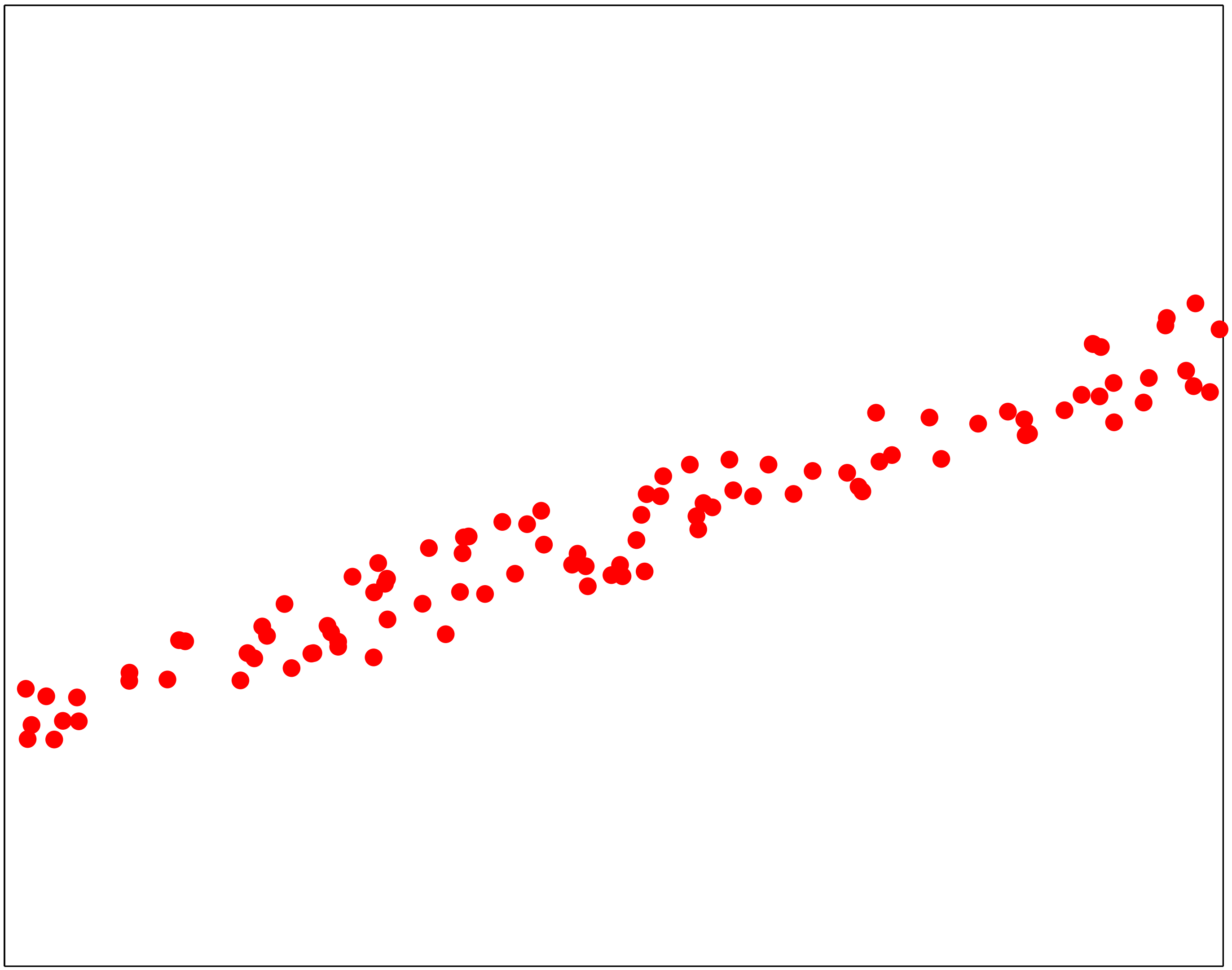
