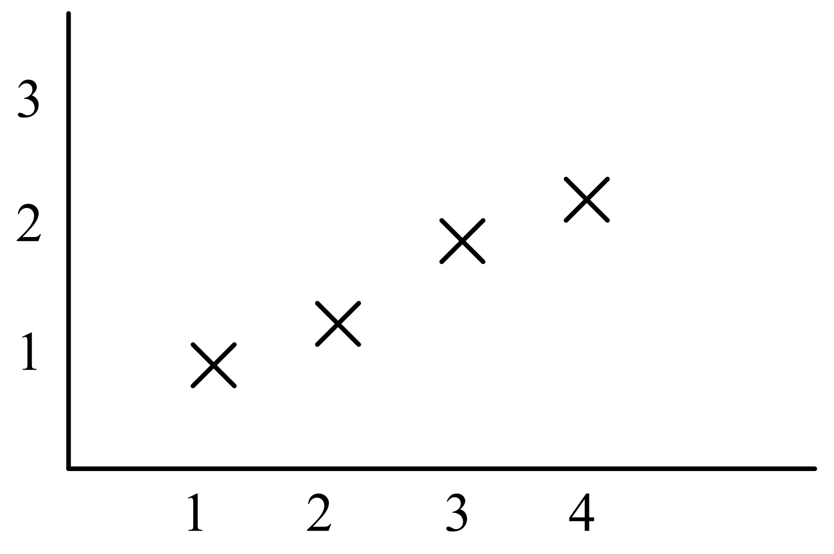
Dimensionality reduction

Usman Roshan

Dimensionality reduction

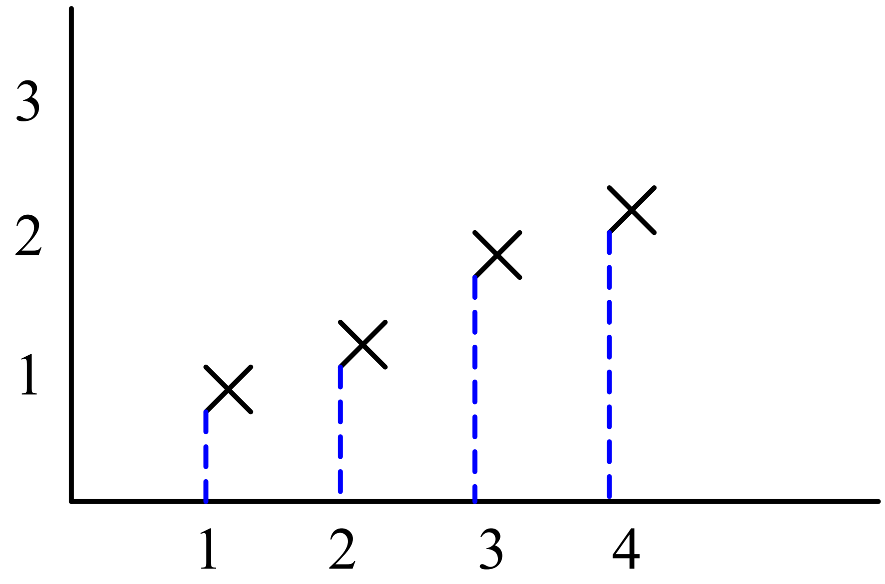
- What is dimensionality reduction?
  - Compress high dimensional data into lower dimensions
- How do we achieve this?
  - PCA (unsupervised): We find a vector w of length 1 such that the variance of the projected data onto w is maximized.
  - Binary classification (supervised): Find a vector w that maximizes ratio (Fisher) or difference (MMC) of means and variances of the two classes.

Data projection



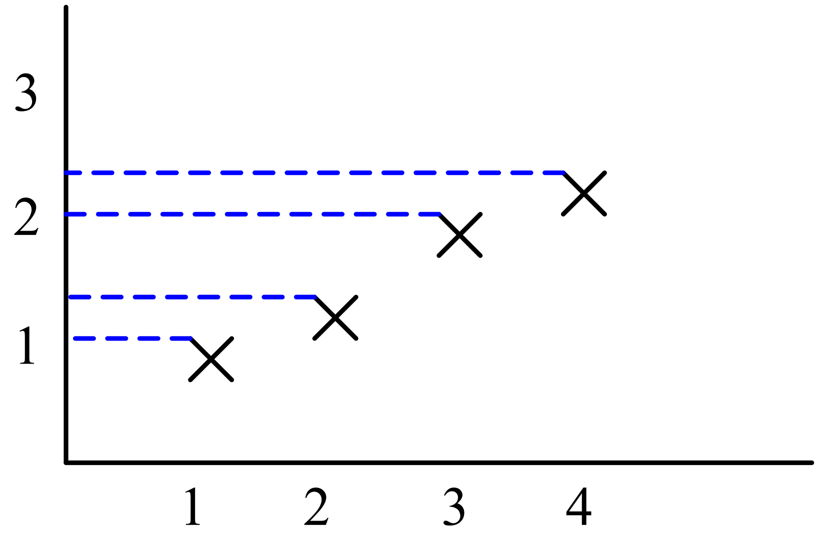
Data projection

- Projection on x-axis



Data projection

- Projection on y-axis



Mean and variance of data

- Original data

Projected data

$$Mean: m = \frac{1}{n}\sum_{i=1}^{n} x_i$$

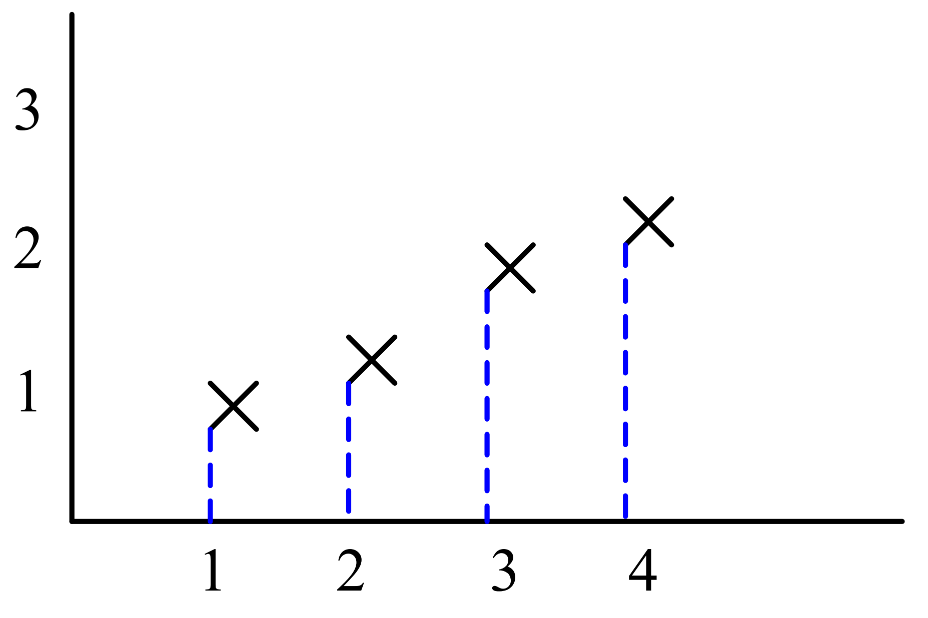
$$Variance = \frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2$$

$$Mean: m' = \frac{1}{n}\sum_{i=1}^{n} w^T x_i = w^T m$$

$$Variance = \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - w^T m)^2$$

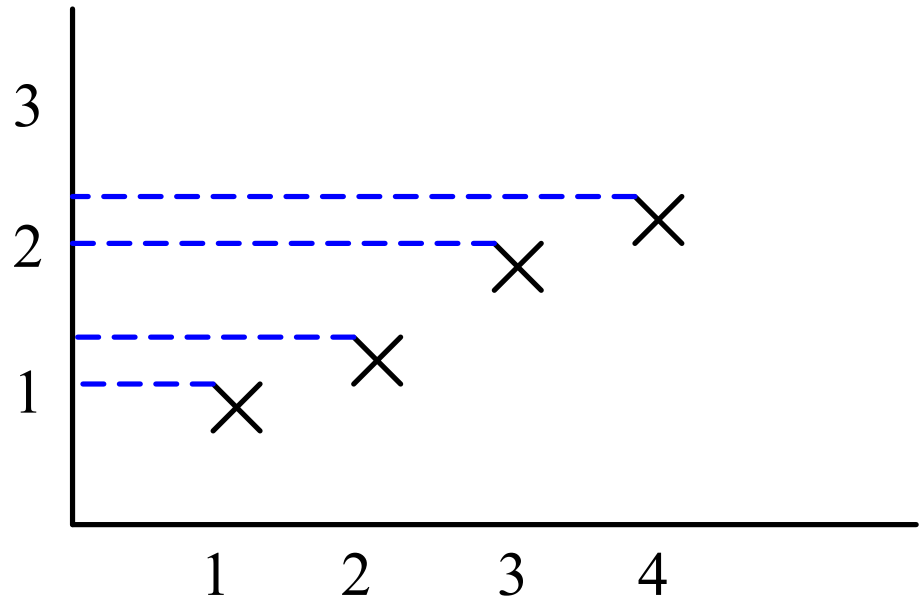
Data projection

- What is the mean and variance of projected data?



