Lan Fangzhou
Pattern-Recognizing
Assignment-2

# 03-Framework.pdf.1, 04-Error.pdf.2, 04-Error.pdf.5, 04-Error.pdf.7, 05-PCA.1, 05-PCA.2, 05-PCA.3

## 03-Framework.pdf.1

# 03Framework.1

April 20, 2018

```
In [10]: function [res] = NN(data)
           tic
           n = size(data)(1); m = size(data)(2);
           res = zeros(1,n);
           for i = 1:n
             dist = sum(((data - data(i,:)).*(data - data(i,:)))');
             dist(i) = realmax;
             [mini, res(i)] = min(dist);
           end
           toc
         end

In [11]: x = rand(5000, 10);

In [12]: tic
         res1 = NN(x)
         toc

Elapsed time is 3.28171 seconds.
res1 =

 Columns 1 through 11:

   4566   2526   1862   1366   3082   4625   1791   2433   3549   1687   4092

 Columns 12 through 22:

   3089   3688   2906   1708   2248   4976   2211   4509   3580   4897   4638


Elapsed time is 4.04794 seconds.


In [17]: addpath /home/lanfangzhou/Downloads/Softwares/vlfeat-0.9.21/toolbox

In [16]: pkg list

Package Name  | Version | Installation directory
--------------+---------+-----------------------
```

```
     control  |   3.0.0 | /usr/share/octave/packages/control-3.0.0
     general  |   2.0.0 | /usr/share/octave/packages/general-2.0.0
       image *|   2.6.1 | /usr/share/octave/packages/image-2.6.1
```

```
In [18]: vl_setup

In [30]: vl_version verbose

error: invalid use of script /home/lanfangzhou/Downloads/Softwares/vlfeat-0.9.21/toolbox/misc/vl
```

Since we cannot get octave with version 3.x to install now because of lots of problems, and octave4 is not compatible with vlfeat, the following tasks are impractical for me.

■

**04-Error.pdf.2**

(a)

$$\text{find } \beta \in R^d \text{ to minimize } \sum_{i=1}^{n}(y_i - x_i^T\beta)^2$$

(b)

$$\text{find } \beta \in R^d \text{ to minimize } (y - X\beta)^T(y - X\beta)$$

(c)

We first calculate the derivative of the objective

$$\frac{d((y - X\beta)^T(y - X\beta))}{d\beta} = -2X^T(y - X\beta)$$

Let it = 0 :

$$\beta = (X^TX)^{-1}X^Ty$$

Which is the optimal value.

(d)

No. When $d > n$, $rank(X^TX) \leq rank(X) = d < n = size(X^TX)$

(e)

The regularizer will decrease the $norm_2$ of $\beta$, i.e, decrease the magnitude of $\beta$.

(f)

find $\beta \in R^d$ to minimize $(y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta$
We first calculate the derivative of the objective

$$\frac{d((y - X\beta)^T(y - X\beta)) + \beta^T\beta}{d\beta} = -2X^T(y - X\beta) + 2\lambda\beta$$

Let it = 0 :

$\beta = (X^TX + \lambda I)^{-1}X^Ty$ (if $X^TX + \lambda I$ is invertible)

(g)
The parameter $\lambda$ will make the matrix more likely to be invertible, and definitely invertible when $\lambda \to +\infty$.

(h)
No. For the definition of the cost function, the optimal value for $\lambda$ is when $\lambda \to 0$, which is not what we want. ■

