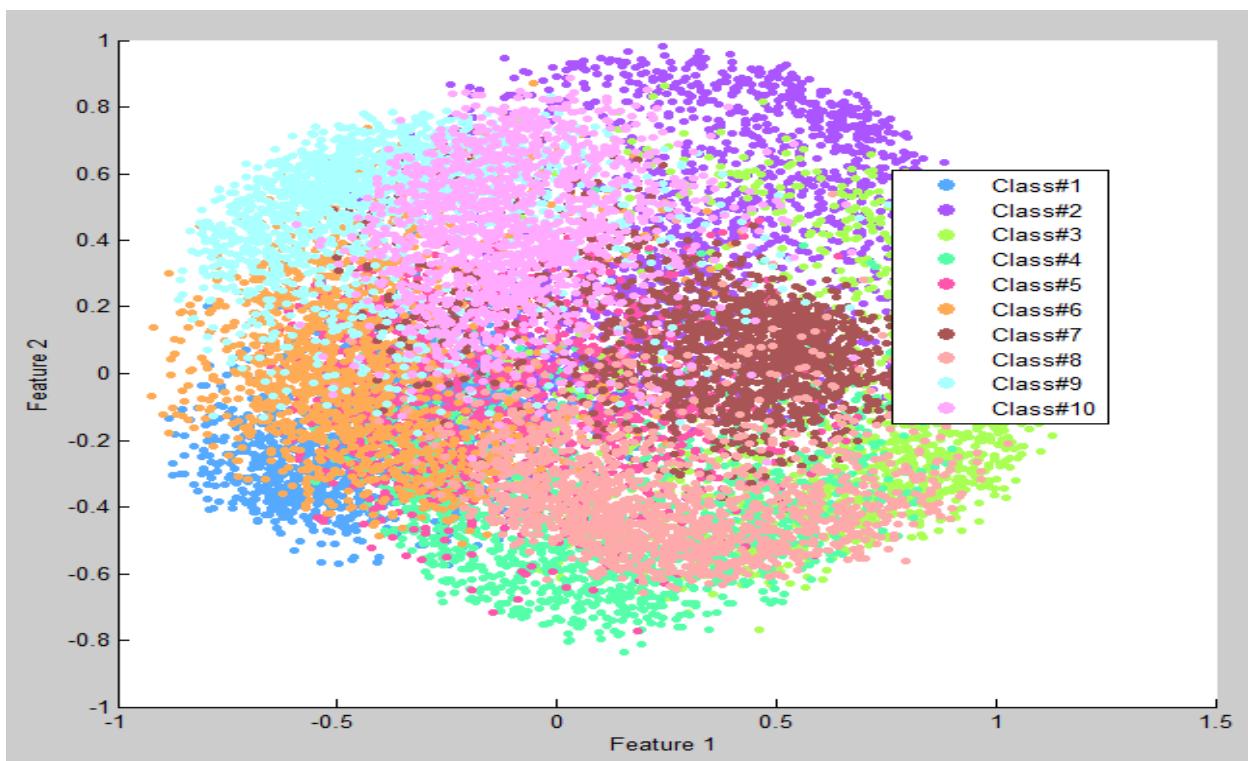
## Kernel PCA با هسته گوسی و پارامتر ۱۰۰۰