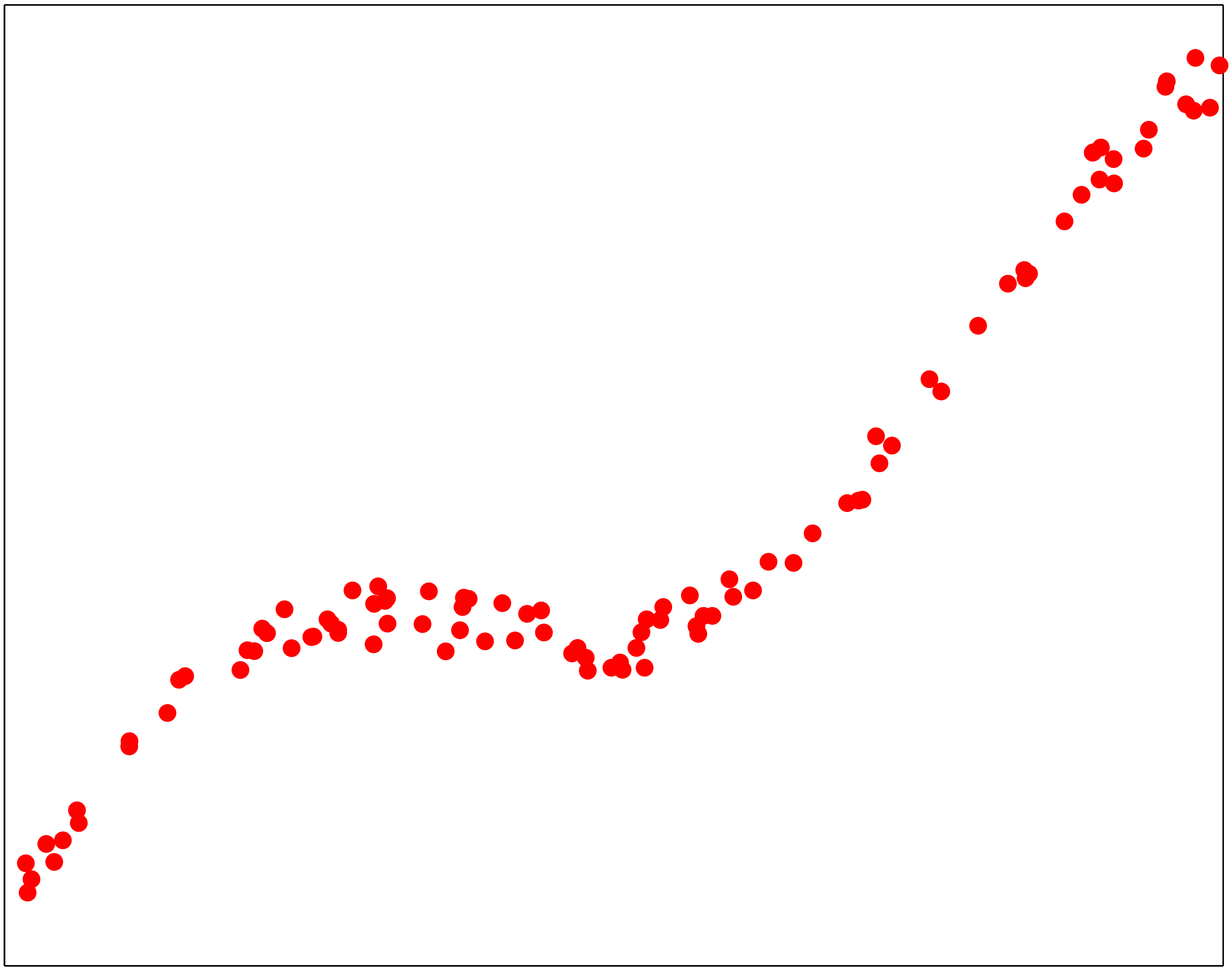
What is dimensionality reduction?