**04-Error.pdf.5**

(a)(b)
AUC-PR: $(r_i - r_{i-1})\frac{p_i+p_{i-1}}{2}$
AP: $(r_i - r_{i-1})p_i$

| index | label | score | precision | recall | AUC-PR | AP |
|---|---|---|---|---|---|---|
| 0 | | | 1.0000 | 0.0000 | - | - |
| 1 | 1 | 1.0 | 1.0000 | 0.2000 | 0.2000 | 0.2000 |
| 2 | 2 | 0.9 | 0.5000 | 0.2000 | 0.0000 | 0.0000 |
| 3 | 1 | 0.8 | 0.6667 | 0.4000 | 0.1167 | 0.1667 |
| 4 | 1 | 0.7 | 0.7500 | 0.6000 | 0.1417 | 0.1333 |
| 5 | 2 | 0.6 | 0.6000 | 0.6000 | 0.0000 | 0.0000 |
| 6 | 1 | 0.5 | 0.6667 | 0.8000 | 0.1267 | 0.1333 |
| 7 | 2 | 0.4 | 0.5714 | 0.8000 | 0.0000 | 0.0000 |
| 8 | 2 | 0.3 | 0.5000 | 0.8000 | 0.0000 | 0.0000 |
| 9 | 1 | 0.2 | 0.5556 | 1.0000 | 0.1111 | 0.1056 |
| 10 | 2 | 0.1 | 0.5000 | 1.0000 | 0.0000 | 0.0000 |
| | | | | | 0.6962 | 0.7389 |

(c)
newAUC: 0.6907
newAP: 0.7333

(d)

In [34]:

```
function x = AUC(label, score)
    [tmp, pos] = sort(score, 'descend');
    label = label(pos);
    n = length(score);
    precision = zeros(1, n); recall = zeros(1, n);
    x = 0.0;
    TP = 0;
    P = length(find(label == 1));
    for i = 1:n
        if label(i) == 1
            TP++;
        end
        precision(i) = TP / i;
        if P == 0
            recall(i) = 0
        else
            recall(i) = TP / P;
        end
    end

    for i = 1:n
        if (i > 1)
            x += (recall(i) - recall(i-1)) * (precision(i) + precision(i-1)) / 2;
        else
            x += (recall(i) - 0) * (precision(i) + 1.0) / 2;
        end
    end
end
```

executed in 66ms, finished 23:52:13 2018-04-20

In [35]:

```
label = [1,2,1,1,2,1,2,2,1,2];
score = 1.0:-0.1:0.1;
AUC(label, score)
```

executed in 16ms, finished 23:52:15 2018-04-20

```
ans =  0.69056
```

■

## 04-Error.pdf.7

(a)

$$p(x,y) = \begin{cases} 0.5 \cdot \frac{1}{\sqrt{2\pi} \cdot 0.5} e^{-\frac{(x+1)^2}{0.5^2}}, & \text{y=1} \\ 0.5 \cdot \frac{1}{\sqrt{2\pi} \cdot 0.5} e^{-\frac{(x-1)^2}{0.5^2}}, & \text{y=2} \end{cases}$$

(b)

1°
One solution is:

$$\begin{aligned} f(x) &= argmax_y p(y|x) \\ &= argmax_y \frac{p(x|y)p(y)}{p(x)} \\ &= argmax_y p(x|y) \end{aligned}$$

The cost:

$$
\begin{aligned}
E_{(x,y)}[c_{y,f(x)}] &= 0.5E_x[c_{1,f(x)}] + 0.5E_x[c_{2,f(x)}] \\
&= 0.5(Pr(f(x) = 2|y = 1) + Pr(f(x) = 1|y = 2)) \\
&= 0.5(\int_{f(x)=2} p(x|y = 1) + \int_{f(x)\neq 2} p(x|y = 2)) \\
&= 0.5(\int_{-\infty}^{+\infty} min(p(x|y = 1), p(x|y = 2)) \\
&\geq 0.5(Pr(x \geq 0|y = 1) + Pr(x \leq 0|y = 2)) \\
&\text{the minimum is obtained when f(x) is set as above.} \\
&\text{Under this solution, the cost is:} \\
&= 0.5(Pr(x \geq 0|y = 1) + Pr(x \leq 0|y = 2)) \\
&= 2(1 - \Phi(2)) \ (\Phi \text{ is cdf of normal distribution.}) \\
&\approx 2(1 - 0.97725) \\
&= 0.045500
\end{aligned}
$$

$2^\circ$
In a multi-class classification problem, also true.