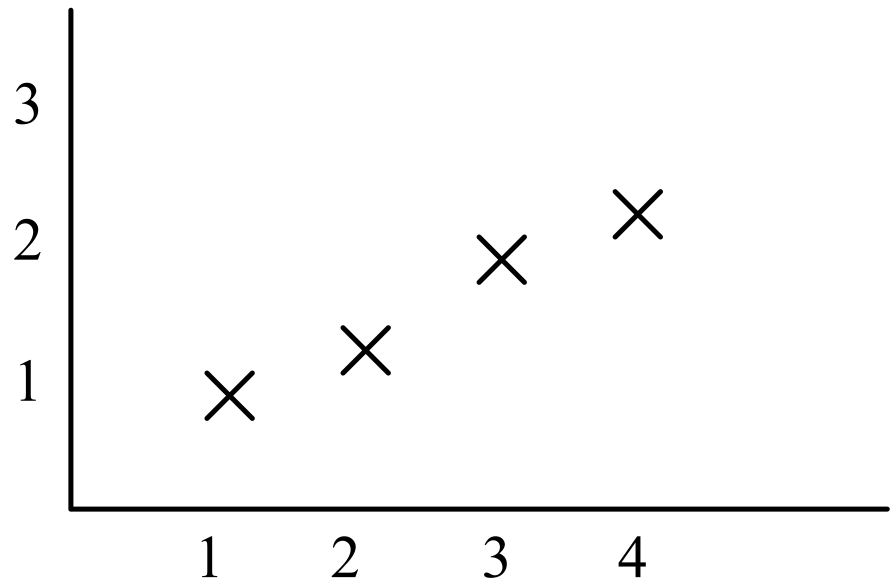
Data projection

- What is the mean and variance here?



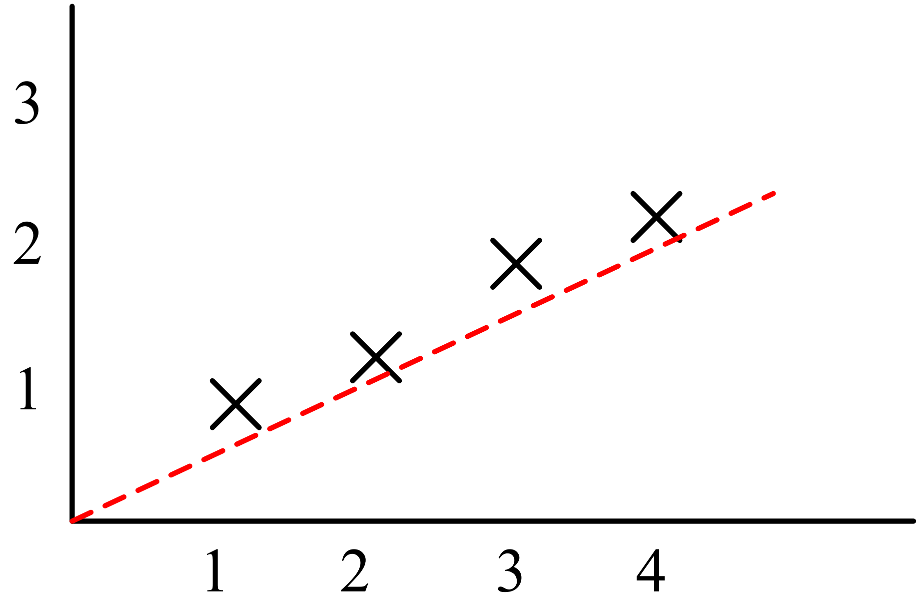
Data projection

- Which line maximizes variance?



Data projection

- Which line maximizes variance?



Principal component analysis

- Find vector w of length 1 that maximizes variance of projected data

PCA optimization problem

$$\arg\max_{w} \frac{1}{n}\sum_{i=1}^{n}(w^T x_i - w^T m)^2 \text{ subject to } w^T w = 1$$

The optimization criterion can be rewritten as

$$\arg\max_{w} \frac{1}{n}\sum_{i=1}^{n}(w^T (x_i - m))^2 =$$

$$\arg\max_{w} \frac{1}{n}\sum_{i=1}^{n}(w^T (x_i - m))^T (w^T (x_i - m)) =$$

$$\arg\max_{w} \frac{1}{n}\sum_{i=1}^{n}((x_i - m)^T w)(w^T (x_i - m)) =$$

$$\arg\max_{w} \frac{1}{n}\sum_{i=1}^{n}w^T (x_i - m)(x_i - m)^T w =$$

$$\arg\max_{w} w^T \frac{1}{n}\sum_{i=1}^{n}(x_i - m)(x_i - m)^T w =$$

$$\arg\max_{w} w^T \Sigma w \text{ subject to } w^T w = 1$$

PCA optimization problem

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - m)(x_i - m)^T$$

is also called the scatter matrix

If we let $X = [x_1 - m, x_2 - m, \ldots, x_n - m]$

where each x_i is a column vector then

$$\Sigma = XX^T$$

PCA solution

- Using Lagrange multipliers we can show that w is given by the largest eigenvector of Σ.
- With this we can compress all the vectors x_i into $w^T x_i$
- Does this help? Before looking at examples, what if we want to compute a second projection $u^T x_i$ such that $w^T u=0$ and $u^T u=1$?
- It turns out that u is given by the second largest eigenvector of Σ.

PCA space and runtime considerations

- Depends on eigenvector computation
- BLAS and LAPACK subroutines
  - Provides Basic Linear Algebra Subroutines.
  - Fast C and FORTRAN implementations.
  - Foundation for linear algebra routines in most contemporary software and programming languages.
  - Different subroutines for eigenvector computation available

PCA space and runtime considerations

- Eigenvector computation requires quadratic space in number of columns
- Poses a problem for high dimensional data
- Instead we can use the Singular Value Decomposition

PCA via SVD

- Every n by n symmetric matrix Σ has an eigenvector decomposition $\Sigma = QDQ^T$ where D is a diagonal matrix containing eigenvalues of Σ and the columns of Q are the eigenvectors of Σ.
- Every m by n matrix A has a singular value decomposition $A = USV^T$ where S is m by n matrix containing singular values of A, U is m by m containing left singular vectors (as columns), and V is n by n containing right singular vectors. Singular vectors are of length 1 and orthogonal to each other.

PCA via SVD

- In PCA the matrix $\Sigma = XX^T$ is symmetric and so the eigenvectors are given by columns of Q in $\Sigma = QDQ^T$.
- The data matrix X (mean subtracted) has the singular value decomposition $X = USV^T$.
- This gives
  - $\Sigma = XX^T = USV^T(USV^T)^T$
  - $USV^T(USV^T)^T = USV^TVSU^T$
  - $USV^TVSU^T = US^2U^T$
- Thus $\Sigma = XX^T = US^2U^T \Rightarrow XX^TU = US^2U^TU = US^2$
- This means the eigenvectors of Σ (principal components of X) are the columns of U and the eigenvalues are the diagonal entries of S^2.

PCA via SVD

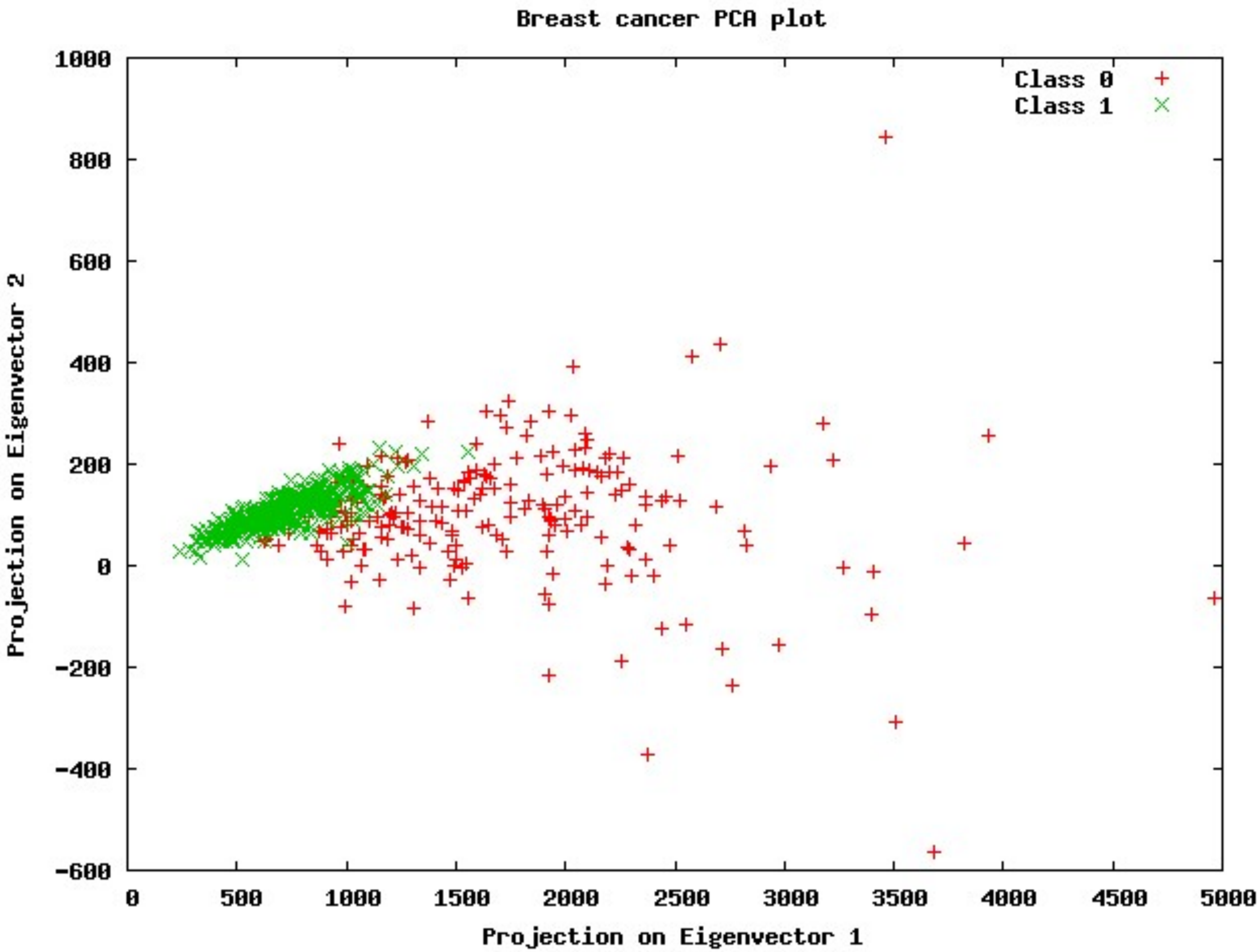
- And so an alternative way to compute PCA is to find the left singular values of X.
- If we want just the first few principal components (instead of all cols) we can implement PCA in rows x cols space with BLAS and LAPACK libraries
- Useful when dimensionality is very high at least in the order of 100s of thousands.

