# 6. Geometric applications

- localization from multiple camera views
- orthogonal Procrustes problem
- fitting affine sets to points
- linear discriminant analysis

# Introduction

applications in this lecture use matrix methods to solve problems in geometry

- $m \times n$ matrix is interpreted as collection of $m$ points in $\mathbf{R}^n$ or $n$ points in $\mathbf{R}^m$
- $m \times n$ matrices parametrize affine functions $f(x) = Ax + b$ from $\mathbf{R}^n$ to $\mathbf{R}^m$
- $m \times n$ matrices parametrize affine sets $\{x \mid Ax = b\}$ in $\mathbf{R}^n$

# Multiple view geometry

- $n$ objects at positions $x_j \in \mathbf{R}^3$, $j = 1, \ldots, n$, are viewed by $l$ cameras
- $y_{ij} \in \mathbf{R}^2$ is the location of object $j$ in the image acquired by camera $i$
- each camera is modeled as an affine mapping:

$$y_{ij} = P_i x_j + q_i, \quad i = 1, \ldots, l, \quad j = 1, \ldots, n$$

define a $2l \times n$ matrix with the observations $y_{ij}$:

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & & \vdots \\ y_{l1} & y_{l2} & \cdots & y_{ln} \end{bmatrix} = \begin{bmatrix} P_1 & q_1 \\ P_2 & q_2 \\ \vdots & \\ P_l & q_l \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

- 2nd equality assumes noise-free observations and perfectly affine cameras
- the goal is to estimate the positions $x_j$ and the camera models $P_i$, $q_i$

# Factorization algorithm

minimize Frobenius norm of error between model predictions and observations $Y$

$$\text{minimize} \quad \| PX + q\mathbf{1}^T - Y \|_F^2$$

- $P$ is $2l \times 3$ matrix and $q$ is $2l$-vector with the $l$ camera models:

$$P = \begin{bmatrix} P_1 \\ \vdots \\ P_l \end{bmatrix}, \qquad q = \begin{bmatrix} q_1 \\ \vdots \\ q_l \end{bmatrix}$$

- variables are the $3 \times n$ position matrix $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}$ and camera models $P$, $q$
- variable $q$ can be eliminated: least squares estimate is $q = (1/n)(Y - PX)\mathbf{1}$
- substituting expression for optimal $q$ gives

$$\begin{array}{ll} \text{minimize} & \|PX_\mathrm{c} - Y_\mathrm{c}\|_F^2 \\ \text{subject to} & X_\mathrm{c}\mathbf{1} = 0 \end{array}$$

here $Y_c = Y(I - (1/n)\mathbf{1}\mathbf{1}^T)$ and the variable is $X_\mathrm{c} = X(I - (1/n)\mathbf{1}\mathbf{1}^T)$

# Factorization algorithm

$$\begin{array}{ll} \text{minimize} & \|PX_\mathrm{c} - Y_\mathrm{c}\|_F^2 \\ \text{subject to} & X_\mathrm{c}\mathbf{1} = 0 \end{array}$$

with variables $P$ (a $2l \times 3$ matrix) and $X_\mathrm{c}$ (a $3 \times 2n$ matrix)

- the solution follows from an SVD of $Y_\mathrm{c}$:

$$Y_\mathrm{c} = \sum_{i=1}^{2n} \sigma_i u_i v_i^T$$

- (assuming $\mathrm{rank}(Y_\mathrm{c}) \geq 3$) truncate SVD after 3 terms and define:

$$P = \left[ \begin{array}{ccc} \sigma_1 u_1 & \sigma_2 u_2 & \sigma_3 u_3 \end{array} \right], \qquad X_\mathrm{c} = \left[ \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \right]^T$$

- vectors $v_1$, $v_2$, $v_3$ are in the row space of $Y_\mathrm{c}$, hence orthogonal to $\mathbf{1}$
- solution is not unique, since $PX_\mathrm{c} = (PT)(T^{-1}X_\mathrm{c})$ for any nonsingular $T$
- this ambiguity corresponds to the choice of coordinate system in $\mathbf{R}^3$

# References


- Carlo Tomasi and Takeo Kanade, *Shape and motion from image streams under orthography: A factorization approach*, International Journal of Computer Vision (1992).

  the original paper on the factorization method

- Takeo Kanade and Daniel D. Morris, *Factorization methods for structure from motion*, Phil. Trans. R. Soc. of Lond. A (1998).

  a more recent survey of the factorization method and extensions

# Outline

- localization from multiple camera views
- **orthogonal Procrustes problem**
- fitting affine sets to points
- linear discriminant analysis

# Orthogonal Procrustes problem

given $m \times n$ matrices $A$, $B$, solve the optimization problem

$$\begin{array}{ll} \text{minimize} & \|AX - B\|_F^2 \\ \text{subject to} & X^T X = I \end{array} \qquad (1)$$

the variable is an $n \times n$ matrix $X$

- a matrix least squares problem with constraint that $X$ is orthogonal
- rows of $B$ are approximated by orthogonal linear function applied to rows of $A$

**Solution:** $X = VU^T$ with $U$, $V$ from an SVD of the $n \times n$ matrix $B^T A = U\Sigma V^T$

# Solution of orthogonal Procrustes problem

- the problem is equivalent to maximizing $\mathrm{trace}(B^TAX)$ over orthogonal $X$:

$$\begin{aligned}
\|AX-B\|_F^2 &= \mathrm{trace}((AX-B)(AX-B)^T) \\
&= \|A\|_F^2 + \|B\|_F^2 - 2\,\mathrm{trace}(AXB^T) \\
&= \|A\|_F^2 + \|B\|_F^2 - 2\,\mathrm{trace}(B^TAX)
\end{aligned}$$

- compute $n\times n$ SVD $B^TA = U\Sigma V^T$ and make change of variables $Y = V^TXU$:

$$\begin{array}{ll}
\text{maximize} & \mathrm{trace}(\Sigma Y) = \sum_{i=1}^n \sigma_i Y_{ii} \\
\text{subject to} & Y^TY = I
\end{array} \tag{2}$$

- if $Y$ is orthogonal, then $Y_{ii}\le 1$ and $\mathrm{trace}(\Sigma Y)\le \sum_{i=1}^n \sigma_i$:

$$1 = (Y^TY)_{ii} = Y_{ii}^2 + \sum_{j\neq i} Y_{ji}^2 \ge Y_{ii}^2$$

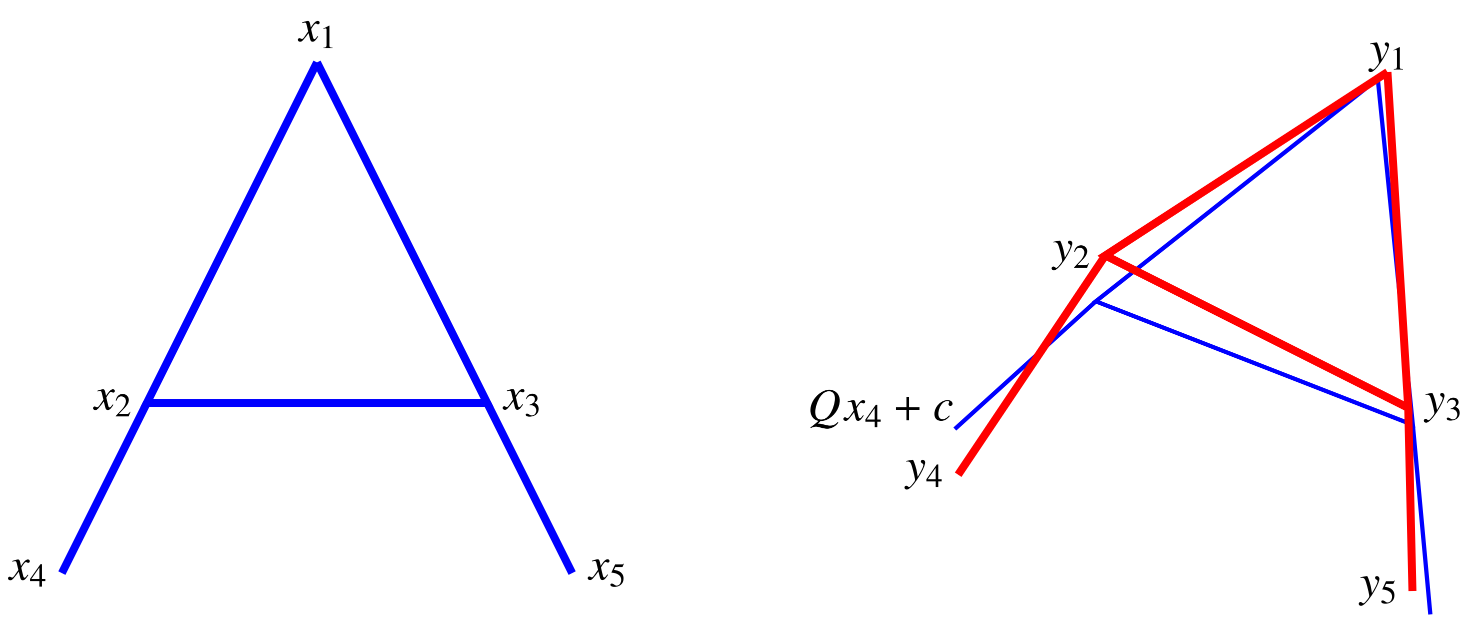
- hence $Y = I$ is optimal for (2) and $X = VYU^T = VU^T$ is optimal for (1)