- Mapping data objects to (short) real vectors
- For visualization, comparison, outlier detection
- For further machine learning
- Some techniques:
  - Principal components analysis (linear)
  - Independent components analysis (linear or nonlinear)
  - Self-organizing maps (nonlinear)
  - Multi-dimensional scaling (nonlinear, allows non-numeric data objects)

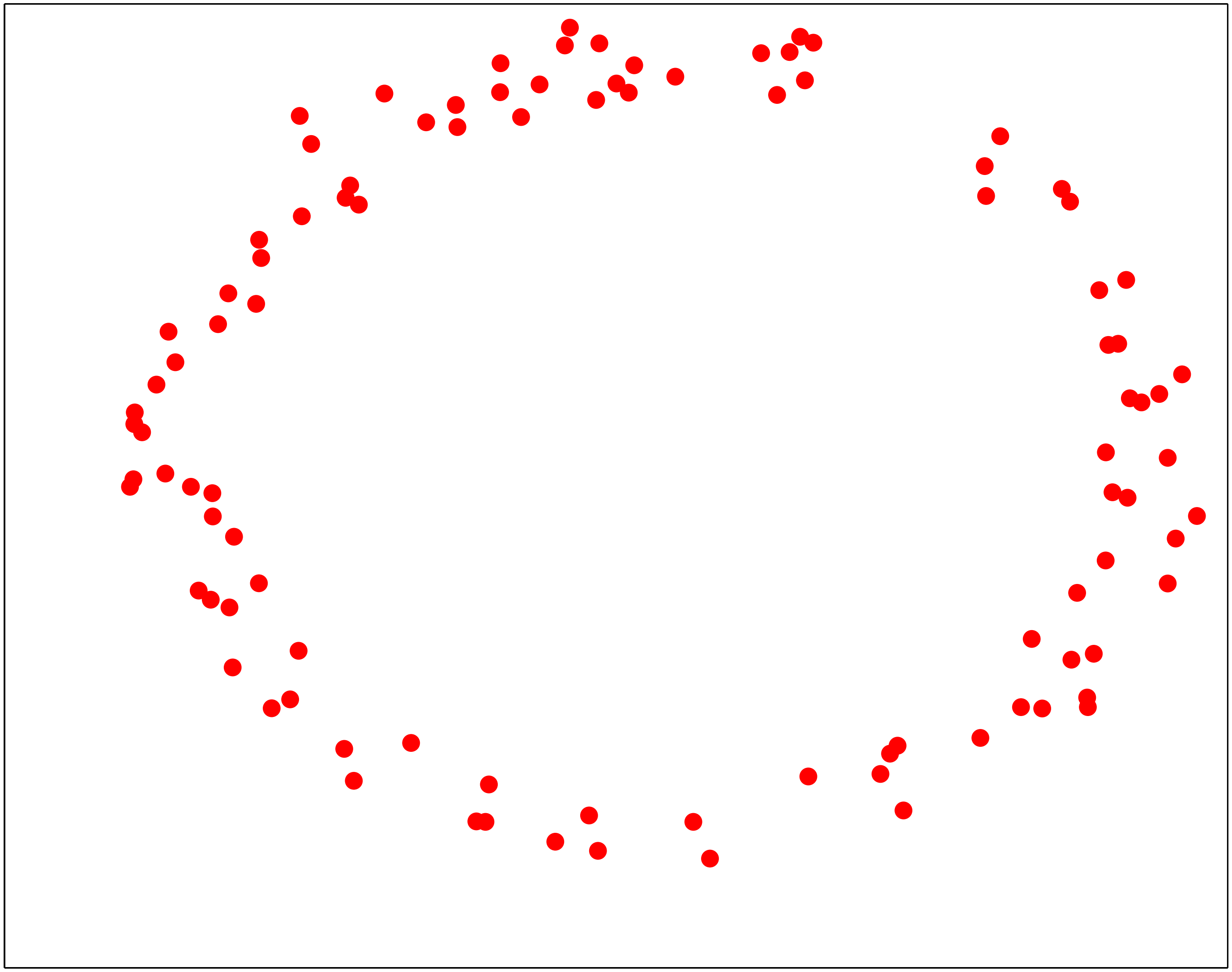
Good case



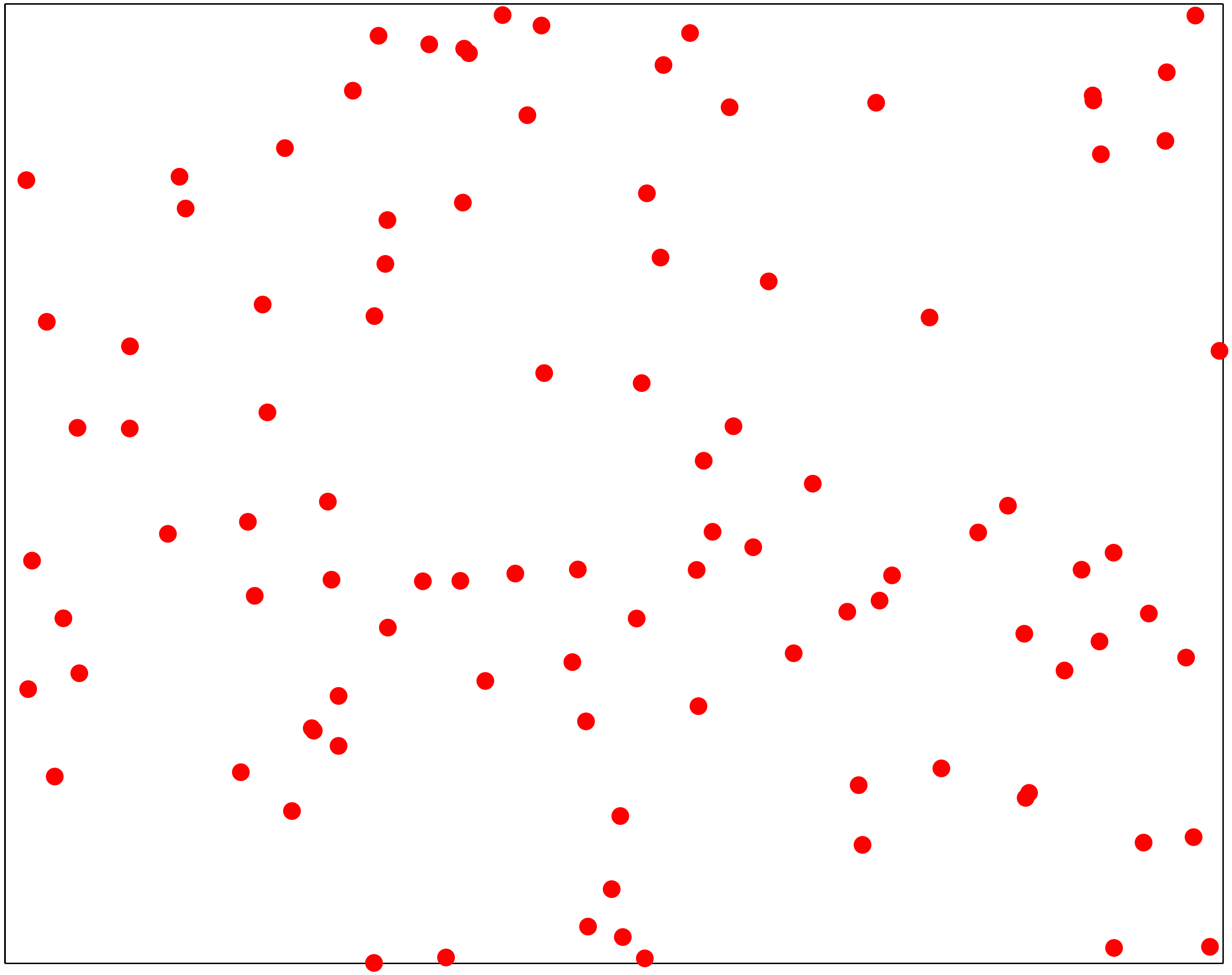
Not too bad case



Hard case



Forget it!



Today

- Reviewing some basic stats
- Principal components analysis
- Refs for today's material:
  - Duda, Hart, Stork pp. 114–117