(c)
Let $y = f(x) = argmax_y p(y|x)$
Bayes risk: 0.045500

(d)

$$
\begin{aligned}
E_{(x,y)}[c_{y,f(x)}] &= 0.5E_x[c_{1,f(x)}] + 0.5E_x[c_{2,f(x)}] \\
&= 0.5(Pr(f(x) = 2|y = 1) + 10Pr(f(x) = 1|y = 2)) \\
&= 0.5(\int_{-\infty}^{+\infty} min(p(x|y = 1), 10p(x|y = 2))
\end{aligned}
$$

Thus

$$
f(x) = \begin{cases} 1, & p(x|y = 1) \geq 10p(x|y = 2) \\ 2, & p(x|y = 1) < 10p(x|y = 2) \end{cases}
$$

■

**05-PCA.1**

(a)
Since $U, V$ are orthogonal:

$XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma(V^T V)\Sigma^T U^T = U(\Sigma\Sigma^T)U^T$
i.e. $(XX^T)U = (\Sigma\Sigma^T)U$
Therefore, the eigenvalues of $XX^T$ are $\sigma_1^2, \sigma_2^2, ..., \sigma_{min(m,n)}^2, 0, ..., 0$ (The number of following 0 is $max(0, m-n)$), while corresponding eigenvectors are $u_1, u_2, ..., u_m$ $(U = (u_1, u_2, ..., u_m))$

(b)
Since $U, V$ are orthogonal:
$X^T X = (U\Sigma V^T)^T(U\Sigma V^T) = V\Sigma^T(U^T U)\Sigma V^T = V(\Sigma^T\Sigma)V^T$
i.e. $(X^T X)V = (\Sigma^T\Sigma)V$
Therefore, the eigenvalues of $X^T X$ are $\sigma_1^2, \sigma_2^2, ..., \sigma_{min(m,n)}^2, 0, ..., 0$ (The number of following 0 is $max(0, m-n)$), while corresponding eigenvectors are $v_1, v_2, ..., v_n$ $(V = (v_1, v_2, ..., v_m))$

(c)
Equivalent except for number of 0

(d)
Eigenvalues of $XX^T(X^T X)$ are square of singular value of $X$ except for number of 0.

(e)
By calculating the eigenvalues of $XX_T$ (which are almost equivalent). ■

**05-PCA.2**

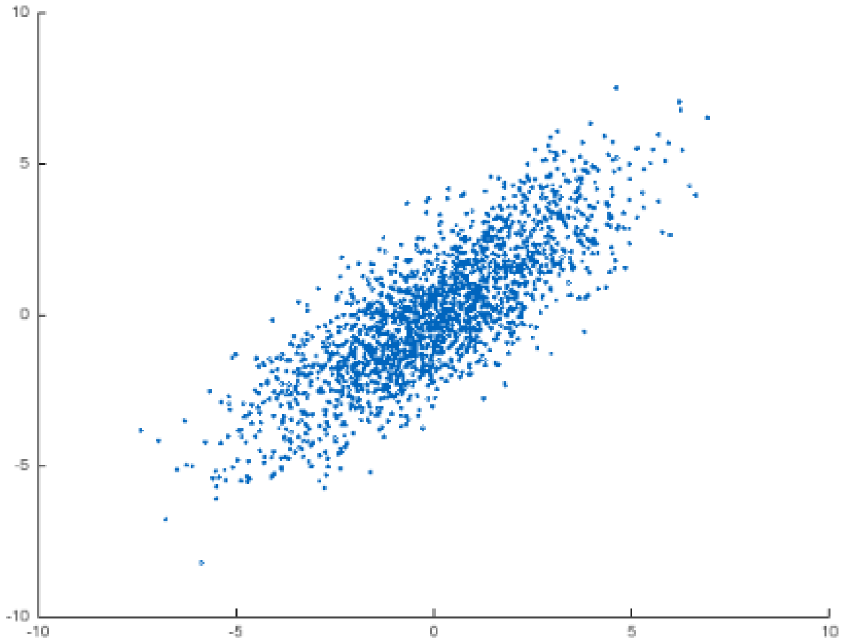
When scale is relatively big, yes. When $scale \geq 0.1$, corr1 usually is more than 0.99.
When scale is relatively small, no.
The second one is correct. ■

**05-PCA.3**

In [75]:

```
#(a)
x = randn(2000, 2) * [2 1;1 2];
scatter(x(:,1), x(:,2))
```

executed in 270ms, finished 15:43:26 2018-04-21




n [530]:

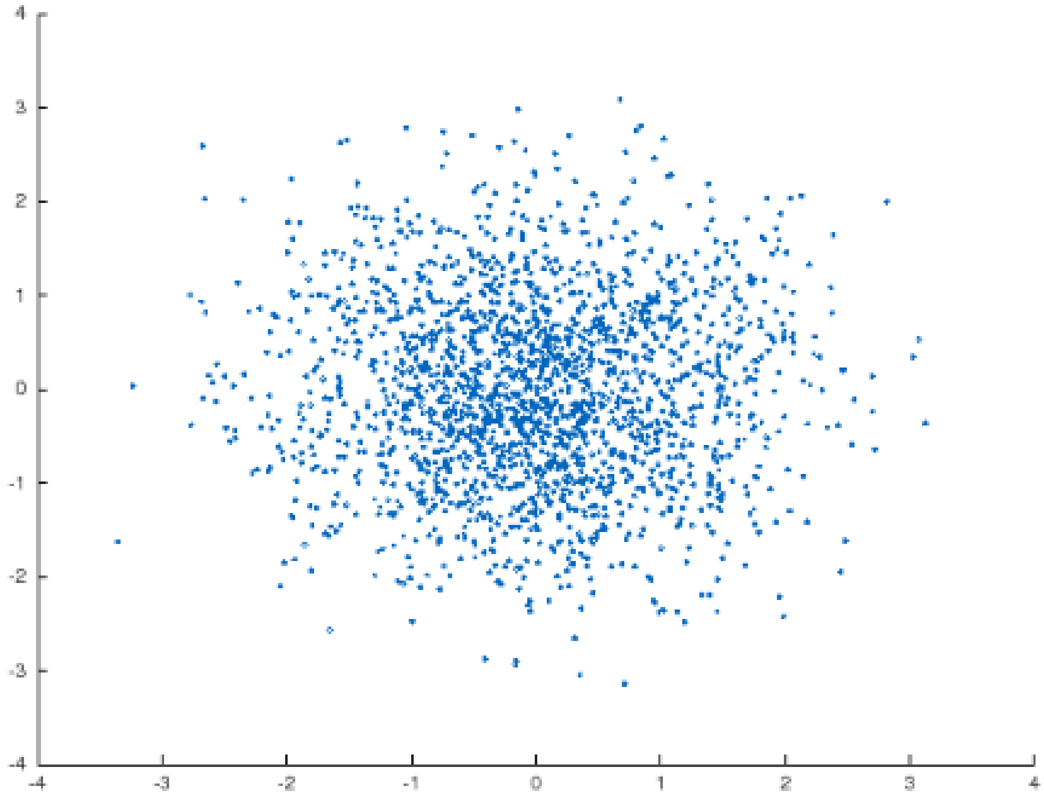
```
#(b)(c)
function [coeff, score, X_bar, L_ret, explained] =  PCA(X, d = 1, whiten = false)
    N = size(X)(1); D = size(X)(2);
    L_ret = ones(1,d);
    X_bar = mean(X);
    X_center = X - X_bar;
    Cov = (X_center' * X_center) / N;
    [V, L] = eig(Cov);
    [L_sort, pos] = sort(diag(L), 'descend');
    V = V(:,pos); L = diag(L_sort);
    if whiten == true
        V = V * diag(1./(sqrt(L_sort) + 1e-12));
        L_ret = L_sort(1:d);
    end
    V = V(:,1:d);
    coeff = V;
    score = X_center * coeff;
    explained = L_sort(1:d) / sum(L_sort);
end
```

executed in 47ms, finished 20:15:37 2018-04-21

In [536]:

```
#With whitening
[coeff,score,x_bar,L,explained]=PCA(x,2,true);
scatter(score(:,1), score(:,2))
```