# Application

given two sets of points $x_1, \ldots, x_m$ and $y_1, \ldots, y_m$ in $\mathbf{R}^n$, solve the problem

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} \|Qx_i + c - y_i\|^2 \\ \text{subject to} & Q^T Q = I \end{array}$$

- the variables are an $n \times n$ matrix $Q$ and $n$-vector $c$
- $Q$ and $c$ define a shape-preserving affine mapping $f(x) = Qx + c$



# Solution

the problem is equivalent to an orthogonal Procrustes problem

- for given $Q$, optimal $c$ is
$$c = \frac{1}{m}\sum_{i=1}^{m}(y_i - Qx_i)$$
- substitute expression for optimal $c$ in the cost function:
$$\sum_{i=1}^{m} \|Qx_i + c - y_i\|^2 = \sum_{i=1}^{m} \|Q\tilde{x}_i - \tilde{y}_i\|^2 = \|Q\tilde{X} - \tilde{Y}\|_F^2$$
where $\tilde{X} = \begin{bmatrix} \tilde{x}_1 & \cdots & \tilde{x}_m \end{bmatrix}$, $\tilde{Y} = \begin{bmatrix} \tilde{y}_1 & \cdots & \tilde{y}_m \end{bmatrix}$, and $\tilde{x}_i$, $\tilde{y}_i$ are the centered points
$$\tilde{x}_i = x_i - \frac{1}{m}\sum_{j=1}^{m} x_j, \qquad \tilde{y}_i = y_i - \frac{1}{m}\sum_{j=1}^{m} y_j,$$
- optimal $Q$ minimizes $\|Q\tilde{X} - \tilde{Y}\|_F^2 = \|\tilde{X}^T Q^T - \tilde{Y}^T\|_F^2$ over orthogonal matrices

# Outline

- localization from multiple camera views
- orthogonal Procrustes problem
- **fitting affine sets to points**
- linear discriminant analysis

# Affine set

a subset $\mathcal{S}$ of $\mathbf{R}^n$ is *affine* if

$$\alpha x + \beta y \in \mathcal{S}$$

for all vectors $x, y \in \mathcal{S}$ and all scalars $\alpha, \beta$ with $\alpha + \beta = 1$

- affine combinations of elements of $\mathcal{S}$ are in $\mathcal{S}$
- if $x \neq y$ are two points in $\mathcal{S}$, then the entire line through $x$, $y$ is in $\mathcal{S}$

**Examples**

- a subspace $\mathcal{V}$ is an affine set: if $x, y \in \mathcal{V}$ then $\alpha x + \beta y \in \mathcal{V}$ for all $\alpha, \beta$
- subspace plus vector: $\{x + a \mid x \in \mathcal{V}\}$ where $\mathcal{V}$ is a subspace and $a$ a vector
- solution set of linear equation $\{x \mid Ax = b\}$
- the empty set is affine (but not a subspace)

# Parallel subspace

suppose $\mathcal{S}$ is a nonempty affine set, $x_0$ is a point in $\mathcal{S}$, and define

$$\mathcal{V} = \{x - x_0 \mid x \in \mathcal{S}\}$$

- $\mathcal{V}$ is a subspace: if $x \in \mathcal{V}$, $y \in \mathcal{V}$, then $x + x_0 \in \mathcal{S}$, $y + x_0 \in \mathcal{S}$, and

$$\alpha x + \beta y + x_0 = \alpha(x + x_0) + \beta(y + x_0) + (1 - \alpha - \beta)x_0 \in \mathcal{S} \quad \text{for all } \alpha, \beta$$

  (right-hand side is affine combination of 3 points $x + x_0$, $y + x_0$, and $x_0$ in $\mathcal{S}$)

- $\mathcal{V}$ does not depend on the choice of $x_0 \in \mathcal{S}$: if $x + x_0 \in \mathcal{S}$ and $y_0 \in \mathcal{S}$, then

$$x + y_0 = (x + x_0) - x_0 + y_0 \in \mathcal{S}$$

  (right-hand side is affine combination of 3 points $x + x_0$, $x_0$, $y_0$ in $\mathcal{S}$)

- the dimension of $\mathcal{S}$ is defined as the dimension of the parallel subspace $\mathcal{V}$

# Range representation

every nonempty affine set $\mathcal{S} \subseteq \mathbf{R}^m$ can be represented as

$$\mathcal{S} = \{Ax + b \mid x \in \mathbf{R}^n\}$$

- $b$ is any vector in $\mathcal{S}$
- $A$ is any matrix with range equal to the parallel subspace: $\mathcal{S} = \operatorname{range}(A) + b$
- $\dim(\mathcal{S}) = \operatorname{rank}(A)$

# Nullspace representation

every affine set $\mathcal{S} \subseteq \mathbf{R}^n$ (including the empty set) can be represented as

$$\mathcal{S} = \{x \in \mathbf{R}^n \mid Ax = b\}$$

for a nonempty affine set $\mathcal{S}$:

- $b = Ax_0$ where $x_0$ is any vector in $\mathcal{S}$
- $A$ is any matrix with nullspace equal to the parallel subspace: $\mathcal{S} = \mathrm{null}(A) + x_0$
- $\dim(\mathcal{S}) = \mathrm{rank}(A) - n$

the empty set is the solution set of an inconsistent equation (*e.g.*, $A = 0$, $b \neq 0$)

# Distance to affine set

suppose $\mathcal{S}$ is the affine set $\mathcal{S} = \{y \mid Ay = b\}$

**Projection:** projection of $x$ on $\mathcal{S}$ is the solution $y$ of the "least-distance" problem

$$\begin{array}{ll} \text{minimize} & \|y - x\| \\ \text{subject to} & Ay = b \end{array}$$

- if $A$ has linearly independent rows, $y = x + A^\dagger(b - Ax)$
- if $A$ has orthonormal rows, $y = x + A^T(b - Ax)$