Scatter plots for second dataset

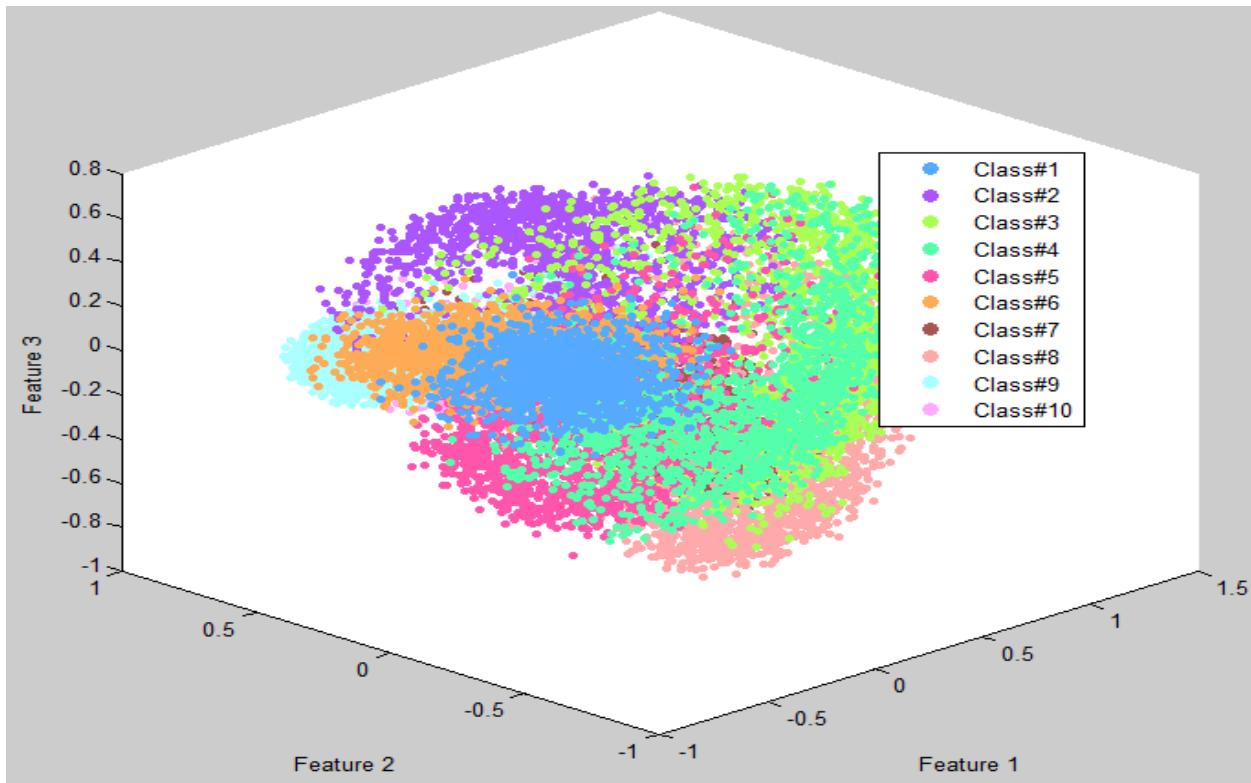
First 2 dimensions:



First 2 dimensions:

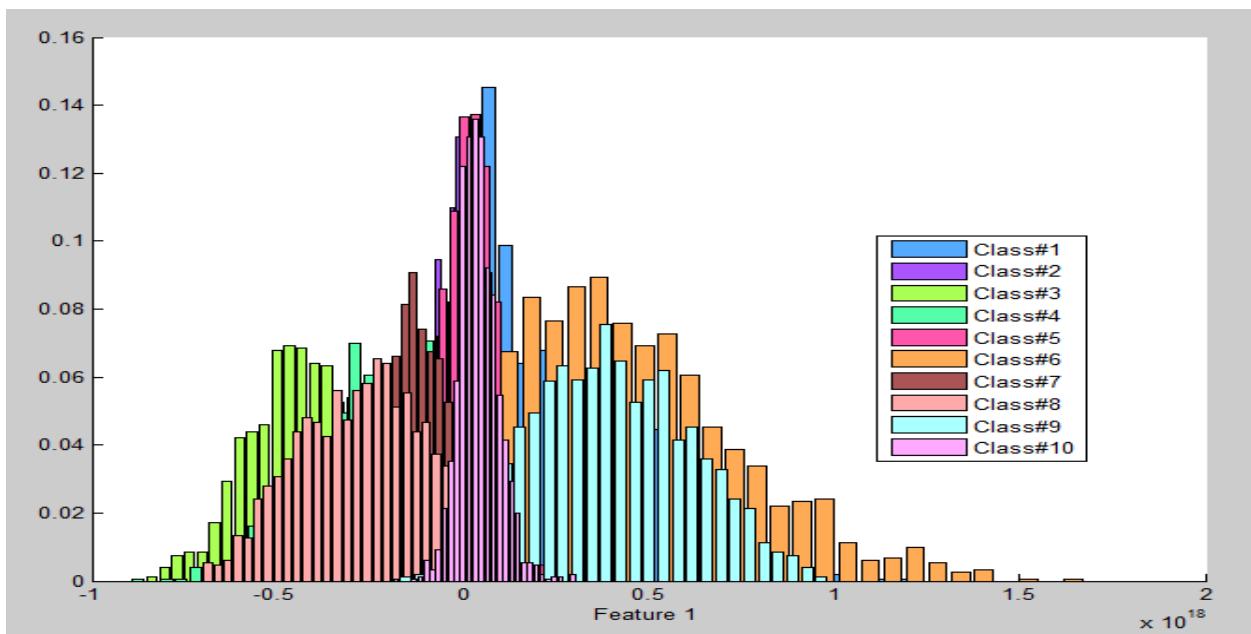


First 3 dimensions:

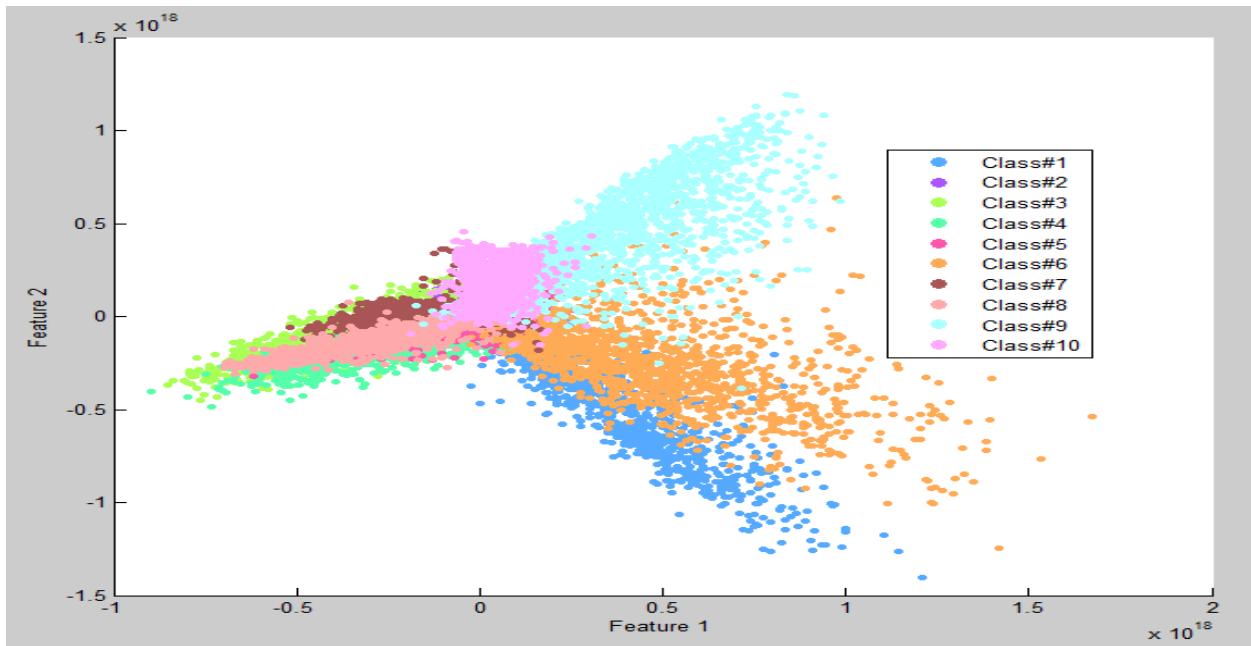


### با هسته چندجمله ای و پارامتر $\gamma$ Kernel PCA