PCA on genomic population data

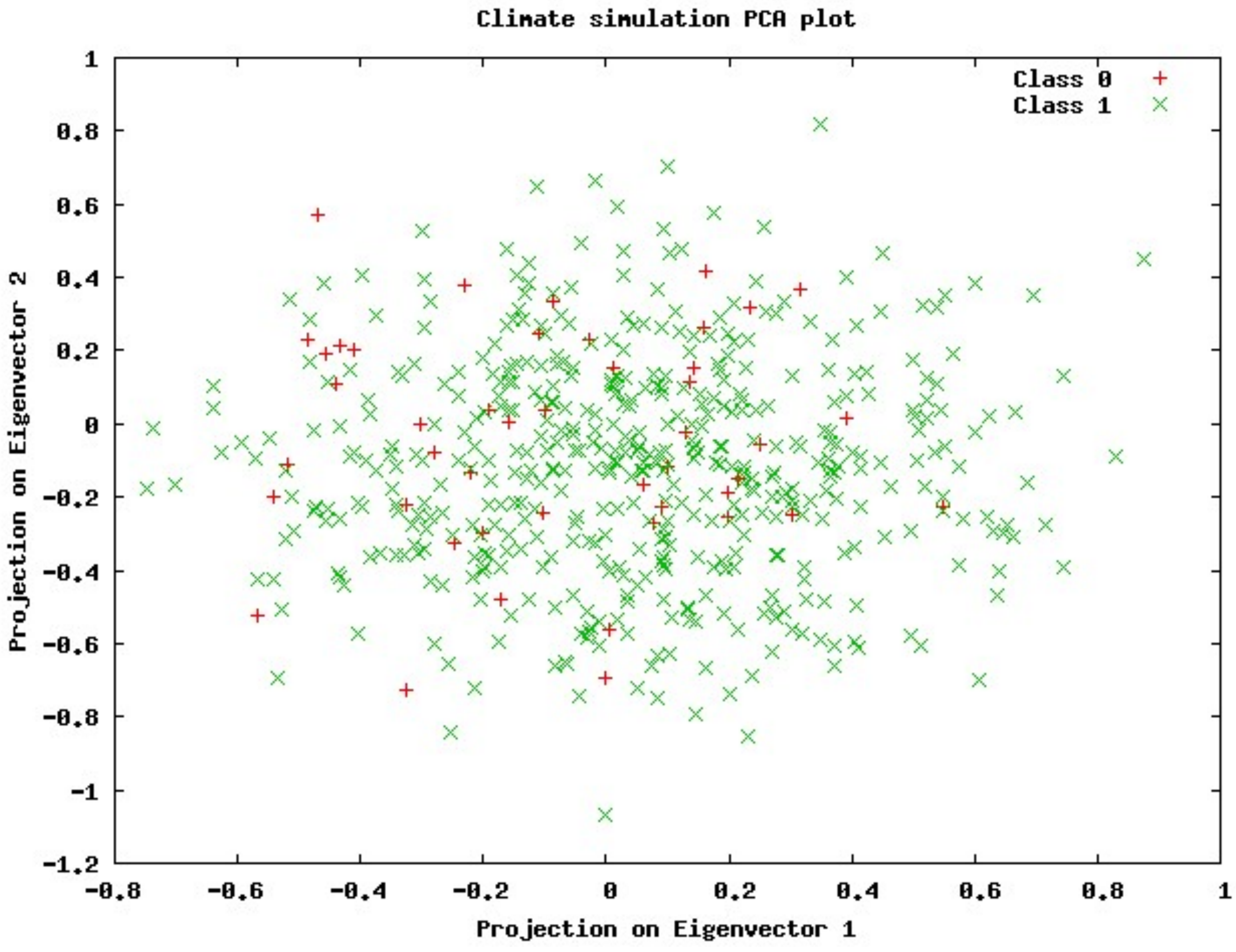
- 45 Japanese and 45 Han Chinese from the International HapMap Project
- PCA applied on 1.7 million SNPs

Taken from "PCA-Correlated SNPs for Structure Identification in Worldwide Human Populations" by Paschou et. al. in PLoS Genetics 2007