**Distance**: we denote the distance of $x$ to $\mathcal{S}$ by $d(x, \mathcal{S})$

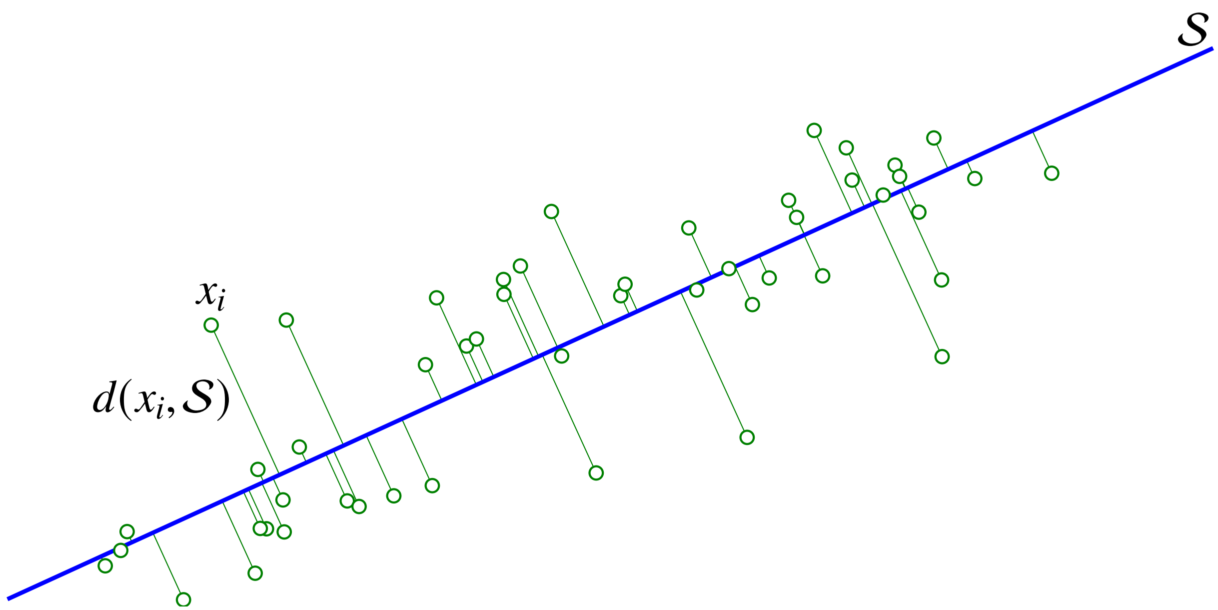
- if $A$ has linearly independent rows, $d(x, \mathcal{S}) = \|A^\dagger(Ax - b)\|$
- if $A$ has orthonormal rows, $d(x, \mathcal{S}) = \|Ax - b\|$

# Least squares fit of affine set to points

fit an affine set $\mathcal{S}$ of specified dimension $k$ to $N$ points $x_1, \ldots, x_N$ in $\mathbf{R}^n$:

$$\text{minimize} \quad \sum_{i=1}^{N} d(x_i, \mathcal{S})^2$$

**Example:** $k = 1$, $N = 50$, $n = 2$



# Least squares fit of affine set to points

use nullspace representation $\mathcal{S} = \{x \mid Ax = b\}$, where $A$ has orthonormal rows:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} \|Ax_i - b\|^2 \\ \text{subject to} & AA^T = I \end{array}$$

the variables are the $m \times n$ matrix $A$ and $m$-vector $b$, where $m = n - k$

**Algorithm** (assuming $m \leq n \leq N$):

- compute center $\bar{x} = (1/N)(x_1 + \cdots + x_N)$
- rows of optimal $A$ are the last $m$ left singular vectors of matrix of centered points

$$X = \left[\begin{array}{cccc} x_1 - \bar{x} & x_2 - \bar{x} & \cdots & x_N - \bar{x} \end{array}\right]$$

- optimal $b$ is $b = A\bar{x}$

we derive this solution on the next page

# Least squares fit of affine set to points

- for given $A$, optimal $b$ is average $(1/N)A(x_1 + \cdots + x_N) = A\bar{x}$
- eliminating $b$ reduces the problem to an optimization over $m \times n$ variable $A$

$$\begin{array}{ll} \text{minimize} & \|AX\|_F^2 \\ \text{subject to} & AA^T = I \end{array}$$

- denote singular values of $m \times N$ matrix $AX$ by $\tau_1 \geq \cdots \geq \tau_m$
- if $X = U\Sigma V^T$ is a full SVD, then by the min–max properties of singular values,

$$\tau_1 \geq \sigma_{n-m+1}, \qquad \tau_2 \geq \sigma_{n-m+2}, \qquad \ldots, \qquad \tau_{m-1} \geq \sigma_{n-1}, \qquad \tau_m \geq \sigma_n,$$

  with equality if $A = \begin{bmatrix} u_{n-m+1} & \cdots & u_n \end{bmatrix}^T$

- this choice of $A$ also minimizes

$$\|AX\|_F^2 = \tau_1^2 + \cdots + \tau_m^2$$

# $k$-means clustering with affine sets

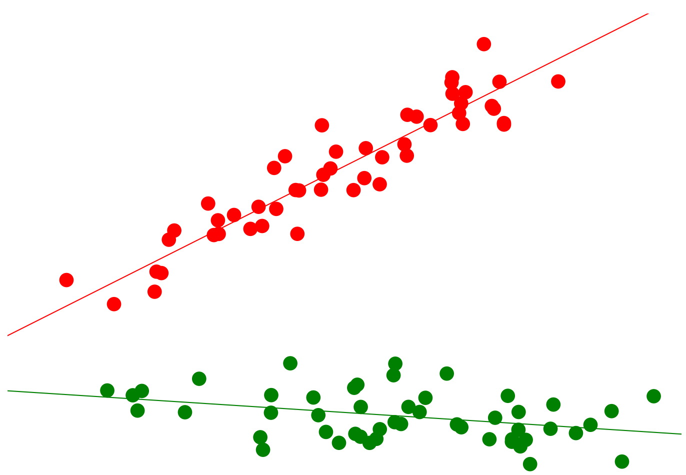
partition $N$ points $x_1, \ldots, x_N$ in $k$ classes

- in the $k$-means algorithm, clusters are represented by representative vectors $s_j$
- the $k$-means algorithm is a heuristic for minimizing the clustering objective

$$J^{\text{clust}} = \frac{1}{N} \sum_{i=1}^{N} \|x_i - s_{j_i}\|^2 \quad (j_i \text{ denotes index of the group point } i \text{ is assigned to})$$

by alternating minimization over assignment and over representatives

as an extension, we can use affine sets as representatives



# $k$-means clustering with affine sets

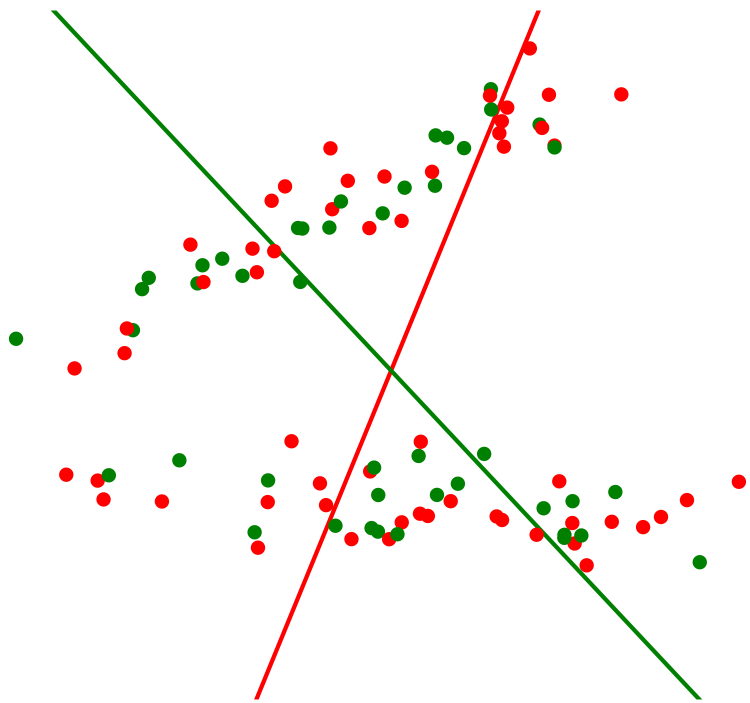
- represent the $k$ clusters by affine sets $\mathcal{S}_1, \ldots, \mathcal{S}_k$ of specified dimension
- use the $k$-means alternating minimization heuristic to minimize the objective

$$J^{\text{clust}} = \frac{1}{N} \sum_{i=1}^{N} d(x_i, \mathcal{S}_{j_i})^2 \quad (j_i \text{ denotes index of the group point } i \text{ is assigned to})$$

