

# ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААНЫ ИХ СУРГУУЛЬ

АХИСАН ТҮВШИНИЙ ТООЦОН БОДОХ

---

## Principal component analysis

---

*Бэлтгэсэн:*

Э.Мөнхбаатар

*Хянасан:*

Док. А.ЭНХБАЯР

2017 оны 5-р сарын 31



# 1 Танилцуулга

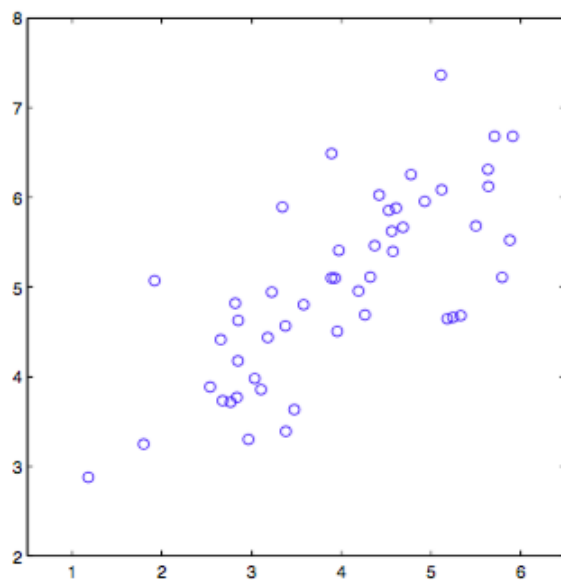
Машин сургалт маш олон ( $n > 100000$  хэмжээст) утгаас хамаарах тохиолдолд сургахад зарцуулах цаг шууд хэдэн хэмжээст байгаагаас нэмэгддэг. Энэ тохиолдолд хугацааг буруулах зорилгоор өгөгдлөөс резонансийг хасах зорилготойгоор Principal component analysis(PCA) алгоритмийг хэрэглэнэ.

## 2 Principal component analysis

### 2.1 Жишээ өгөгдөл

Principal component analysis(PCA) хэрхэн ажиллаж байгааг харуулах үүднээс жишээ 2 хэмжээст өгөгдөл дээр алгоритмыг хэрхэн ажиллаж байгааг харуулая. Энэхүү жишээ дээр 2 хэмжээст (Зураг 1) өгөгдлийг 1 хэмжээст рүү шилжүүлнэ.

Principal component analysis(PCA) - ыг ашиглан 250 хэмжээстийг 50 хэмжээст гэх мэтчилэн янз бүрийн хэмжээст руу буулгах боломтой бөгөөд жишээ дээр дүрслэхэд илүү амар учраас 2 хэмжээстийг сонгон авсан байгаа .



Зураг 1: 2 хэмжээст өгөгдөл.

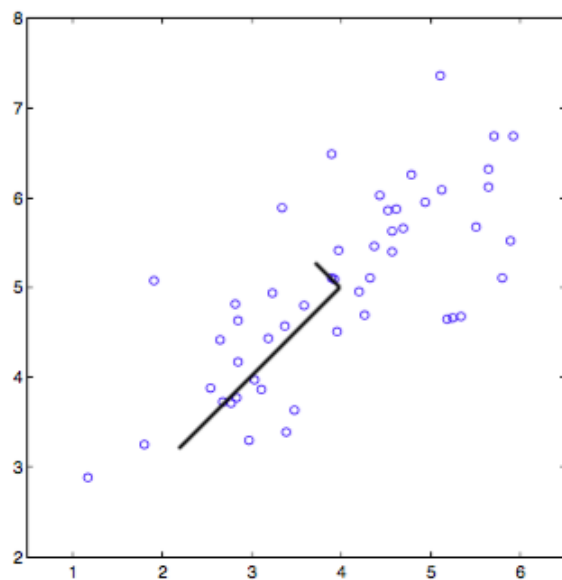
## 2.2 Алгоритмт хэрэгжүүлэлт

Principal component analysis(PCA) хэрэгжүүлэлт нь 2 алхамаас тогтох бөгөөд эхний алхам нь дундаж матриц-г олно. Дараа нь Octave/MATLAB - ийн SVD функцын тусламжтайгаар eigenvector  $U_1, U_2, \dots, U_n$  - ийг олно. Matlab код:

```

1 function [U, S] = pca(X)
2     [m, n] = size(X);
3     U = zeros(n);
4     S = zeros(n);
5     Sigma = 1.0/m .* X' * X;
6     [U, S, V] = svd(Sigma);
7 end

```



Зураг 2: олсон eigenvector - ууд.