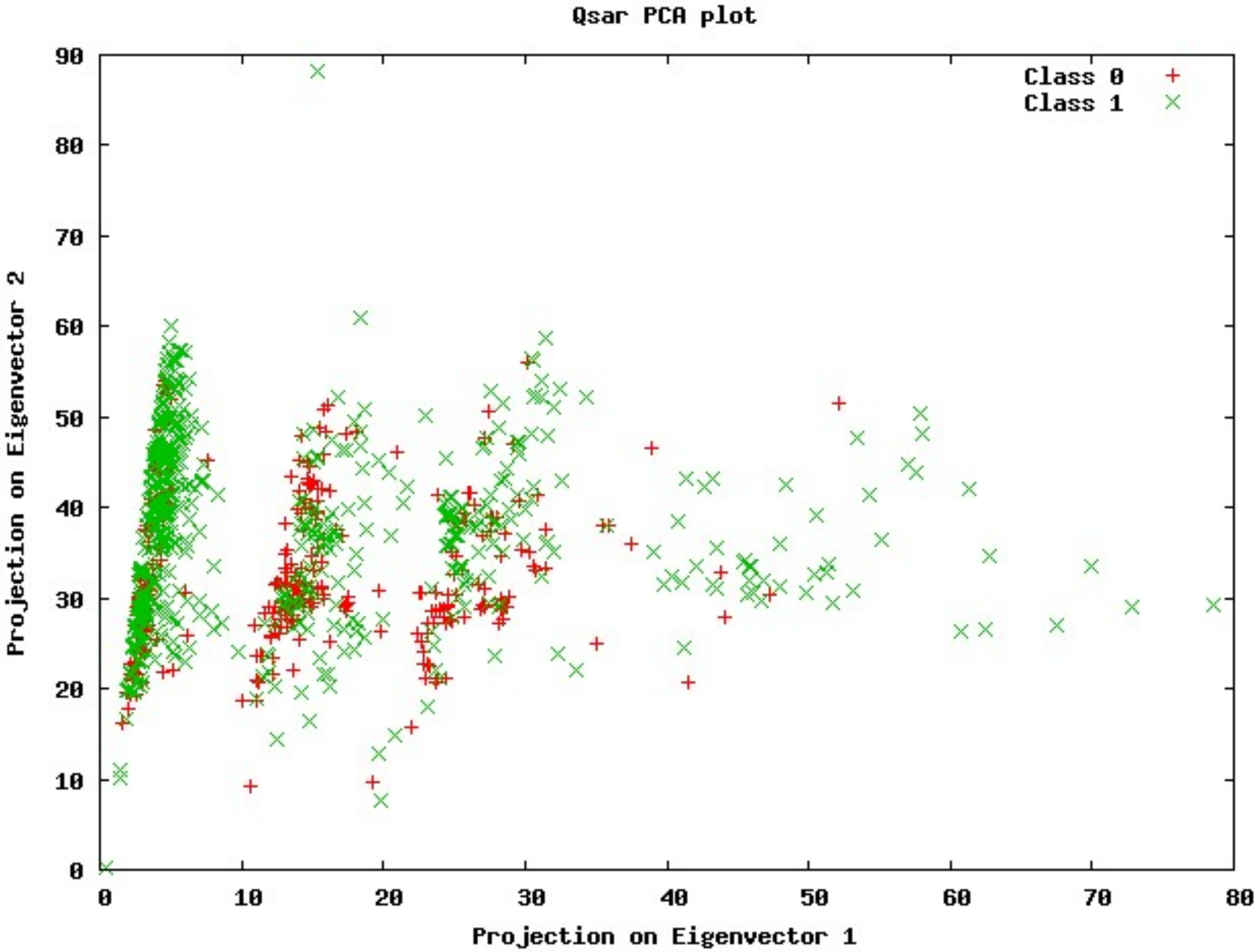
PCA on breast cancer data



PCA on climate simulation

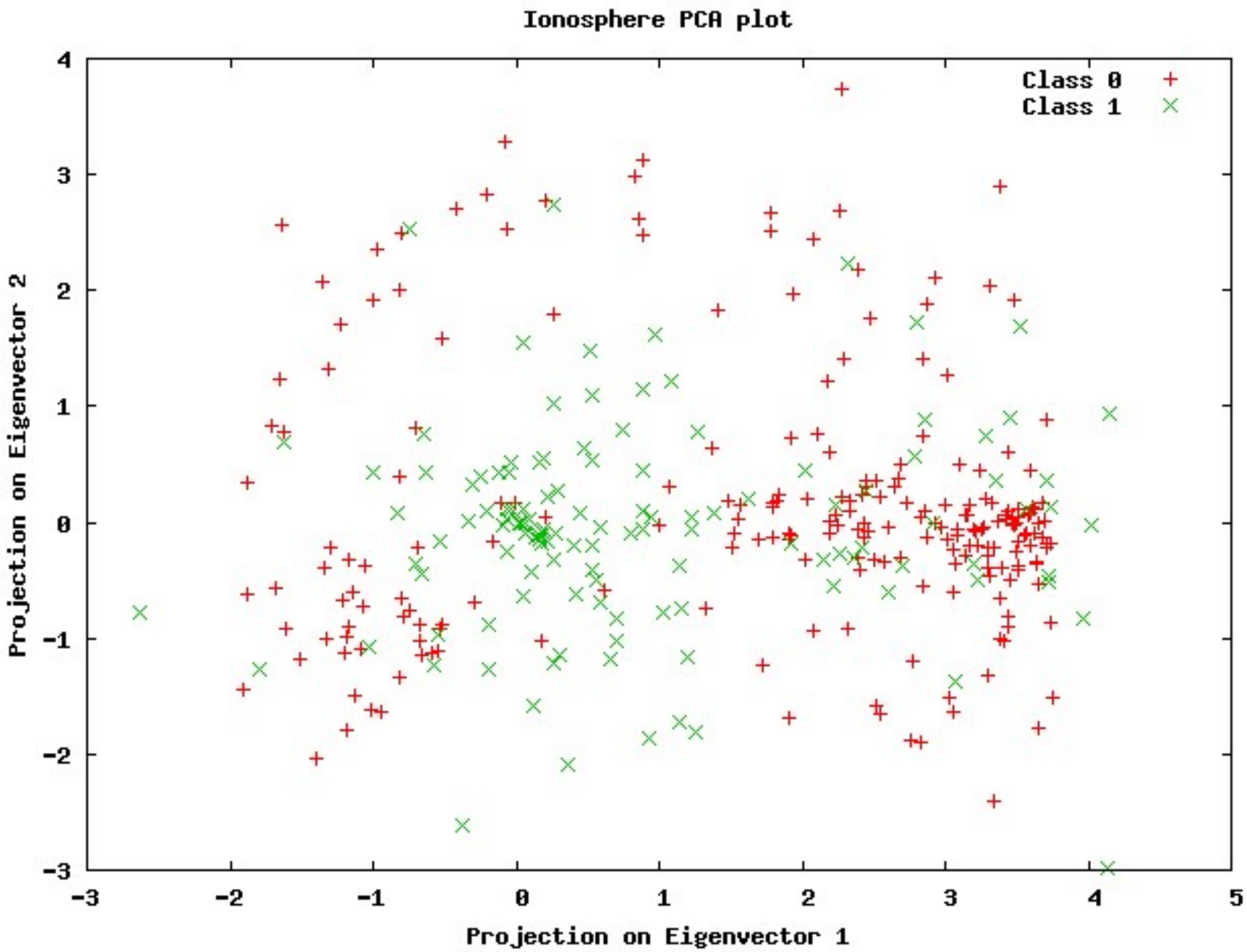


PCA on QSAR



Qsar PCA plot
Class 0
Class 1
Projection on Eigenvector 2
Projection on Eigenvector 1
0
10
20
30
40
50
60
70
80
90

PCA on Ionosphere



Kernel PCA

- Main idea of kernel version
  - $XX^Tw = \lambda w$
  - $X^TXX^Tw = \lambda X^Tw$
  - $(X^TX)X^Tw = \lambda X^Tw$
  - X^Tw is projection of data on the eigenvector w and also the eigenvector of X^TX
- This is also another way to compute projections in space quadratic in number of rows but only gives projections.

Kernel PCA

- In feature space the mean is given by

$$m_{\Phi} = \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)$$

- Suppose for a moment that the data is mean subtracted in feature space. In other words mean is 0. Then the scatter matrix in feature space is given by

$$\Sigma_{\Phi} = \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)\Phi^T(x_i)$$

Kernel PCA

- The eigenvectors of Σ_Φ give us the PCA solution. But what if we only know the kernel matrix?
- First we center the kernel matrix so that mean is 0

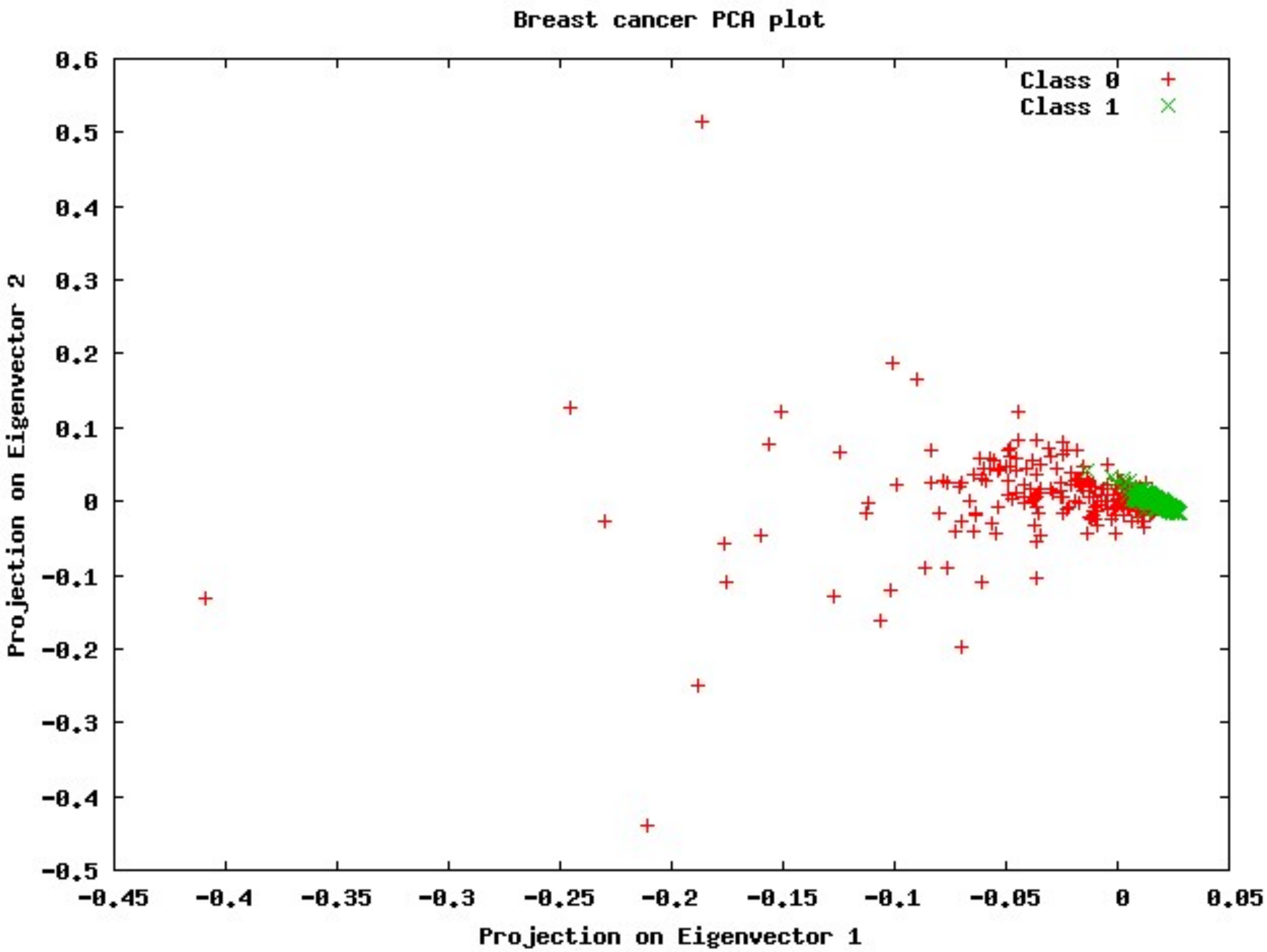
$$\hat{\mathbf{K}} = \mathbf{K} - \frac{1}{\ell}\mathbf{j}\mathbf{j}'\mathbf{K} - \frac{1}{\ell}\mathbf{K}\mathbf{j}\mathbf{j}' + \frac{1}{\ell^2}\left(\mathbf{j}'\mathbf{K}\mathbf{j}\right)\mathbf{j}\mathbf{j}',$$