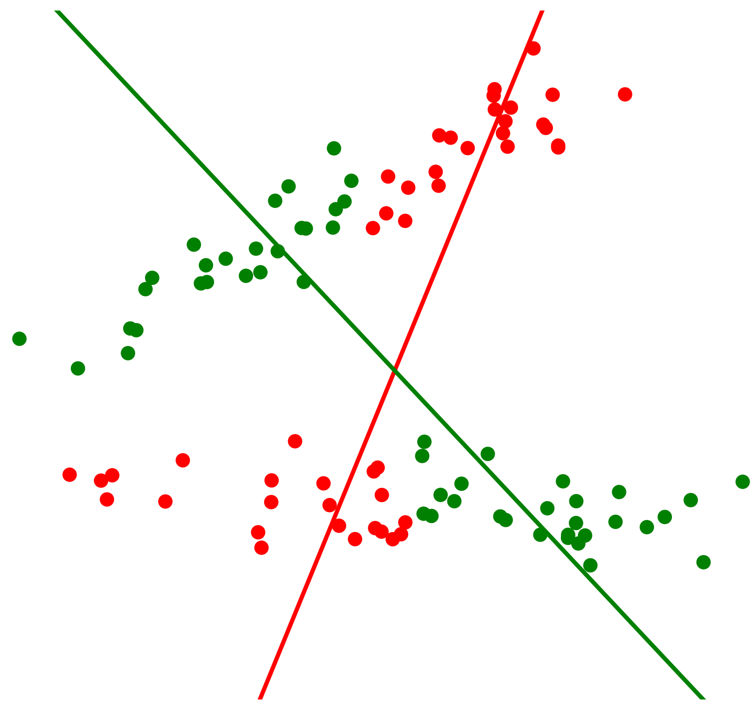
- to update partition we assign each point $x_i$ to nearest representative
- to update each group representative $\mathcal{S}_j$ we fit affine set to points in group $j$
- standard $k$-means is a special case with affine sets of dimension zero