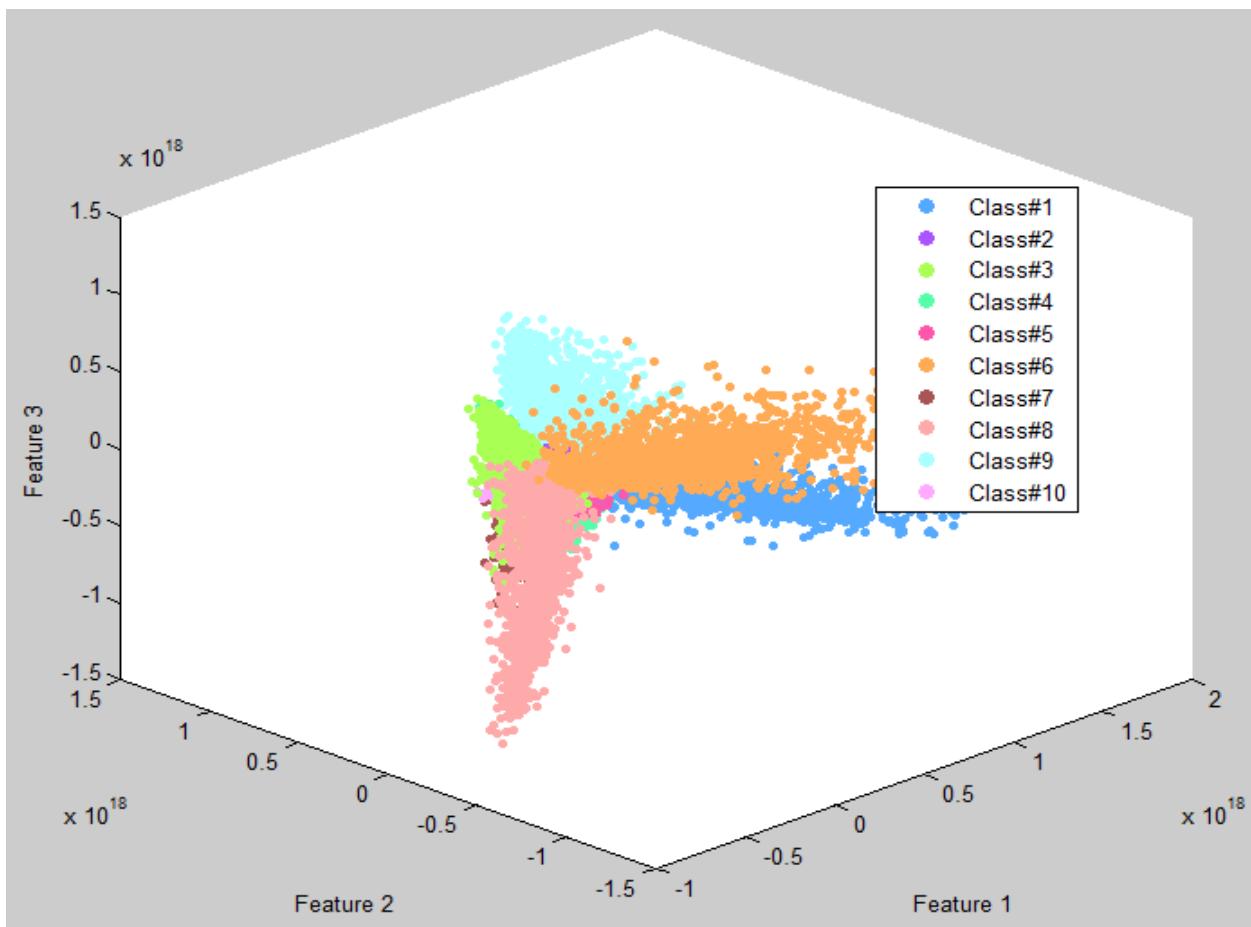
Scatter plots for first dimension:



First 2 dimensions:

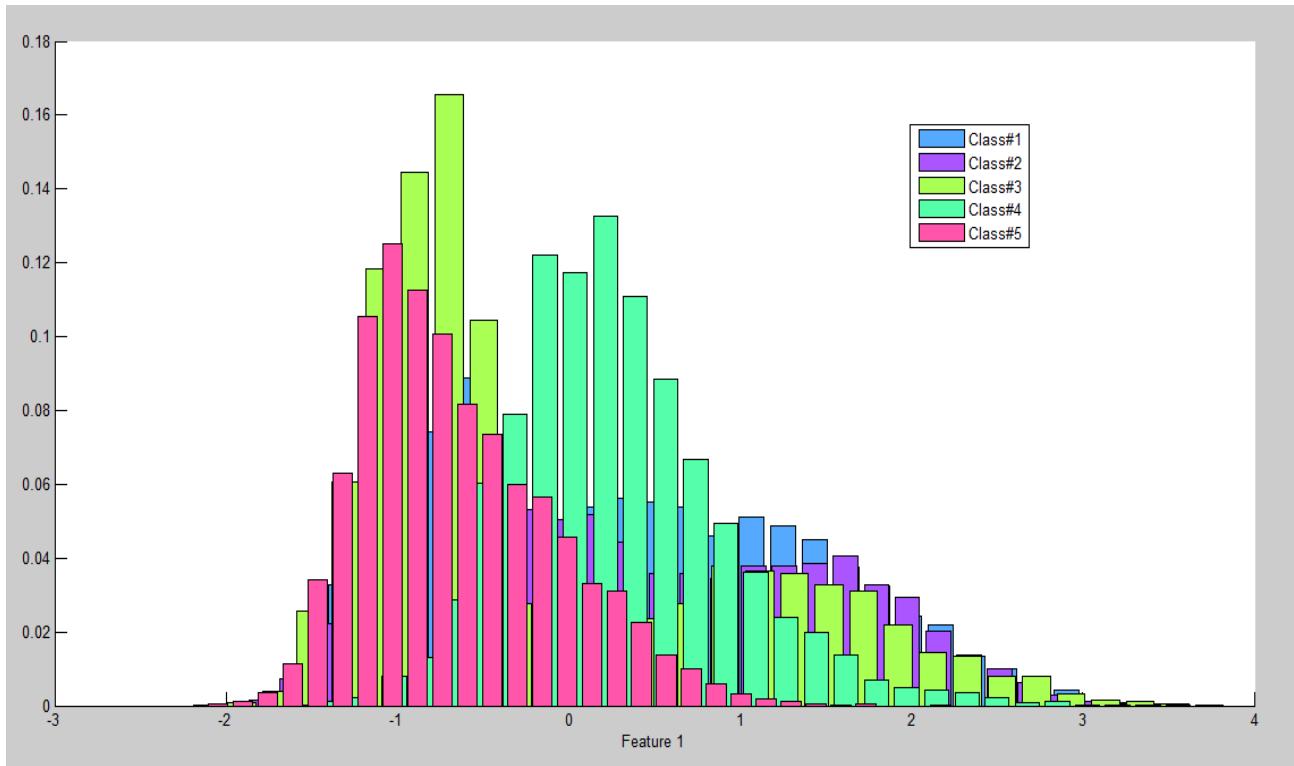


First 3 dimensions:

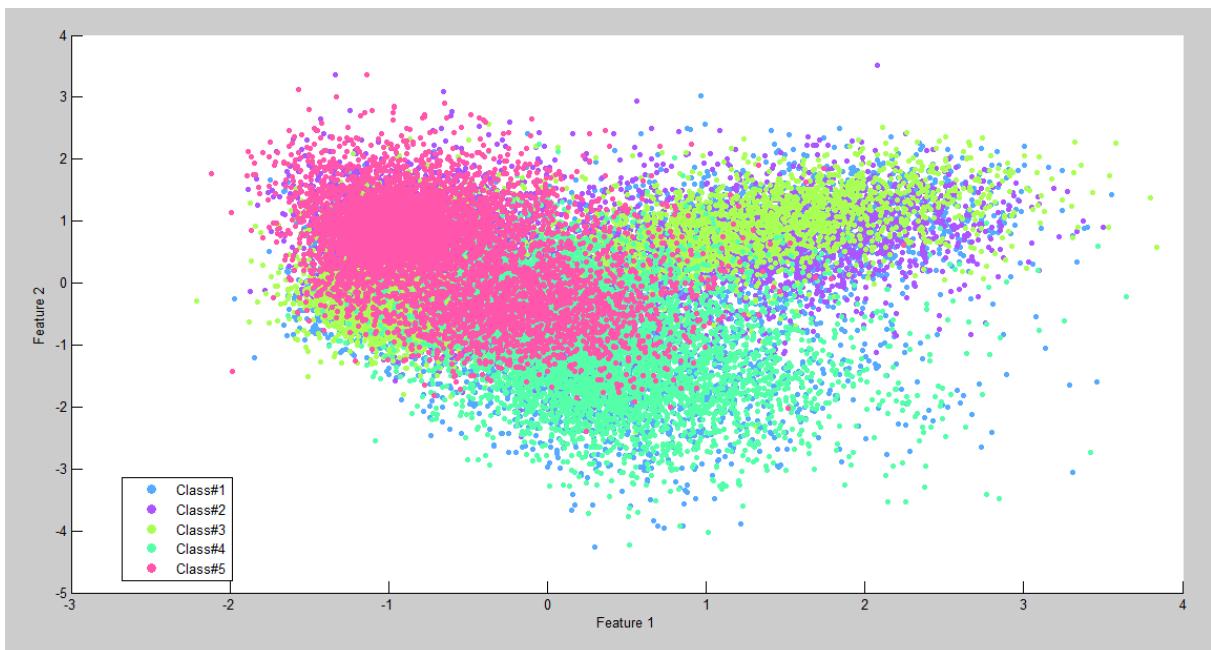


Scatter plots for first dataset:

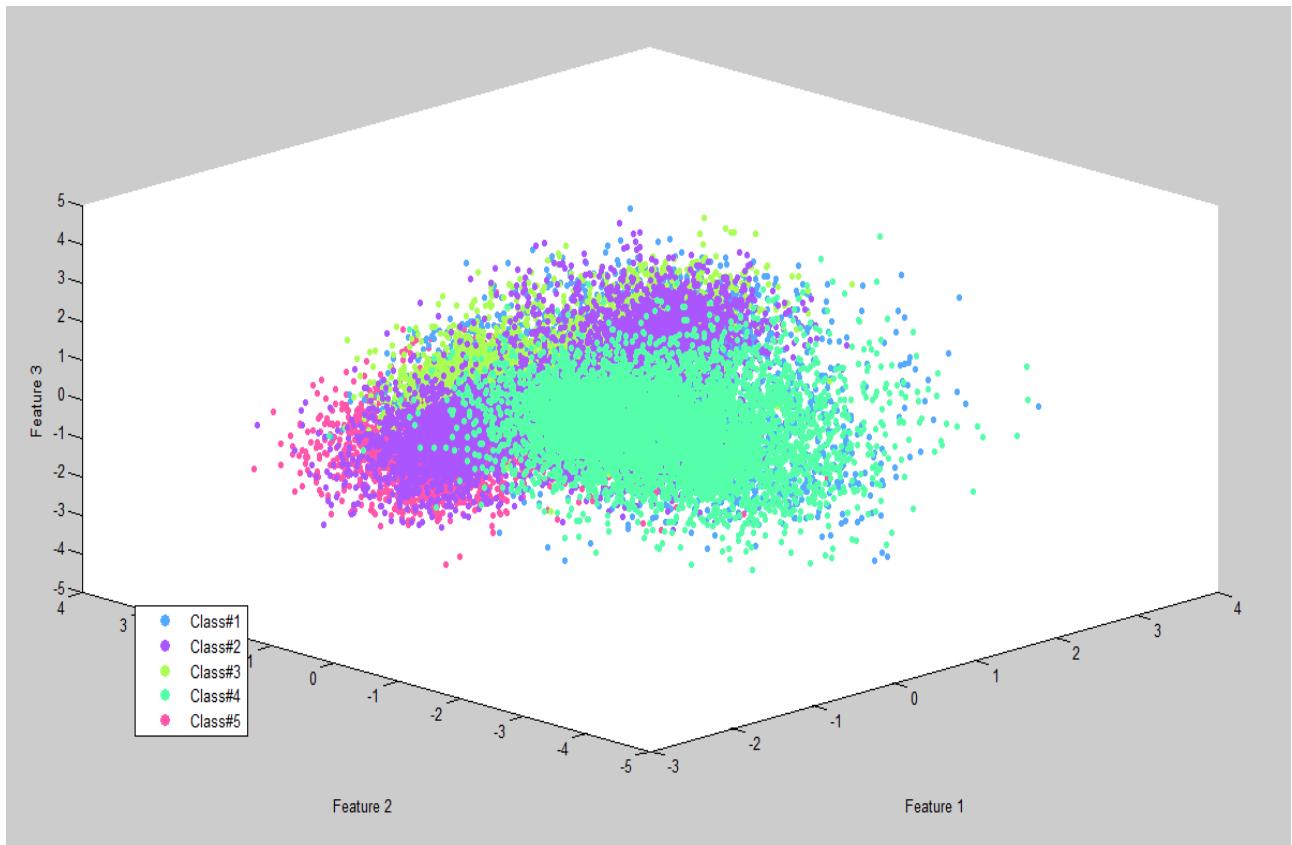
First dimension:



First 2 dimensions:



First 3 dimensions:



## مقایسه عملکرد روشها

برای این عملکرد این ۴ روش را با هم مقایسه کنیم از طبقه بند SVM استفاده میکنیم. هم چنین کرنل مورد استفاده در طبقه بند SVM را کرنل چند جمله ای با درجه ۳ در نظر میگیریم.