where j is a vector of 1's.K = K

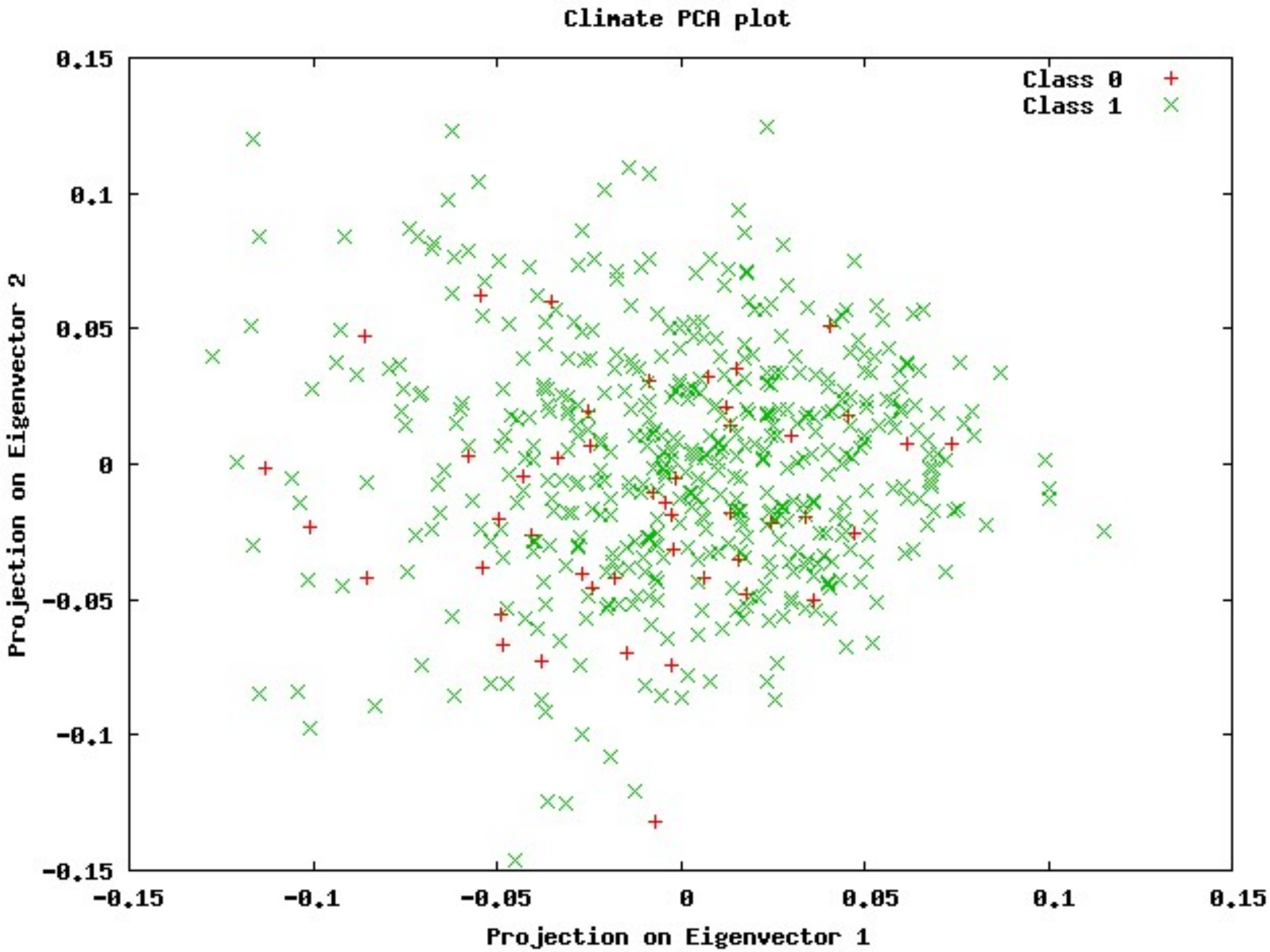
Kernel PCA

- Recall from earlier
  - $XX^T w = \lambda w$
  - $X^T X X^T w = \lambda X^T w$
  - $(X^T X) X^T w = \lambda X^T w$
  - $X^T w$ is projection of data on the eigenvector w and also the eigenvector of $X^T X$
  - $X^T X$ is the linear kernel matrix
- Same idea for kernel PCA
- The projected solution is given by the eigenvectors of the centered kernel matrix.

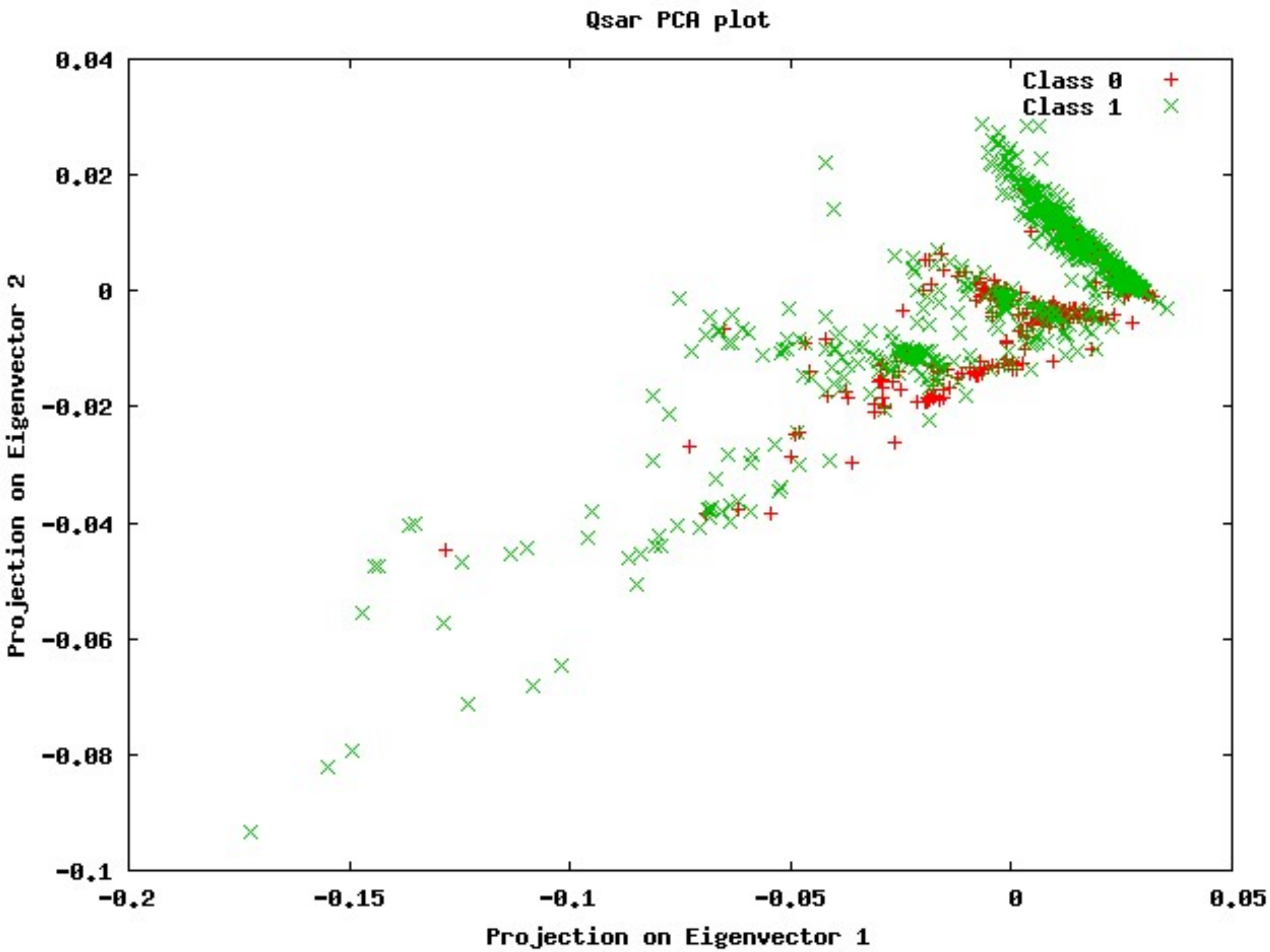
Polynomial degree 2 kernel Breast cancer