# Example: iteration 1

we start with a random initial assignment

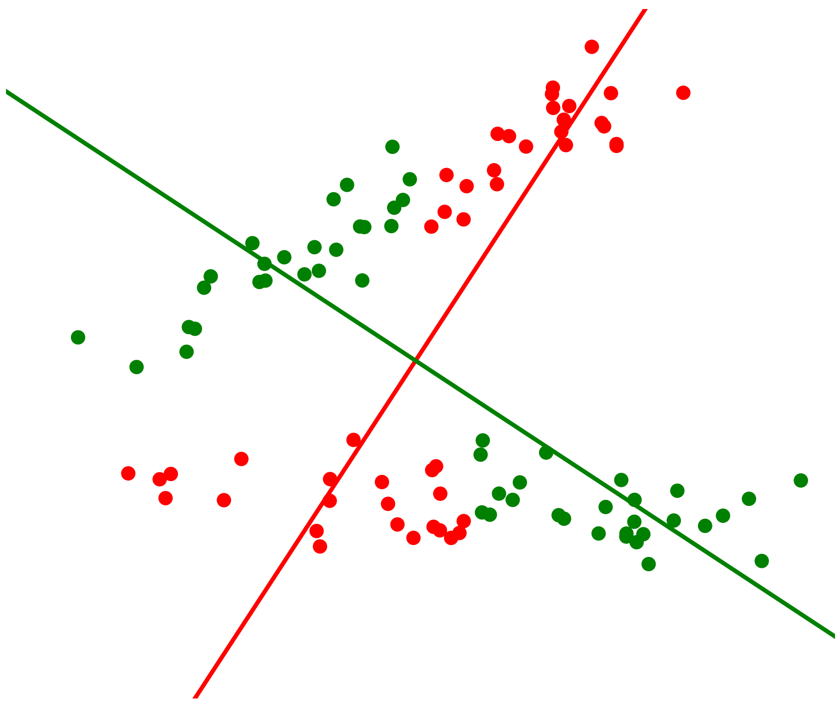


fit representatives to groups

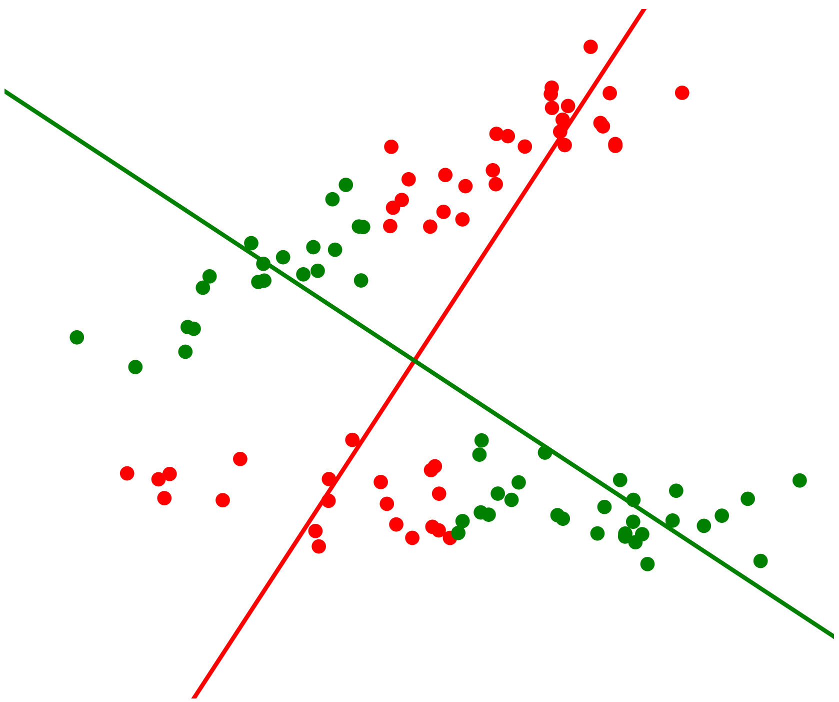


update assignment

# Example: iteration 2

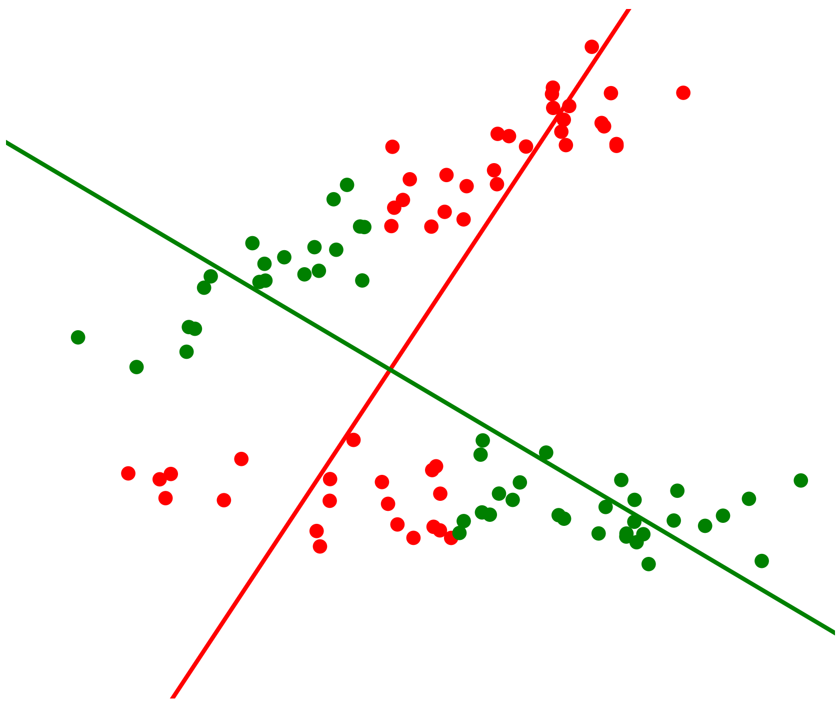


fit representatives to groups

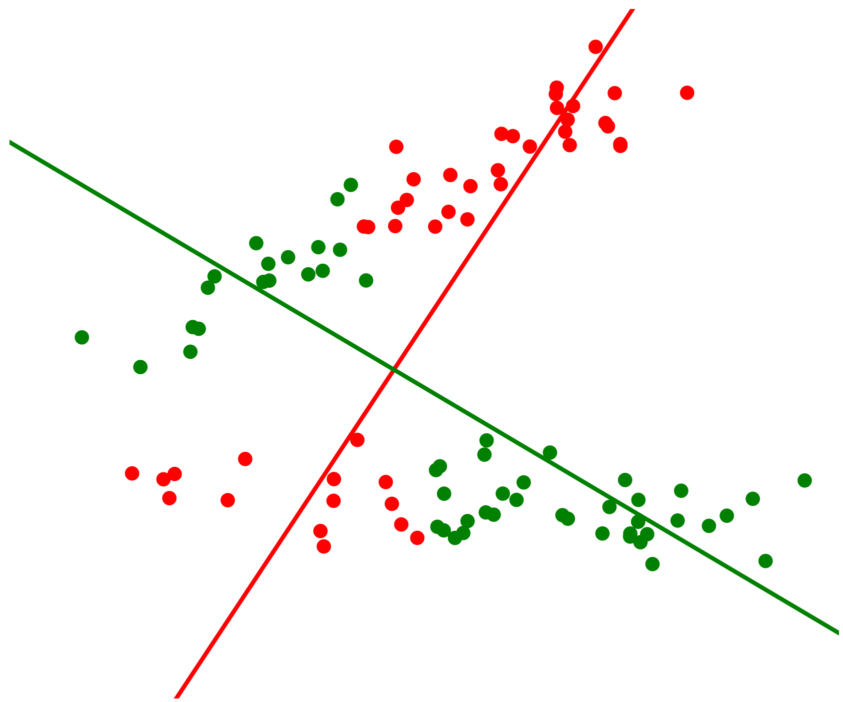


update assignment

# Example: iteration 3

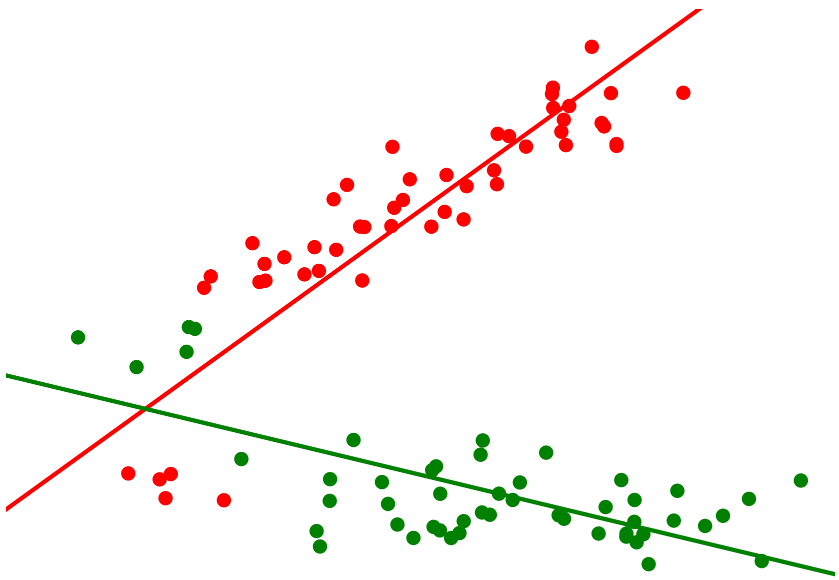


fit representatives to groups

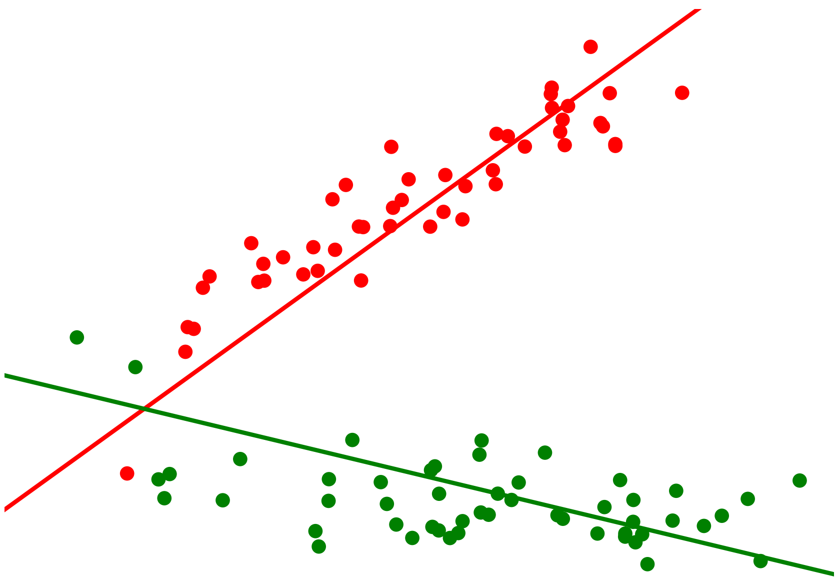


update assignment

# Example: iteration 8

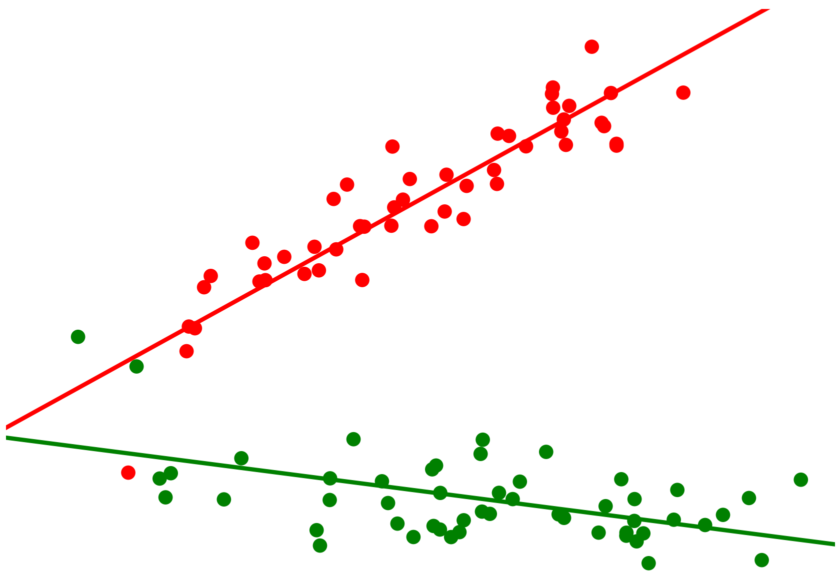


fit representatives to groups

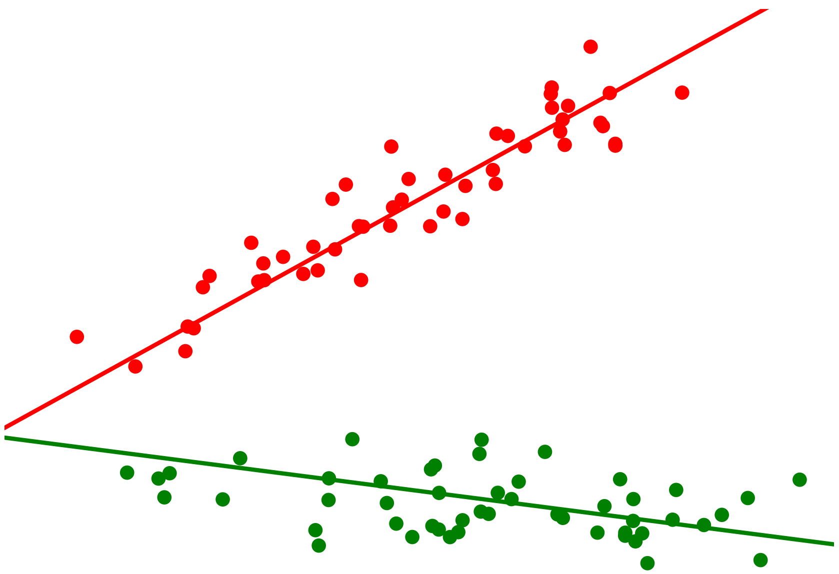


update assignment

# Example: iteration 9

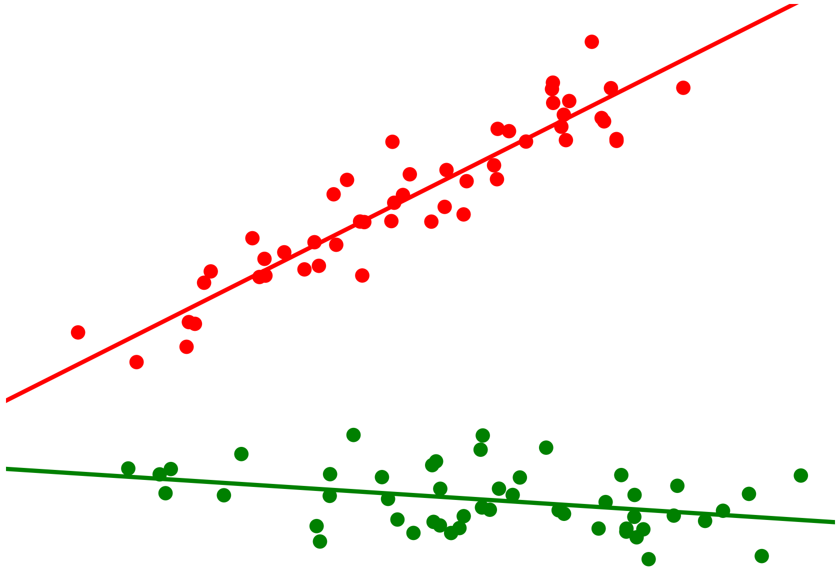


fit representatives to groups

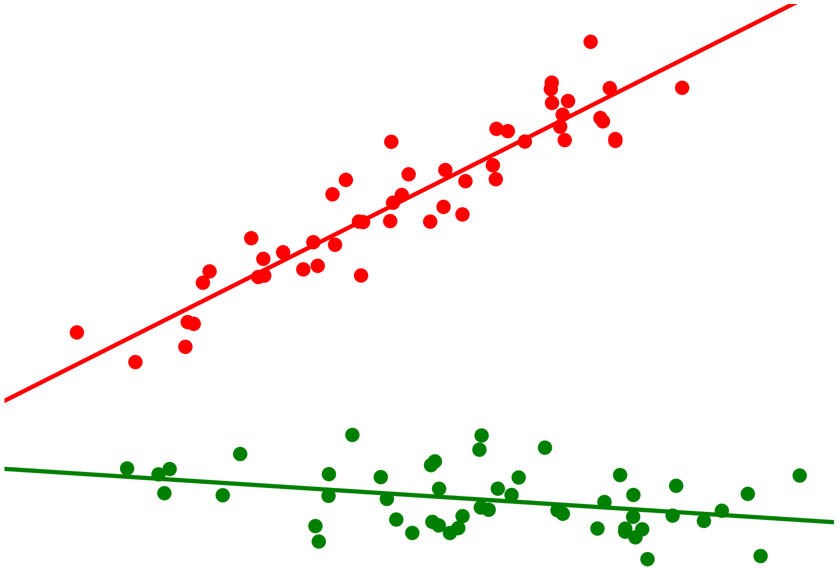


update assignment