  - Hastie, Tibshirani, Friedman pp. 485–491

Reviewing some basic stats

Expected value, sample average

- For a numeric random variable X, the expected value (mean) is

$$E(X) = \sum_x x\mathrm{P}(X = x) \quad \text{or} \quad \int_x x\mathrm{p}(x)dx \quad \text{or} \quad \int_x x d\mathrm{p}(x)$$

- If we take m samples from the same distribution/density, $x_1, \ldots, x_n$, then the sample average

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

is an unbiased estimated of $E(X)$.

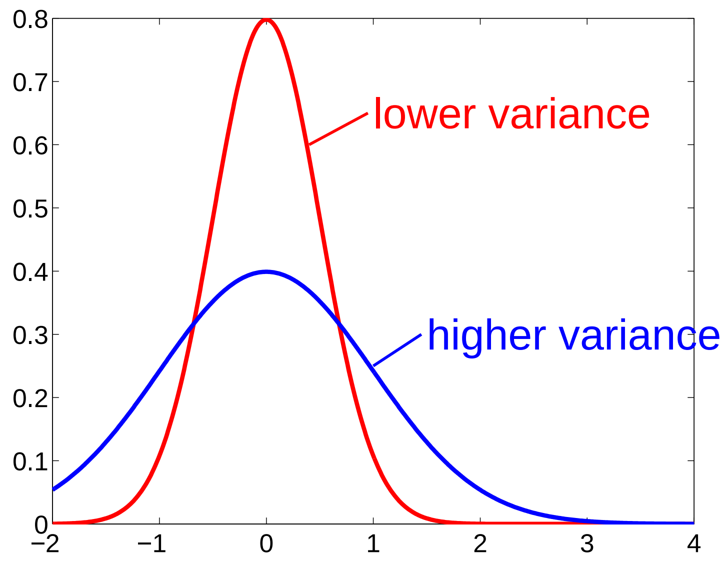
(That is, $E\left(\frac{1}{m}\sum_{i=1}^{m} x_i\right) = E(X)$.)