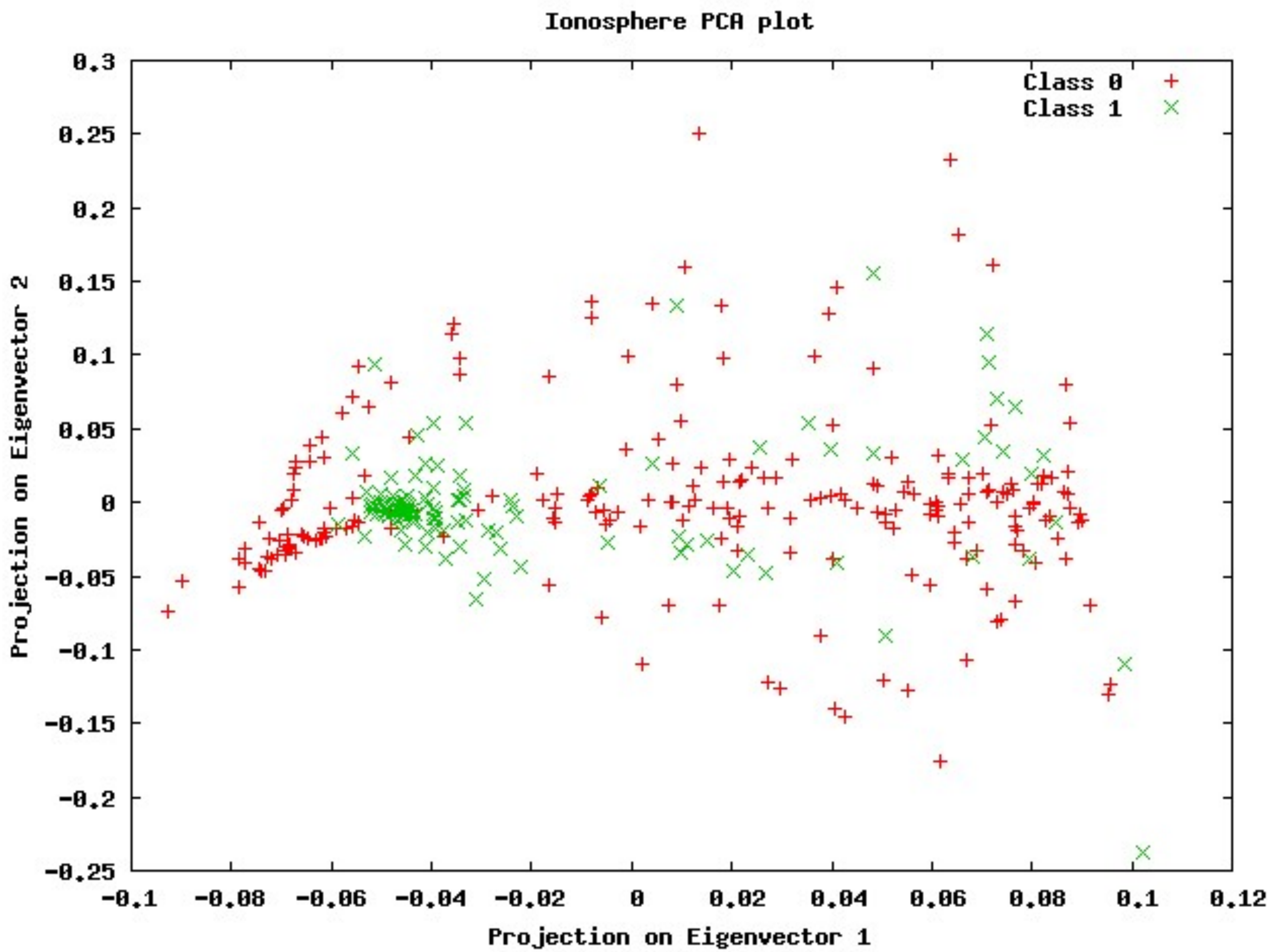
Polynomial degree 2 kernel Climate



Polynomial degree 2 kernel Qsar



Polynomial degree 2 kernel Ionosphere



Supervised dim reduction: Linear discriminant analysis

- Fisher linear discriminant:
  - Maximize ratio of difference means to sum of variance

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2}$$

Linear discriminant analysis

- Fisher linear discriminant:
  - Difference in means of projected data gives us the between-class scatter matrix

$$
\begin{aligned}
(m_1 - m_2)^2 &= (\boldsymbol{w}^T \boldsymbol{m}_1 - \boldsymbol{w}^T \boldsymbol{m}_2)^2 \\
&= \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w} \\
&= \boldsymbol{w}^T \mathbf{S}_B \boldsymbol{w}
\end{aligned}
$$

  - Variance gives us within-class scatter matrix

$$
\begin{aligned}
s_1^2 &= \sum_t (\boldsymbol{w}^T \boldsymbol{x}^t - m_1)^2 r^t \\
&= \sum_t \boldsymbol{w}^T (\boldsymbol{x}^t - \boldsymbol{m}_1)(\boldsymbol{x}^t - \boldsymbol{m}_1)^T \boldsymbol{w} r^t \\
&= \boldsymbol{w}^T \mathbf{S}_1 \boldsymbol{w}
\end{aligned}
$$

Linear discriminant analysis

- Fisher linear discriminant solution:
  - Take derivative w.r.t. w and set to 0
  - This gives us $w = cS_w^{-1}(m_1 - m_2)$

Scatter matrices

- S_b is between class scatter matrix
- S_w is within-class scatter matrix
- $S_t = S_b + S_w$ is total scatter matrix

$$\boldsymbol{S}_b = \frac{1}{n}\sum_{k=1}^{c} n_k\left(\boldsymbol{m}^{(k)} - \boldsymbol{m}\right)\left(\boldsymbol{m}^{(k)} - \boldsymbol{m}\right)^T,$$

$$\boldsymbol{S}_w = \frac{1}{n}\sum_{k=1}^{c}\sum_{j=1}^{n_k}\left(\boldsymbol{x}_j^{(k)} - \boldsymbol{m}^{(k)}\right)\left(\boldsymbol{x}_j^{(k)} - \boldsymbol{m}^{(k)}\right)^T,$$

Fisher linear discriminant

- General solution is given by eigenvectors of $S_w^{-1}S_b$

Fisher linear discriminant

- Problems can happen with calculating the inverse
- A different approach is the maximum margin criterion

Maximum margin criterion (MMC)

- Define the separation between two classes as

$$\|m_1 - m_2\|^2 - s(C_1) - s(C_2)$$

- S(C) represents the variance of the class. In MMC we use the trace of the scatter matrix to represent the variance.
- The scatter matrix is

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - m)(x_i - m)^T$$

Maximum margin criterion (MMC)

- The scatter matrix is

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - m)(x_i - m)^T$$

- The trace (sum of diagonals) is

$$\frac{1}{n}\sum_{j=1}^{d}\sum_{i=1}^{n}(x_{ij} - m_j)^2$$

- Consider an example with two vectors *x* and *y*

Maximum margin criterion (MMC)

- Plug in trace for S(C)   and we get

$$\|m_1 - m_2\|^2 - tr(S_1) - tr(S_2)$$

- The above can be rewritten as

$$tr(S_b) - tr(S_w)$$

- Where S_w is the within-class scatter matrix

$$S_w = \sum_{k=1}^{c} \sum_{x_i \in C_k} (x_i - m_k)(x_i - m_k)^T$$

- And S_b is the between-class scatter matrix

$$S_b = \sum_{k=1}^{c} (m_k - m)(m_k - m)^T$$

Weighted maximum margin criterion (WMMC)

- Adding a weight parameter gives us

$$tr(S_b) - \alpha tr(S_w)$$

- In WMMC dimensionality reduction we want to find w that maximizes the above quantity in the projected space.
- The solution w is given by the largest eigenvector of the above

$$S_b - \alpha S_w$$

How to use WMMC for classification?

- Reduce dimensionality to fewer features
- Run any classification algorithm like nearest means or nearest neighbor.

K-nearest neighbor

- Classify a given datapoint to be the majority label of the k closest points
- The parameter k is cross-validated
- Simple yet can obtain high classification accuracy

Weighted maximum variance (WMV)

- Find w that maximizes the weighted variance

$$\arg\max_{w} \frac{1}{2n} \sum_{i,j} C_{ij} (w^T (x_i - x_j))^2$$

Weighted maximum variance (WMV)

- Reduces to PCA if $C_{ij} = 1/n$

$$\frac{1}{2n}\sum_{i,j}\frac{1}{n}(w^T(x_i - x_j))^2 =$$

$$\frac{1}{2n}\sum_{i,j}\frac{1}{n}w^T(x_i - x_j)(x_i - x_j)^T w =$$

$$\frac{1}{2n}\sum_{i,j}\frac{1}{n}w^T(x_i x_i^T - x_i x_j^T - x_j x_i^T + x_j x_j^T)w =$$

$$\frac{1}{2n}w^T\frac{1}{n}(\sum_{i,j}(x_i x_i^T - x_i x_j^T - x_j x_i^T + x_j x_j^T))w =$$

$$\frac{1}{2n}w^T\frac{1}{n}(\sum_{i,j} x_i x_i^T - \sum_{i,j} x_i x_j^T - \sum_{i,j} x_j x_i^T + \sum_{i,j} x_j x_j^T)w$$

$$\frac{1}{2n}w^T\frac{1}{n}(2\sum_{i,j} x_i x_i^T - 2\sum_{i,j} x_i x_j^T)w =$$

$$\frac{1}{2n}w^T\frac{1}{n}(2n\sum_{i} x_i x_i^T - 2n^2 mm^T)w =$$

$$\frac{1}{n}w^T(\sum_{i} x_i x_i^T - nmm^T)w =$$

$$w^T(\frac{1}{n}\sum_i (x_i - m)(x_i - m)^T)w =$$

$$w^T S_t w$$

MMC via WMV

- Let y_i be class labels and let nk be the size of class k.
- Let G_{ij} be 1/n for all i and j and L_{ij} be 1/ n_k if i and j are in same class.
- Then MMC is given by

$$\arg\max_{w} \frac{1}{2n}\left(\sum_{i,j} G_{ij}(w^T(x_i - x_j))^2 - \sum_{i,j} 2L_{ij}(w^T(x_i - x_j))^2\right)$$