# Example: iteration 10



fit representatives to groups



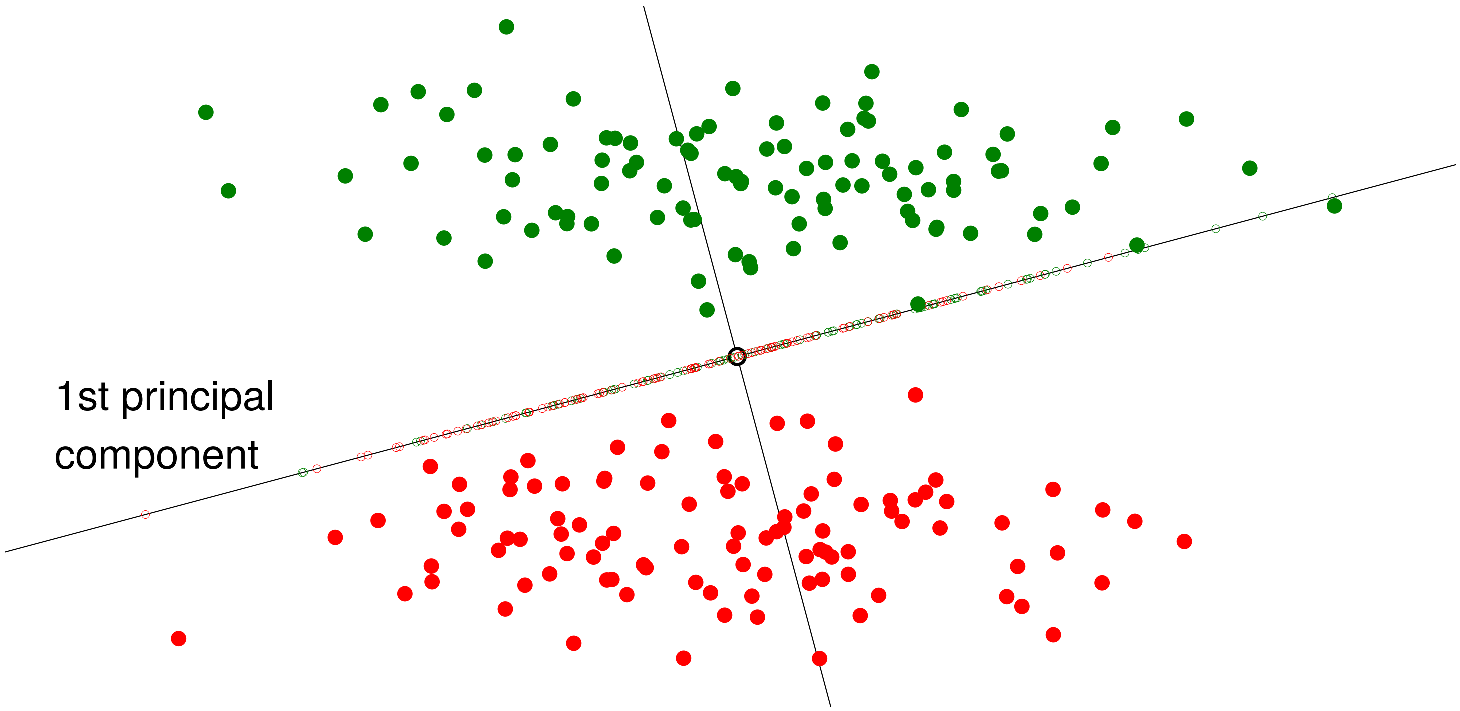
update assignment

# Outline

- localization from multiple camera views
- orthogonal Procrustes problem
- fitting affine sets to points
- **linear discriminant analysis**

# Motivation

principal components are not necessarily good features for classification




- the two sets of points (large dots) are linearly separable
- their projections on the 1st principal component direction (small circles) are not

# Classification problem

we are given a training set of examples of $K$ classes

| | |
|---|---|
| $C_k$: | set of examples for class $k$ |
| $N_k$: | number of examples for class $k$ |
| $C$: | set of all training examples $C = C_1 \cup \cdots \cup C_K$ |
| $N$: | total number of training examples $N = N_1 + \cdots + N_K$ |