SVD - ийн тусламжтайгаар олдсон eigenvector - уудыг зураг 2 дээр дүрслэн үзүүлсэн байгаа.

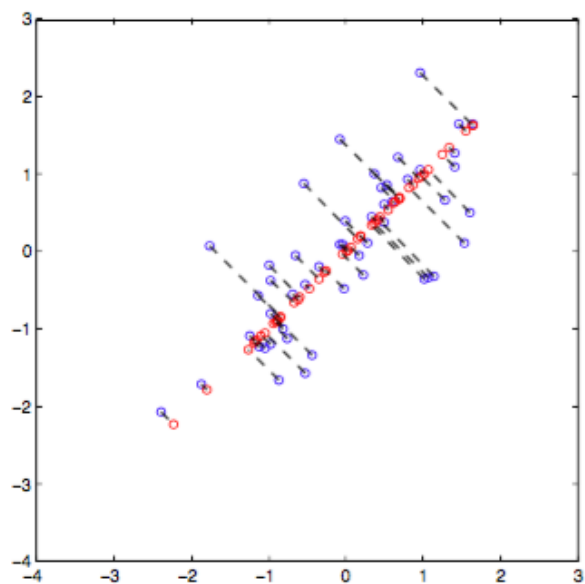
## 2.3 Хэмжээс бууруулах

Principal component analysis(PCA) ажиллуулсаны дараа олсон eigenvector - уудын тусламжтайгаар хэмжээсийг бууруулна.

X бол бидний боловсруулалт хийж буй өгөгдөл.

U бол бидний олсон eigenvector - ууд юм.

K бол бидний хэдэн хэмжээст рүү буулгахыг хүсч байгаа хэмжээст (2,3,4 ... n)



Зураг 3: 1 хэмжээт руу буулгасан байдал.

Matlab код:

```

1 function Z = projectData(X, U, K)
2     Z = zeros(size(X, 1), K);
3     for i=1:size(X, 1),
4         for j=1:K,
5             x = X(i, :)';
6             projection_k = x' * U(:, j);
7             Z(i, j) = projection_k;
8         end
9     end
10 end

```

## 2.4 Дата буцааж сэргээх

eigenvector - ийн тусламжтайгаар анхны хэмжээст рүү буцаан шилжүүлэх

Z бол бууруулсан хэмжээст өгөгдөл U бол бидний олсон eigenvector - ууд юм.

K бол бидний хэдэн хэмжээст рүү буулгахыг хүсч байгаа хэмжээст (2,3,4 ... n)

Matlab код:

```
1    function X_rec = recoverData(Z, U, K)
2    X_rec = zeros(size(Z, 1), size(U, 1));
3    for i=1:size(Z, 1),
4        for j=1:size(U,1),
5            v = Z(i, :)';
6            recovered_j = v' * U(j, 1:K)';
7            X_rec(i, j) = recovered_j;
8        end
9    end
10 end
```

Original numbers

0	1	2	3
4	5	6	7
8	9		

0	1	2	3
4	5	6	7
8	9		

Зураг 4: Харьцуулалт.

## 2.5 Дүгнэлт

Дээр дурдсан бүх алгоритмийг 50x50 хэмжээтэй 0 ... 9 хүртэлх тооны зураг дээр туршилт хийсэн нь зураг 4 дээр харуулсан байна. Энэ туршилт дээр Principal component analysis(PCA) ашиглан 50x50 буюу 2500 хэмжээст зургийг 8 хэмжээст руу буулгасан байгаа. Энэ тохиолдолд зураг дээрээс тоог ялган харах боломжтой харагдаж байгаа бөгөөд машин сургалтыг ажиллуулахад таамаглалд нөлөөлөх нь бага харагдаж байна. Йимээс машин сургахад илүү хурдан ажиллах хэдий ч үр дүнд нөлөөлөх нь бага байна.

Source code:

<https://github.com/HeroPD/pca>