MMC via WMV (proof sketch)

$$\frac{1}{2n}\sum_{i,j} w^T(G_{ij}(x_i - x_j)(x_i - x_j) - 2L_{ij}(x_i - x_j)(x_i - x_j)^T)w =$$

$$\frac{1}{2n}(\sum_{i,j}\frac{1}{n}w^T(x_i - x_j)(x_i - x_j)^T w -$$
$$2\sum_{k=1}^{c}\sum_{cl(x_j)=k, cl(x_i)=k}\frac{1}{n_k}w^T(x_i - x_j)(x_i - x_j)^T w) =$$

$$\frac{1}{2n}(2\sum_i^n w^T(x_i - m)(x_i - m)w -$$
$$2\sum_{k=1}^{c}\frac{1}{n_k}\sum_{cl(x_j)=k, cl(x_i)=k} w^T(x_i x_i^T - x_i x_j^T - x_j x_i^T + x_j x_j^T)w) =$$

$$\frac{1}{2n}(2\sum_i^n w^T(x_i - m)(x_i - m)w -$$
$$2\sum_{k=1}^{c}\frac{1}{n_k}\sum_{cl(x_j)=k, cl(x_i)=k} w^T(2x_i x_i^T - 2x_i x_j^T)w) =$$

$$\frac{1}{2n}(2\sum_i^n w^T(x_i - m)(x_i - m)w -$$
$$2\sum_{k=1}^{c}\frac{1}{n_k}\sum_{cl(x_i)=k} w^T(2n_k x_i x_i^T - 2n_k^2 m_k m_k^T)w) =$$

$$\frac{1}{n}(\sum_i^n w^T(x_i - m)(x_i - m)w -$$
$$2\sum_{k=1}^{c}\sum_{cl(x_i)=k} w^T(x_i x_i^T - n_k m_k m_k^T)w) =$$

$$\frac{1}{n}(\sum_i^n w^T(x_i - m)(x_i - m)w -$$
$$2\sum_{k=1}^{c}\sum_{cl(x_i)=k} w^T(x_i - m_k)(x_i - m_k)^T)w) =$$

$$w^T(S_t - 2S_w)w$$

Graph Laplacians

- We can rewrite WMV with Laplacian matrices.
- Recall WMV is $\arg\max_w \frac{1}{2n} \sum_{i,j} C_{ij}(w^T(x_i - x_j))^2$
- Let L = D – C where $D_{ii} = \Sigma_j C_{ij}$
- Then WMV is given by $\arg\max_w \frac{1}{n} w^T X L X^T w$ where $X = [x_1, x_2, \ldots, x_n]$ contains each x_i as a column.
- w is given by largest eigenvector of XLX^T

Graph Laplacians

- Widely used in spectral clustering (see tutorial on course website)
- Weights C_{ij} may be obtained via
  - Epsilon neighborhood graph
  - K-nearest neighbor graph
  - Fully connected graph
- Allows semi-supervised analysis (where test data is available but not labels)

Graph Laplacians

- We can perform clustering with the Laplacian
- Basic algorithm for k clusters:
  - Compute first k eigenvectors v_i of Laplacian matrix
  - Let $V = [v_1, v_2, \ldots, v_k]$
  - Cluster rows of V (using k-means)
- Why does this work?

Graph Laplacians

- We can cluster data using the mincut problem
- Balanced version is NP-hard
- We can rewrite balanced mincut problem with graph Laplacians. Still NP-hard because solution is allowed only discrete values
- By relaxing to allow real values we obtain spectral clustering.

Back to WMV – a two parameter approach

- Recall that WMV is given by

$$\arg\max_{w} \frac{1}{2n} \sum_{i,j} C_{ij} (w^T (x_i - x_j))^2$$

- Collapse C_{ij} into two parameters
  - $C_{ij} = \alpha < 0$ if i and j are in same class
  - $C_{ij} = \beta > 0$ if i and j are in different classes
- We call this 2-parameter WMV

Experimental results

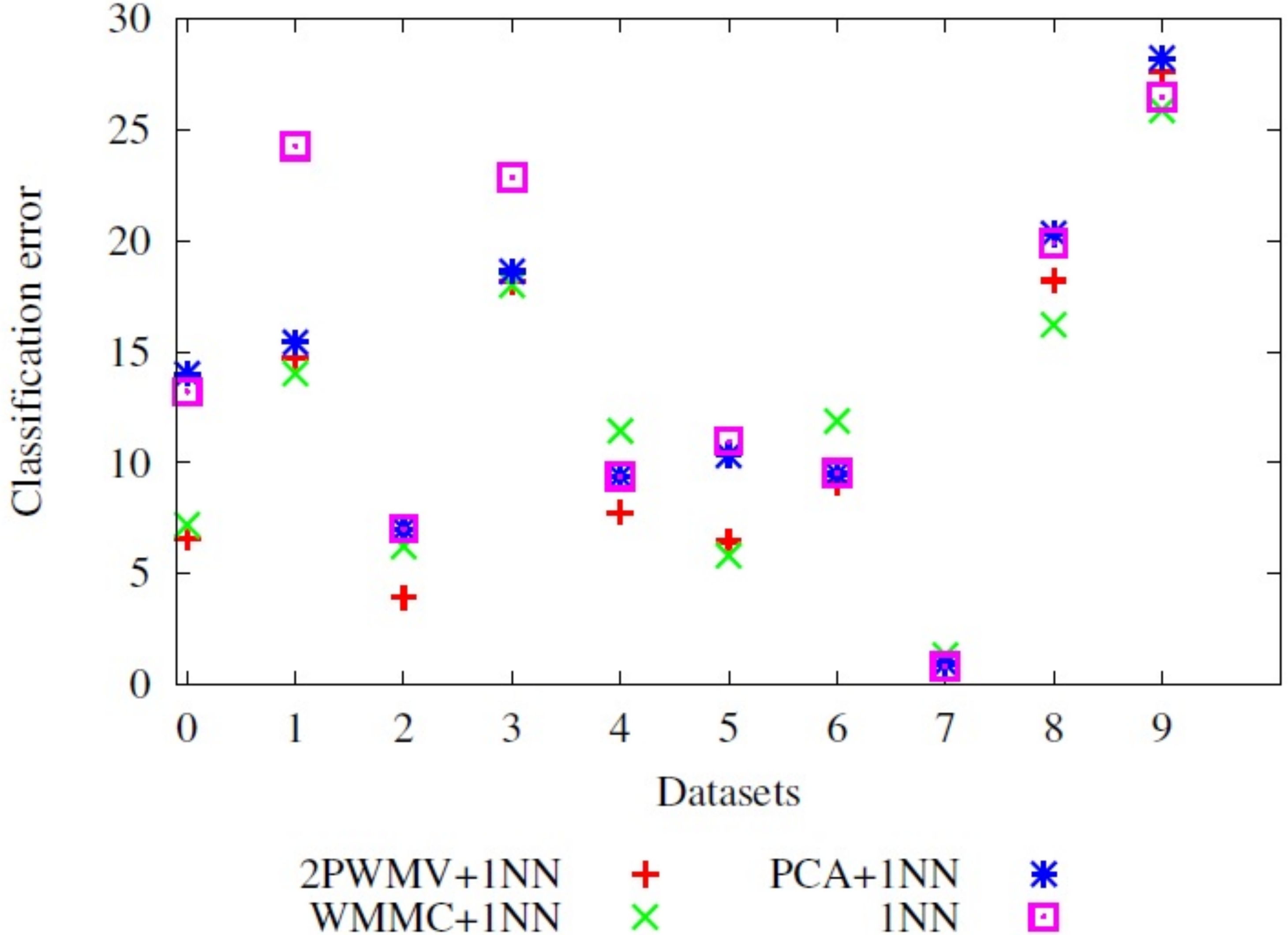
- To evaluate dimensionality reduction for classification we first extract features and then apply 1-nearest neighbor in cross-validation
- 20 datasets from UCI machine learning archive
- Compare 2PWMV+1NN, WMMC+1NN, PCA+1NN, 1NN
- Parameters for 2PWMV+1NN and WMMC+1NN obtained by cross-validation