Variance

- The variance of X is

$$Var(X) = E(X^2 - (E(X))^2) = E(X^2) - (E(X))^2$$

- The variance of X is non-negative and captures how "spread out" X's distribution is.



Estimating variance

- The sample variance is sometimes

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

  where $\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$.

- It turns out that this underestimates the true variance by a factor of $(m-1)/m$.

- An alternative definition of sample variance,

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

  is an unbiased estimator of $Var(X)$.

Covariance

- Covariance quantifies a linear relationship (if any) between two random variables X and Y.

$$Cov(X,Y) = E\{(X - E(X))(Y - E(Y))\}$$

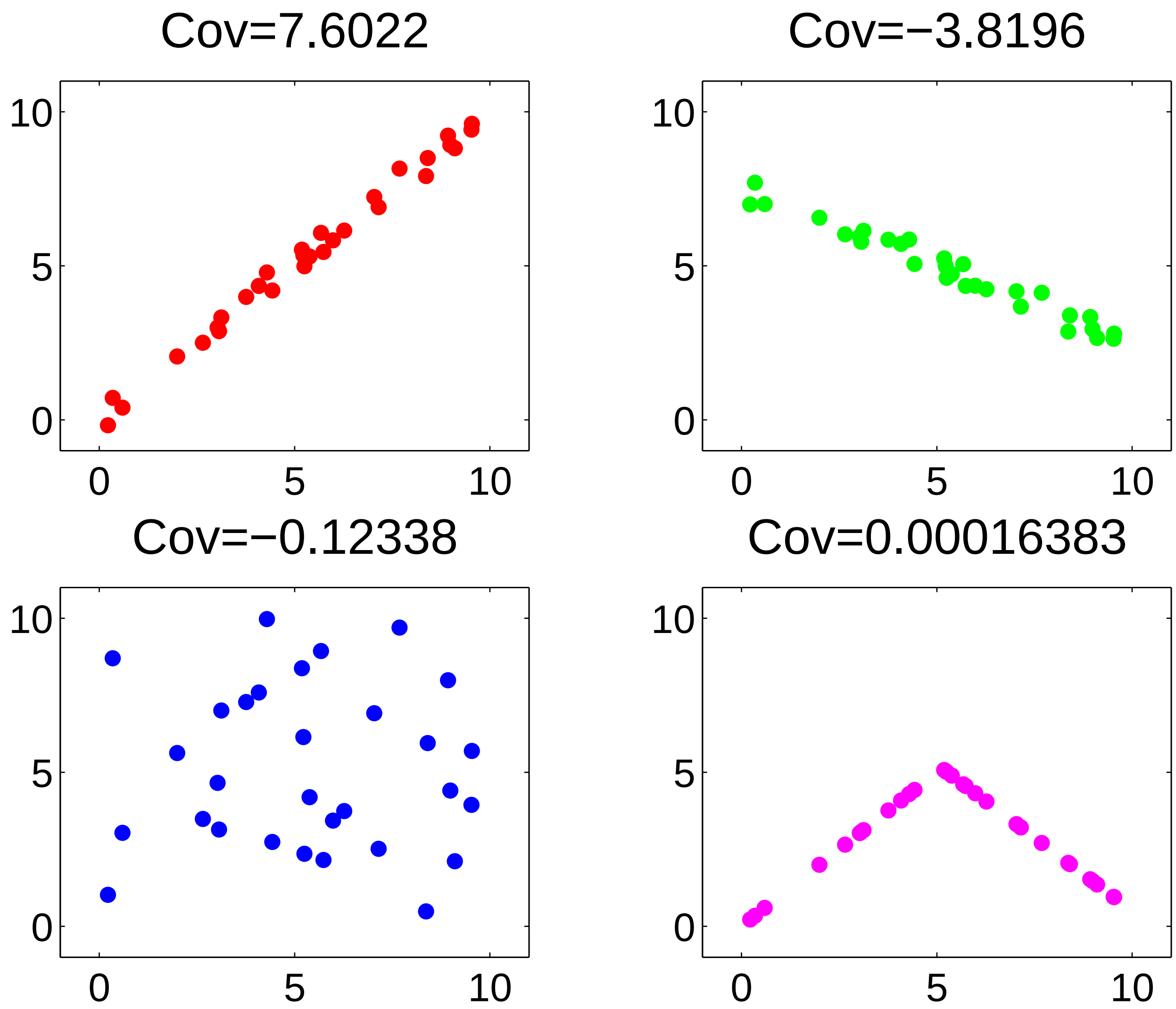
- Given m samples of X and Y, covariance can be estimated as

$$\frac{1}{m-1}\sum_{i=1}^{m}(x_i - \mu_X)(y_i - \mu_Y) \; ,$$

  where $\mu_X = \sum_{i=1}^{m} x_i$ and $\mu_Y = \sum_{i=1}^{m} y_i$.