نتایج طبقه بندی برای هر یک از روشها در ادامه آورده شده است.

PCA •

### Results for first dataset:

Training time is 120.947162 seconds.

Accuracy = 65.24%

Test time is 2.122015 seconds.

### Results for second dataset:

Training time is 98.134865 seconds.

Accuracy = 70.56%

Test time is 2.946083 seconds.

LDA •

### Results for first dataset:

Train time is 119.500594 seconds.

Accuracy = 87.64%

Test time is 1.223289 seconds.

### Results for second dataset:

Train time is 40.702802 seconds.

Accuracy = 63.46%

Test time is 1.221044 seconds.

Kernel PCA •

کرنل گوسی با پارامتر ۱۰

### **Results for first dataset:**

Training time is 18.233527 seconds.

Accuracy = 61.76%

Test time is 2.388036 seconds.

کرنل گوسی با پارامتر ۱۰۰۰

### **Results for second dataset:**

Training time is 7.962477 seconds.

Accuracy = 68.66%

Test time is 3.159919 seconds.

LLE •

### **Results for first dataset:**

Training time is 174.739724 seconds.

Accuracy = 74.49%

Test time is 3.845731 seconds.

### **Results for second dataset:**

Training time is 143.452891 seconds.

Accuracy = 63.46%

Test time is 3.763123 seconds.

## نتایج

با توجه به نتایج طبقه بندی بر اساس SVM مشهود است که در مورد دسته داده اول روش کاهش بعد LDA بهتر از بقیه عمل میکند و روش KPCA نسبت به بقیه روشها از دقت کمتری برخوردار است. هم چنین روش LLE عملکرد خوبی روی داده دارد و علیرغم اینکه بعد داده ها را به ۴ کاهش دادیم، این روش CCR قابل توجه ای دارد و از آنجا که در این روش همسایگی داده ها حفظ میشود، میتوان گفت که روش LLE به خوبی میتواند با کاهش قابل توجه بعد بتواند همسایگیها را حفظ کند و CCR قابل ملاحظه ای در اختیار ما قرار بدهد.

هم چنین مشاهده میشود که علیرغم آسان بودن پیاده سازی روش PCA، این روش CCR بسیار خوبی میدهد. در مورد داده دسته دوم این روش بهترین میزان CCR را در اختیار ما قرار داد. لذا میتوان نتیجه گرفت که PCA در بسیاری از موارد میتواند بسیار به صرفه از لحاظ پیاده سازی باشد و در عین حال CCR قابل قبولی بدهد.

هم چنین مشاهده میشود که روش KPCA علیرغم اینکه از لحاظ پیاده سازی مشکل است و وقت گیر میباشد، با این وجود میزان CCR قابل توجه ای در اختیار ما نمیگذارد. البته ذکر این نکته واجب است که این روش بر روی داده های ما خوب عمل نکرد و ممکن است روی دیگر داده ها بهتر عمل کند.

هم چنین روش LDA هم علیرغم اینکه پیاده سازی راحت تری نسبت به دو روش KPCA و LLE دارد، میتواند CCR قابل توجه ای بدهد و میتوان گفت که از آنجا که پیاده سازی راحتی دارد میتواند گزینه مناسبی برای کاهش بعد در هنگامیکه هیچ ایده ای راجع به داده ها نداریم، باشد.

در آخر میتوان نتیجه گیری کلی را اینگونه گفت که دو روش PCA و LDA هنگامیکه میخواهیم با صرف کمترین وقت و هزینه ابعاد داده را کاهش دهیم، بهترین گزینه ها هستند زیرا که همانطور که از نتایج مشهود است این دو روش CCR قابل قبولی در اختیار ما میگذارند و هم چنین میتوانند نشان دهند که تا چه حد میتوانیم ابعاد داده ها را کاهش دهیم تا کاهش قابل توجه CCR نداشته باشیم.

## مراجع

- 1- Lawrence K.Saul, Sam T.Roweis. An Introduction to Locally Linear Embedding.
- 2- Bernhard Scholkopf, Alexander Smola, Klaus-Robert Muller. Kernel Principal Component Analysis.