Datasets

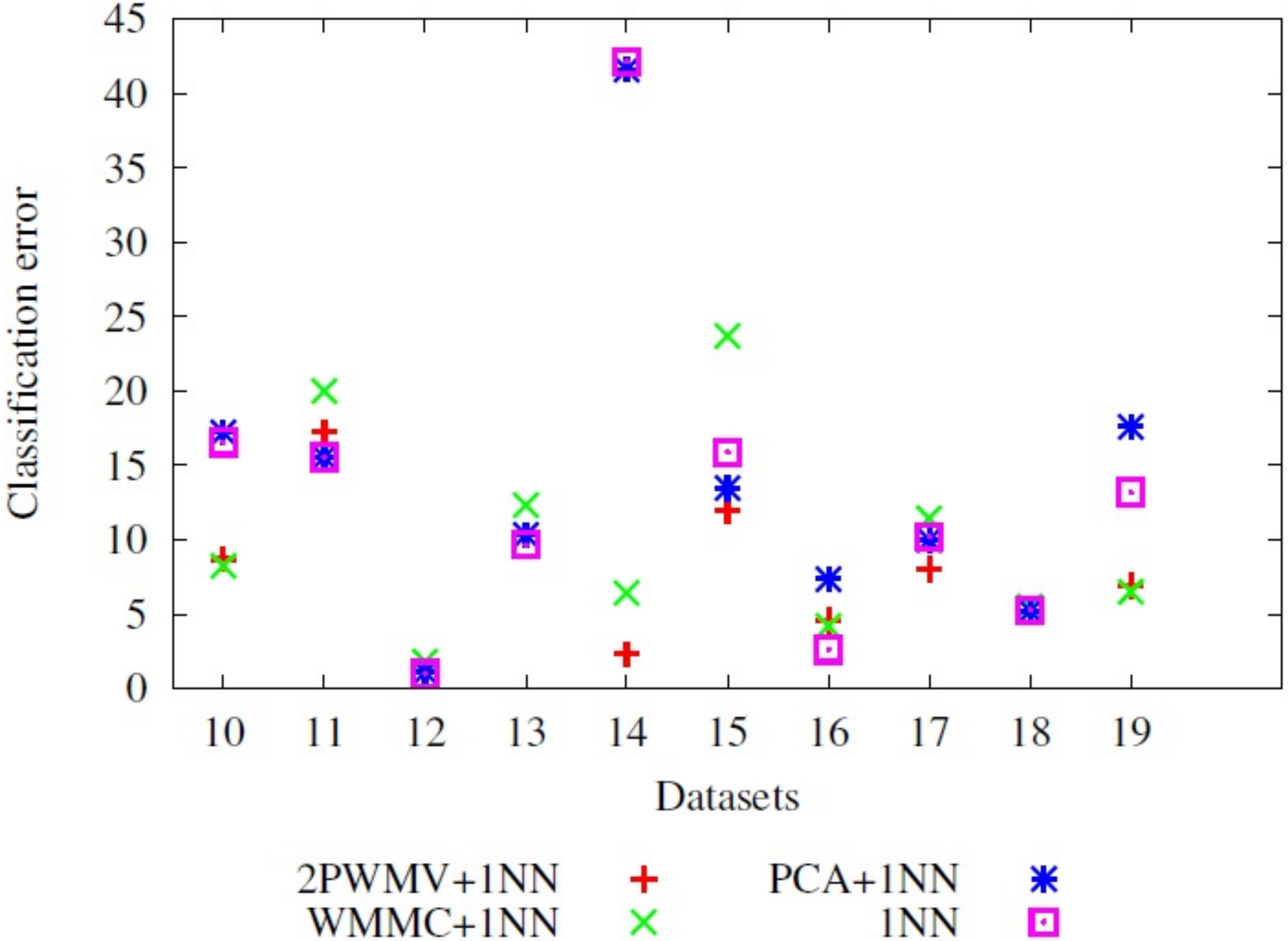
Table 2:Twenty Datasets for Classification

Code	Dataset	Classes	Dimension	Instances
0	Climate	2	18	540
1	Ring	2	20	7400
2	Thyroid	3	21	7200
3	Waveform	3	21	5000
4	Breast cancer	2	30	569
5	Ionosphere	2	34	351
6	Statlog	7	36	6435
7	Texture	11	40	5500
8	Qsar	2	41	1055
9	SPECTF heart	2	44	267
10	Spambase	2	57	4597
11	Sonar	2	60	208
12	Digits	2	63	762
13	Movement libras	15	90	360
14	Hill valley	2	100	606
15	Musk	2	166	476
16	Smartphone	6	561	10299
17	Secom	2	591	1567
18	Mfeat	10	649	2000
19	CNAE-9	9	857	1080

Results


30
25
20
15
10
5
0
Classification error
0
1
2
3
4
5
6
7
8
9
Datasets
2PWMV+1NN
WMMC+1NN
PCA+1NN
1NN

Results


Classification error
45
40
35
30
25
20
15
10
5
0
10
11
12
13
14
15
16
17
18
19
Datasets
2PWMV+1NN
WMMC+1NN
PCA+1NN
1NN

Results

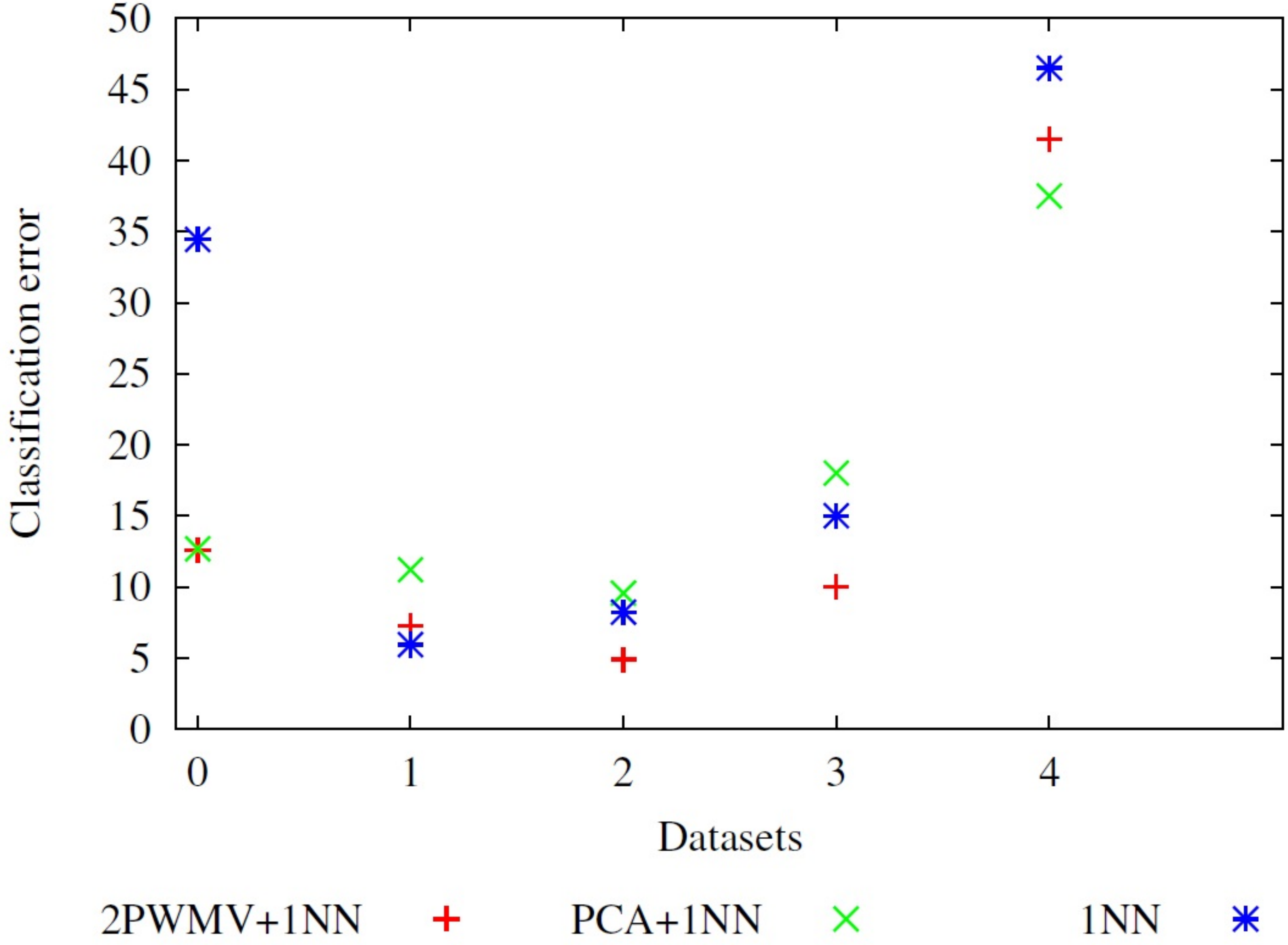
- Average error:
  - 2PWMV+1NN: 9.5% (winner in 9 out of 20)
  - WMMC+1NN: 10% (winner in 7 out of 20)
  - PCA+1NN: 13.6%
  - 1NN: 13.8%
- Parametric dimensionality reduction does help

High dimensional data

Table 1:Five High Dimensional Datasets

Code	Dataset	Classes	Dimension	Instances
0	Madelon	2	500	2600
1	Micromass	2	1300	931
2	Gisette	2	5000	1000
3	Arcene	2	10000	200
4	Dexter	2	20000	300

High dimensional data



Results

- Average error on high dimensional data:
  - 2PWMV+1NN: 15.2%
  - PCA+1NN: 17.8%
  - 1NN: 22%