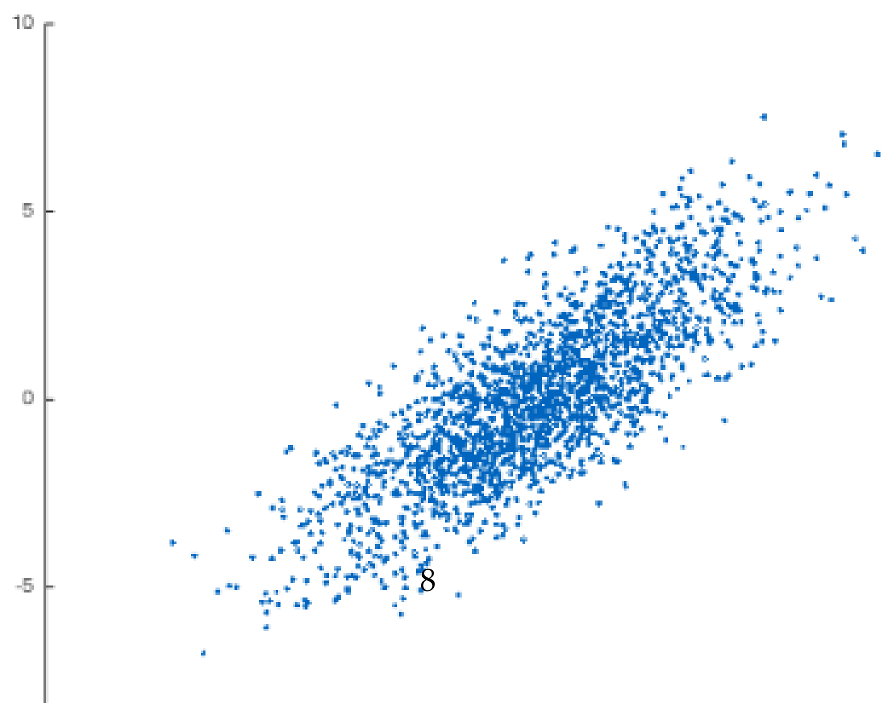
executed in 246ms, finished 20:16:50 2018-04-21




In [537]:

```
#With whitening
[coeff,score,x_bar,L,explained]=PCA(x,2,true);
x_new = x_bar + score * diag(L) * coeff';
scatter(x_new(:,1),x_new(:,2))
```

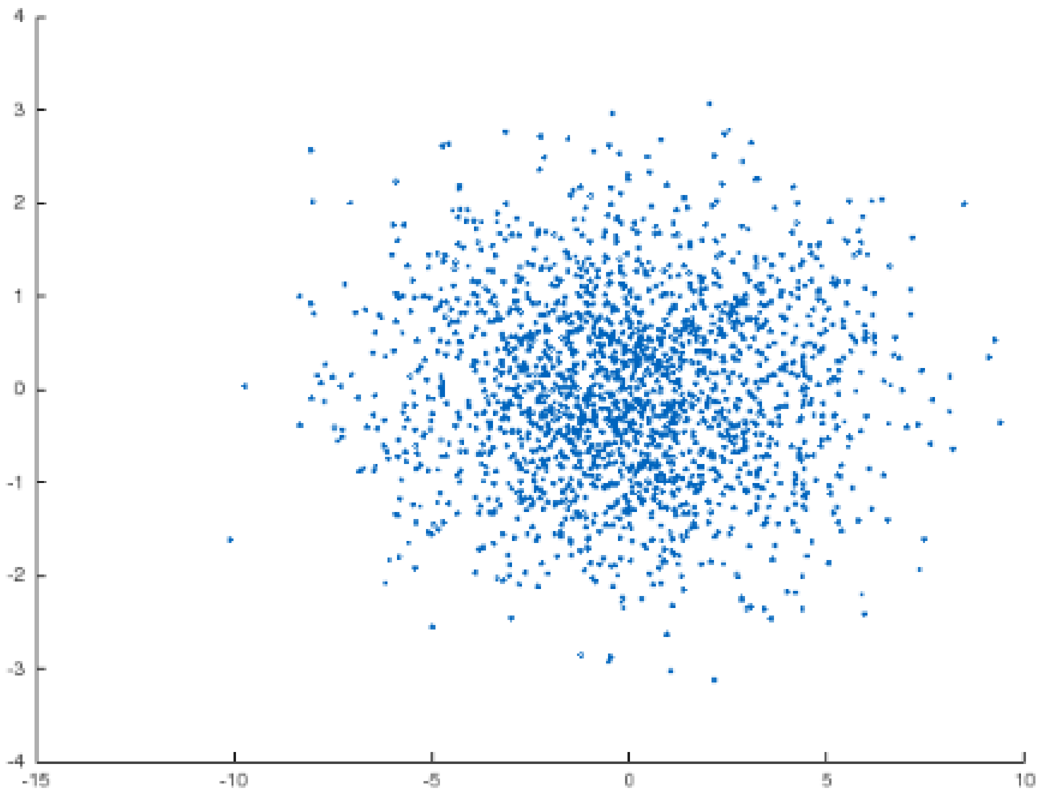
executed in 248ms, finished 20:16:51 2018-04-21