- Note: $Cov(X,X) = Var(X)$.

Examples — all on the same scale

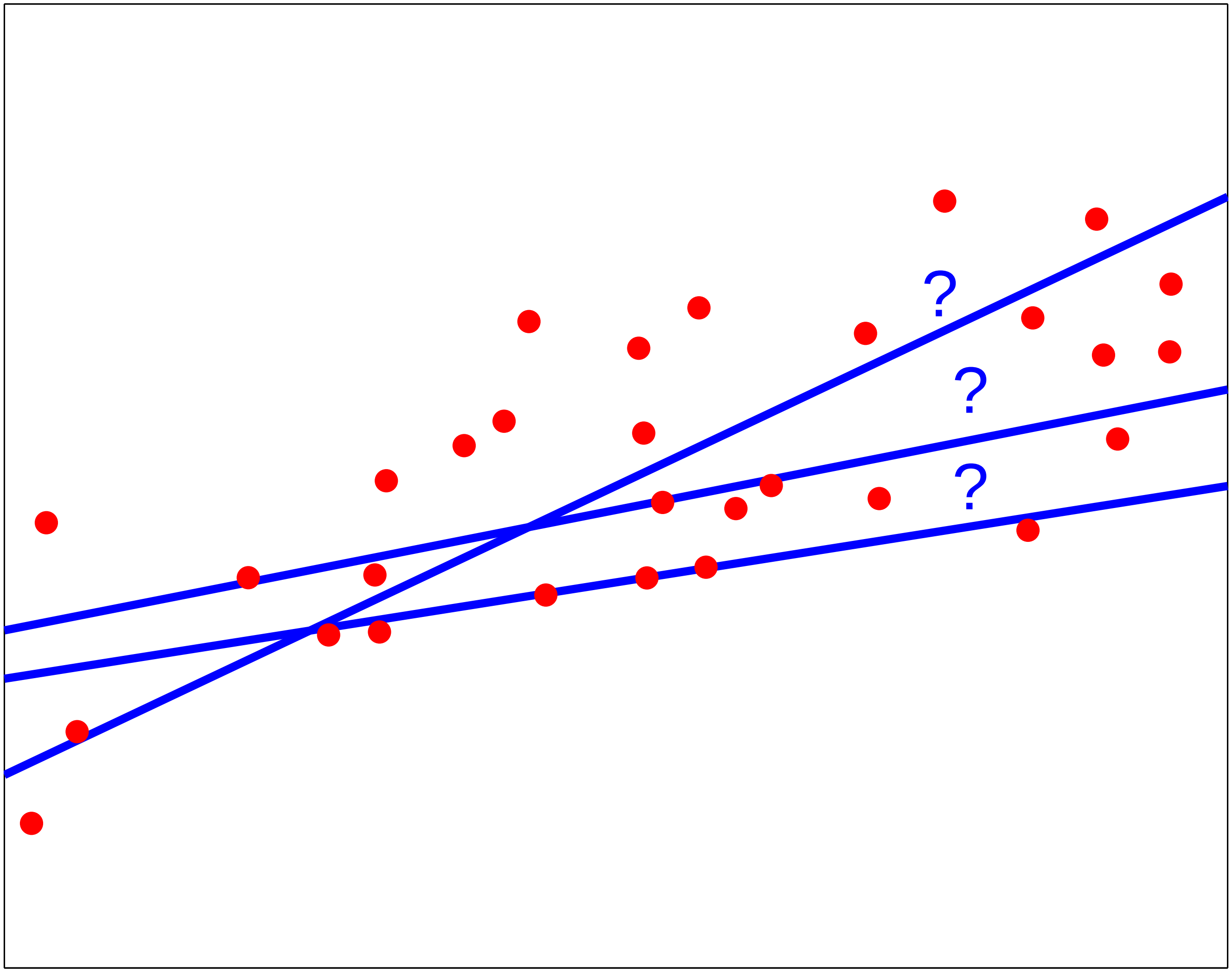


Principal components analysis

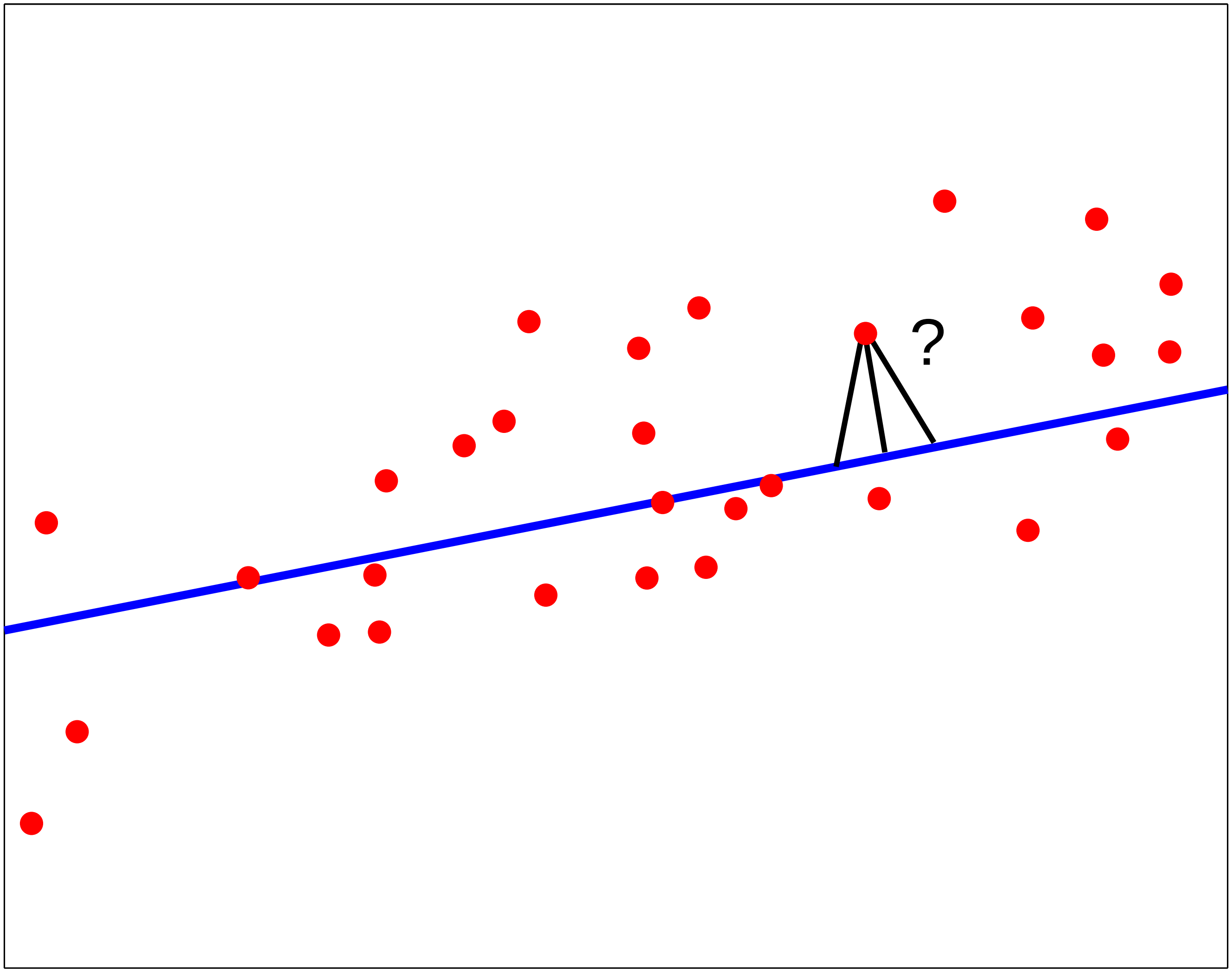
PCA for reduction to 1D

- Given: m data objects, each a length-n real vector.
- Suppose we want a 1-dimensional representation of that data, instead of n-dimensional.
- Specifically, we will:
  - Choose a line in $\Re^n$ that "best represents" the data.
  - Assign each data object to a point along that line.

Which line is best?



How do we assign points to lines?