- $\bar{x}_k$ denotes the mean for class $k$, $\bar{x}$ denotes the mean for the entire set:

$$\bar{x}_k = \frac{1}{N_k} \sum_{x \in C_k} x, \qquad \bar{x} = \frac{1}{N} \sum_{x \in C} x = \frac{1}{N}(N_1 \bar{x}_1 + \cdots + N_K \bar{x}_k)$$

- $S_k$ is the covariance matrix for class $k$:

$$S_k = \frac{1}{N_k} \sum_{x \in C_k} (x - \bar{x}_k)(x - \bar{x}_k)^T = \frac{1}{N_k} \sum_{x \in C_k} x x^T - \bar{x}_k \bar{x}_k^T$$

- $S$ is the covariance matrix for the entire set:

$$S = \frac{1}{N} \sum_{x \in C} (x - \bar{x})(x - \bar{x})^T = \frac{1}{N} \sum_{x \in C} x x^T - \bar{x} \bar{x}^T$$

# Principal components

the principal component directions are the eigenvectors of the covariance matrix

$$S = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

- can be defined recursively: $v_k$ solves

$$\begin{array}{ll} \text{maximize} & x^T S x \\ \text{subject to} & \|x\| = 1 \\ & v_i^T x = 0 \quad \text{for } i = 1, \ldots, k-1 \end{array}$$

- max–min characterization: the matrix of first $k$ eigenvectors $\begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix}$ solves

$$\begin{array}{ll} \text{maximize} & \lambda_{\min}(X^T S X) \\ \text{subject to} & X^T X = I_k \end{array}$$

PCA does not distinguish between variance within and between classes

# Within-class and between-class covariance

the covariance of the entire set can be written as a sum of two terms

$$S = S_{\mathrm{w}} + S_{\mathrm{b}}$$

**Within-class covariance**

$$S_{\mathrm{w}} = \sum_{k=1}^{K} \frac{N_k}{N} S_k = \frac{1}{N}\left(\sum_{x \in C} x x^T - \sum_{k=1}^{K} N_k \bar{x}_k \bar{x}_k^T\right)$$

- $S_{\mathrm{w}}$ is the weighted average of the class covariance matrices $S_k$
- describes the variability of points within the same class

**Between-class covariance**

$$S_{\mathrm{b}} = \frac{1}{N} \sum_{k=1}^{K} N_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T = \frac{1}{N} \sum_{k=1}^{K} N_k \bar{x}_k \bar{x}_k^T - \bar{x}\bar{x}^T$$

- $S_{\mathrm{b}}$ is the covariance matrix of the class means (weighted by class size)
- describes the variability between classes

# Linear discriminant analysis (LDA)

- good directions for classification make $v^T S_\mathrm{b} v$ large while keeping $v^T S_\mathrm{w} v$ small
- instead of maximizing $(v^T S v)/(v^T v)$ as in PCA, it is better to maximize

$$\frac{v^T S_\mathrm{b} v}{v^T S_\mathrm{w} v}$$

**LDA directions:** a sequence of vectors $v_1, v_2, \ldots$

- first direction $v_1$ maximizes $(x^T S_\mathrm{b} x)/(x^T S_\mathrm{w} x)$ or, equivalently, solves

$$\begin{array}{ll} \text{maximize} & x^T S_\mathrm{b} x \\ \text{subject to} & x^T S_\mathrm{w} x = 1 \end{array}$$

- other directions are defined recursively: $v_k$ is the solution $x$ of

$$\begin{array}{ll} \text{maximize} & x^T S_\mathrm{b} x \\ \text{subject to} & x^T S_\mathrm{w} x = 1 \\ & v_i^T S_\mathrm{w} x = 0 \quad \text{for } i = 1, \ldots, k-1 \end{array}$$

# Computation via eigendecomposition

the $k$th LDA direction $v_k$ is the solution $x$ of

$$\begin{array}{ll} \text{maximize} & x^T S_\mathrm{b} x \\ \text{subject to} & x^T S_\mathrm{w} x = 1 \\ & v_i^T S_\mathrm{w} x = 0 \quad \text{for } i = 1, \ldots, k-1 \end{array}$$

we assume $S_\mathrm{w}$ has full rank (is positive definite)

- compute Cholesky factorization $S_\mathrm{w} = R^T R$
- make a change of variables $y = Rx$:

$$\begin{array}{ll} \text{maximize} & y^T (R^{-T} S_\mathrm{b} R^{-1}) y \\ \text{subject to} & y^T y = 1 \\ & v_i^T R^T y = 0 \quad \text{for } i = 1, \ldots, k-1 \end{array}$$

the vectors $w_k = R v_k$ are the eigenvectors of $R^{-T} S_\mathrm{b} R^{-1}$

# Generalized eigenvectors

suppose $A$ and $B$ are symmetric, and $B$ is positive definite

- nonzero $x$ is a *generalized eigenvector* of $A, B$, with *generalized eigenvalue* $\lambda$, if

$$Ax = \lambda Bx$$

- via the Cholesky factorization $B = R^T R$ this can be written as

$$(R^{-T} A R^{-1})(Rx) = \lambda(Rx)$$

- generalized eigenvalues of $A, B$ are eigenvalues of $R^{-T} A R^{-1}$
- $x$ is a generalized eigenvector if and only if $Rx$ is eigenvector of $R^{-T} A R^{-1}$

LDA directions are generalized eigenvectors of $S_{\mathrm{b}}$, $S_{\mathrm{w}}$

# Number of LDA directions

the between-class covariance matrix has rank at most $K-1$

$$S_{\mathrm{b}} = \frac{1}{N}\sum_{k=1}^{K} N_k(\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T = \frac{1}{N}YY^T$$

where $Y$ is the $n \times K$ matrix

$$Y = \begin{bmatrix} \sqrt{N_1}\,(\bar{x}_k - \bar{x})^T \\ \vdots \\ \sqrt{N_K}\,(\bar{x}_K - \bar{x})^T \end{bmatrix}$$

the rank of $Y$ is at most $K-1$ because the rows of $Y$ are linearly dependent:

$$Y^T \begin{bmatrix} \sqrt{N_1} \\ \vdots \\ \sqrt{N_K} \end{bmatrix} = N_1\bar{x}_1 + N_2\bar{x}_2 + \cdots + N_K\bar{x}_K - (N_1 + \cdots + N_K)\bar{x} = 0$$

- therefore $R^{-T}S_{\mathrm{b}}R^{-1}$ has at most $K-1$ nonzero eigenvalues
- there are at most $K-1$ LDA directions (other directions are in $\mathrm{null}(S_{\mathrm{b}})$)

# LDA for Boolean classification ($K = 2$)

in the Boolean case, $\bar{x} = (N_1\bar{x}_1 + N_2\bar{x}_2)/N$ and

$$
\begin{aligned}
S_{\mathrm{b}} &= \frac{N_1}{N}(\bar{x}_1 - \bar{x})(\bar{x}_1 - \bar{x})^T + \frac{N_2}{N}(\bar{x}_2 - \bar{x})(\bar{x}_2 - \bar{x})^T \\
&= \frac{2N_1N_2}{N^2}(\bar{x}_1 - \bar{x}_2)(\bar{x}_1 - \bar{x}_2)^T
\end{aligned}
$$

- the LDA direction $v$ is defined as the solution $x$ of

$$
\begin{array}{ll}
\text{maximize} & x^T S_{\mathrm{b}} x \\
\text{subject to} & x^T S_{\mathrm{w}} x = 1
\end{array}
$$