In [538]:

```
#Without whitening
[coeff,score,x_bar,L,explained]=PCA(x,2,false);
scatter(score(:,1), score(:,2))
```

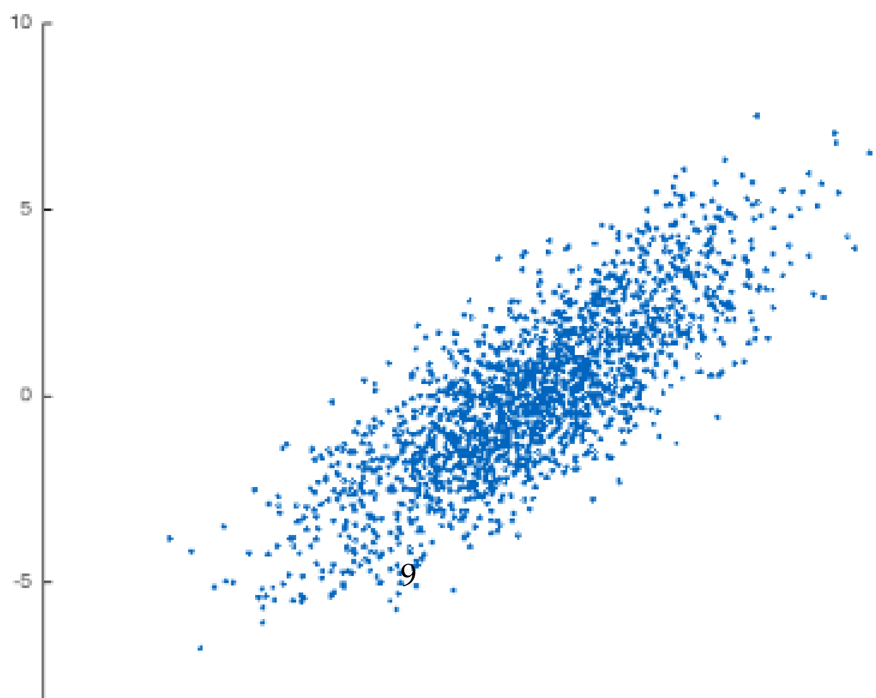
executed in 274ms, finished 20:16:56 2018-04-21




In [540]:

```
#Without whitening
[coeff,score,x_bar,L,explained]=PCA(x,2,false);
x_new = x_bar + score * diag(L) * coeff';
scatter(x_new(:,1),x_new(:,2))
```

executed in 254ms, finished 20:18:05 2018-04-21



In [542]:

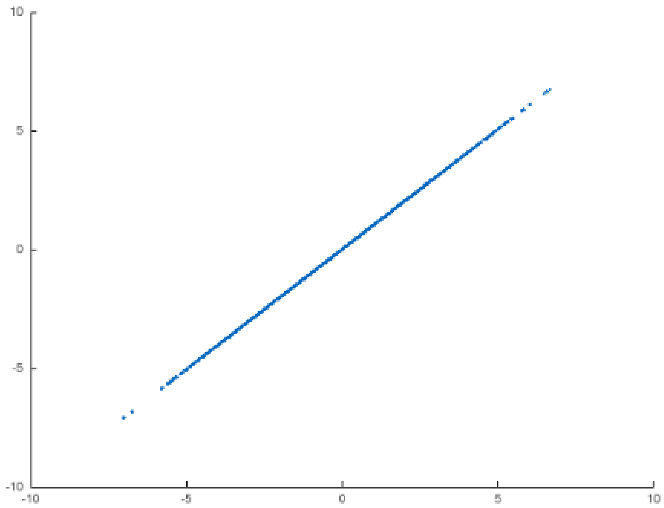
```
#(d)
#The dimentions are changed to other orthogonal dimensions in terms of decreasing variance.
#The operation is useful because we can use fewer dimensions to describe almost all the data patterns.
##Just like below, we use PCA to decrease 2 dim into 1 dim.
```

executed in 35ms, finished 20:21:56 2018-04-21

In [543]:

```
[coeff,score,x_bar,L,explained]=PCA(x,1,false);
x_new = x_bar + score * diag(L) * coeff';
scatter(x_new(:,1),x_new(:,2))
```

executed in 326ms, finished 20:21:58 2018-04-21




■