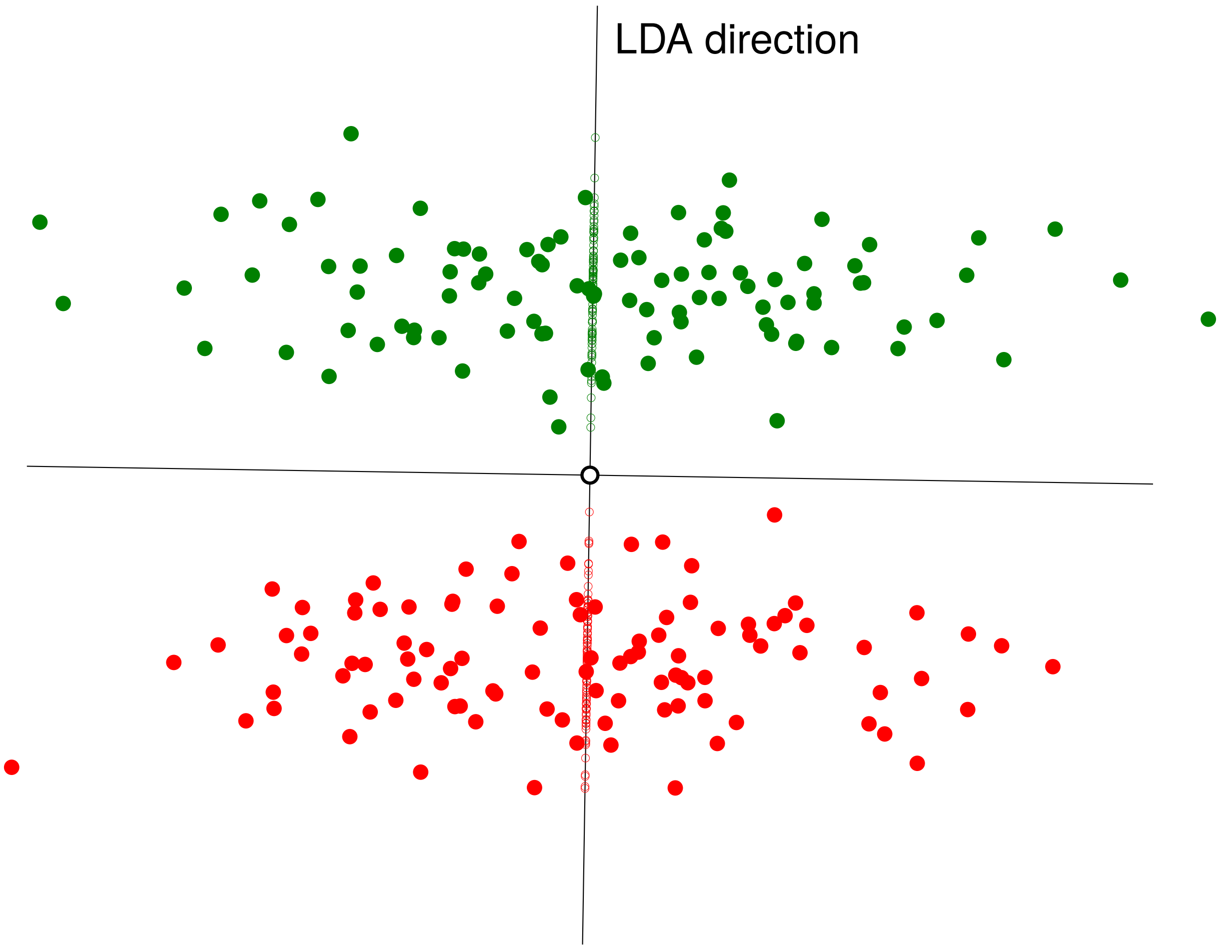
- via the change of variable $y = Rx$, where $S_{\mathrm{w}} = R^T R$, we find the solution

$$
y = \frac{R^{-T}(\bar{x}_1 - \bar{x}_2)}{\|R^{-T}(\bar{x}_1 - \bar{x}_2)\|}, \qquad v = R^{-1}y = \frac{S_{\mathrm{w}}^{-1}(\bar{x}_1 - \bar{x}_2)}{((\bar{x}_1 - \bar{x}_2)^T S_{\mathrm{w}}^{-1}(\bar{x}_1 - \bar{x}_2))^{1/2}}
$$

the LDA direction is the direction of $S_{\mathrm{w}}^{-1}(\bar{x}_1 - \bar{x}_2)$

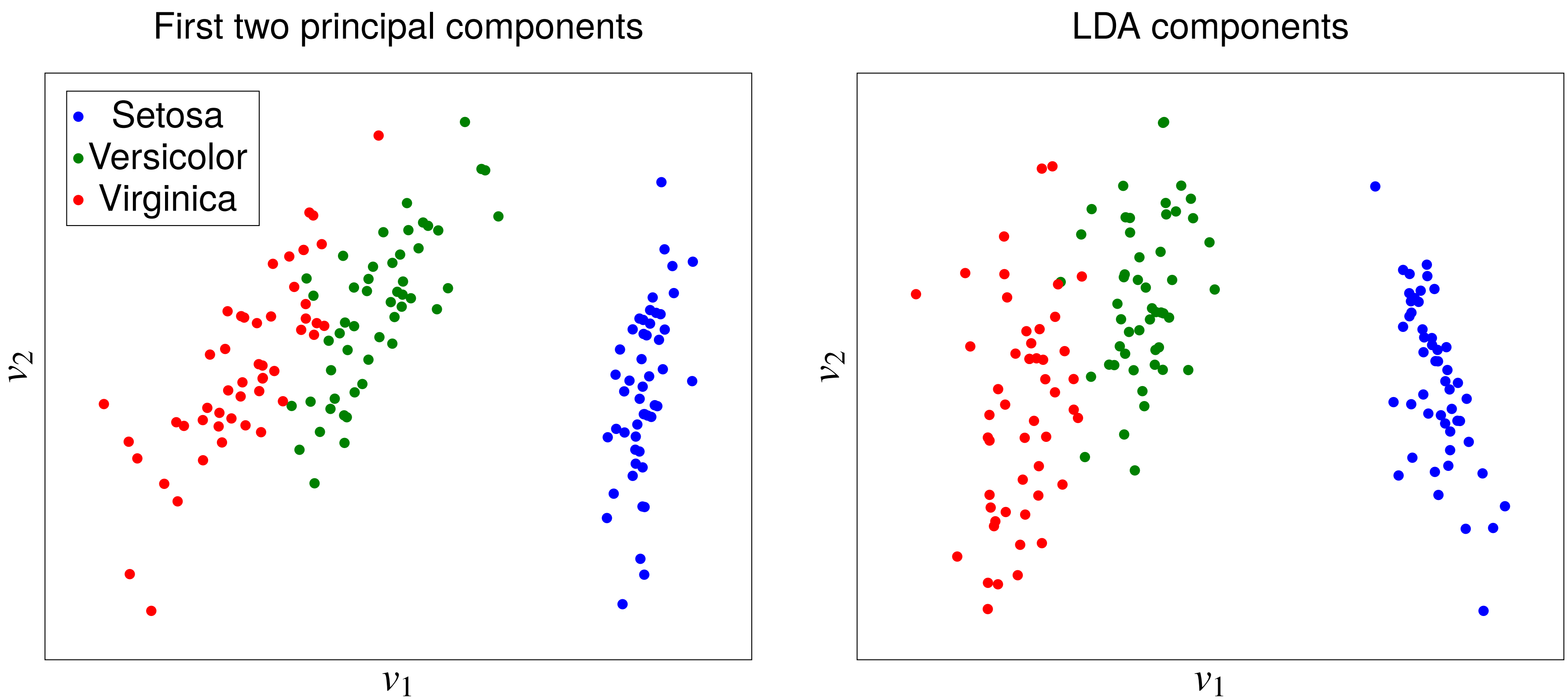
# Example

the example of page 6.27




projections on LDA direction (small circles) are separable

# Fisher's Iris flower data set




- 50 examples of each of the three classes, 4 features
- first LDA direction separates the classes better than first PCA direction
- second LDA direction does not add much information
- eigenvalues of $R^{-T}S_bR^{-1}$ are $(32.19, 0.29, 0, 0)$ (see page 6.32)

# Reference

- Peter N. Belhumeur, João P. Hespanha, David J. Kriegman, *Eigenfaces vs. Fisherfaces: recognition using class specific linear projection*, IEEE Transactions on Pattern Analysis and Machine Intelligence (1997).

  discusses PCA and LDA for face recognition