Reconstruction error

- Let our line be represented as $b + \alpha v$ for $b, v \in \Re^n$, $\alpha \in \Re$. For later convenience, assume $\|v\| = 1$.
- Each data vector x_i is assigned a point on the line $\hat{x}_i = b + \alpha_i v$.
- The (squared Euclidean) reconstruction error for data object i is

$$\|x_i - \hat{x}_i\|^2 = \sum_{j=1}^{n} (x_i(j) - \hat{x}_i(j))^2$$

$\Rightarrow$ Choose b, v, and the α_i to minimize the total reconstruction error over all data points:

$$R = \sum_{i=1}^{m} \|x_i - \hat{x}_i\|^2$$

Minimizing reconstruction error

- Suppose we fix v. A little calculus reveals that (an) optimal choice for b is

$$b = \frac{1}{m} \sum_{i=1}^{m} x_i \ ,$$

and for any α_i,

$$\alpha_i = v \cdot (x_i - b)$$

So $\hat{x}_i = b + v \cdot (x_i - b)$.

Minimizing reconstruction error: b and the α_i

- Suppose we fix v. A little calculus reveals that (an) optimal choice for b is

$$b = \frac{1}{m} \sum_{i=1}^{m} x_i \ ,$$

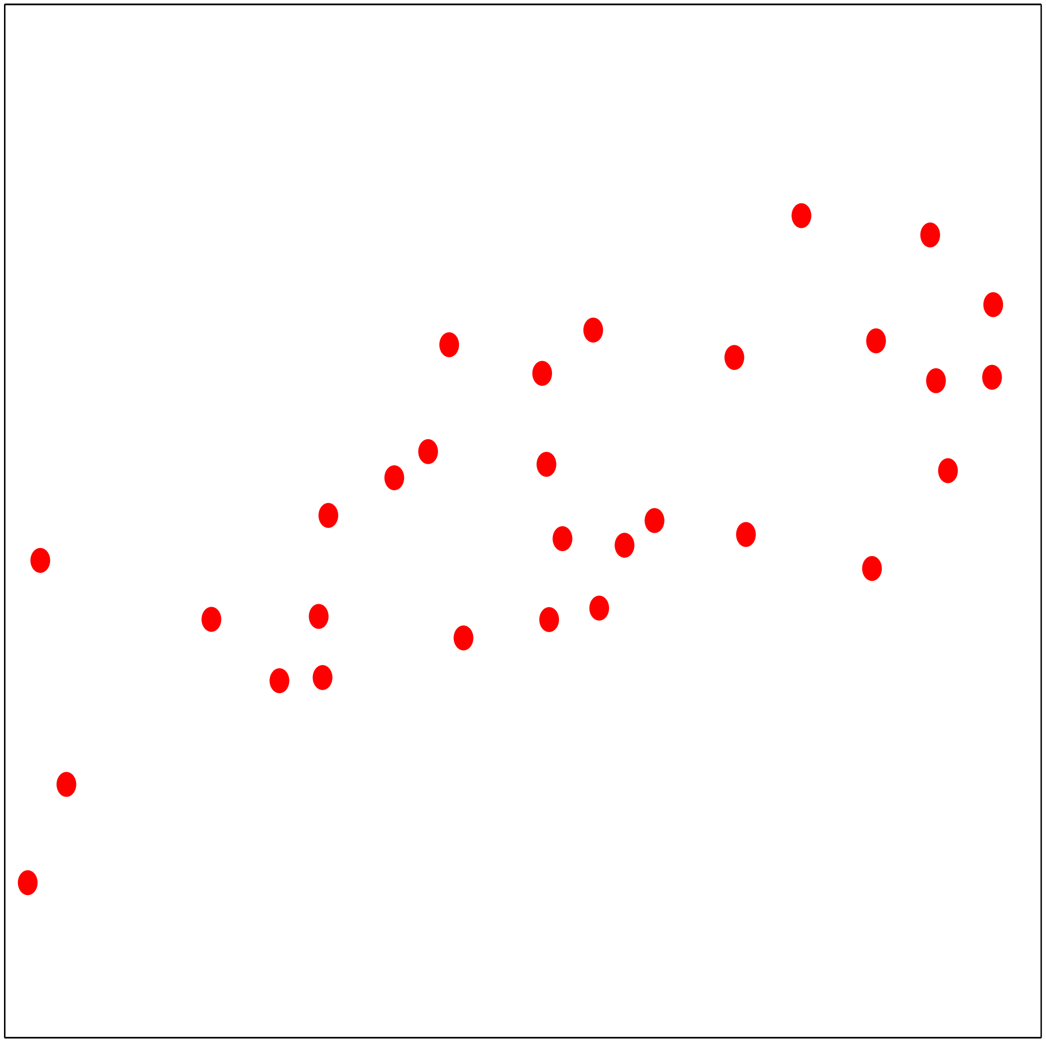
and for any α_i,

$$\alpha_i = v \cdot (x_i - b)$$

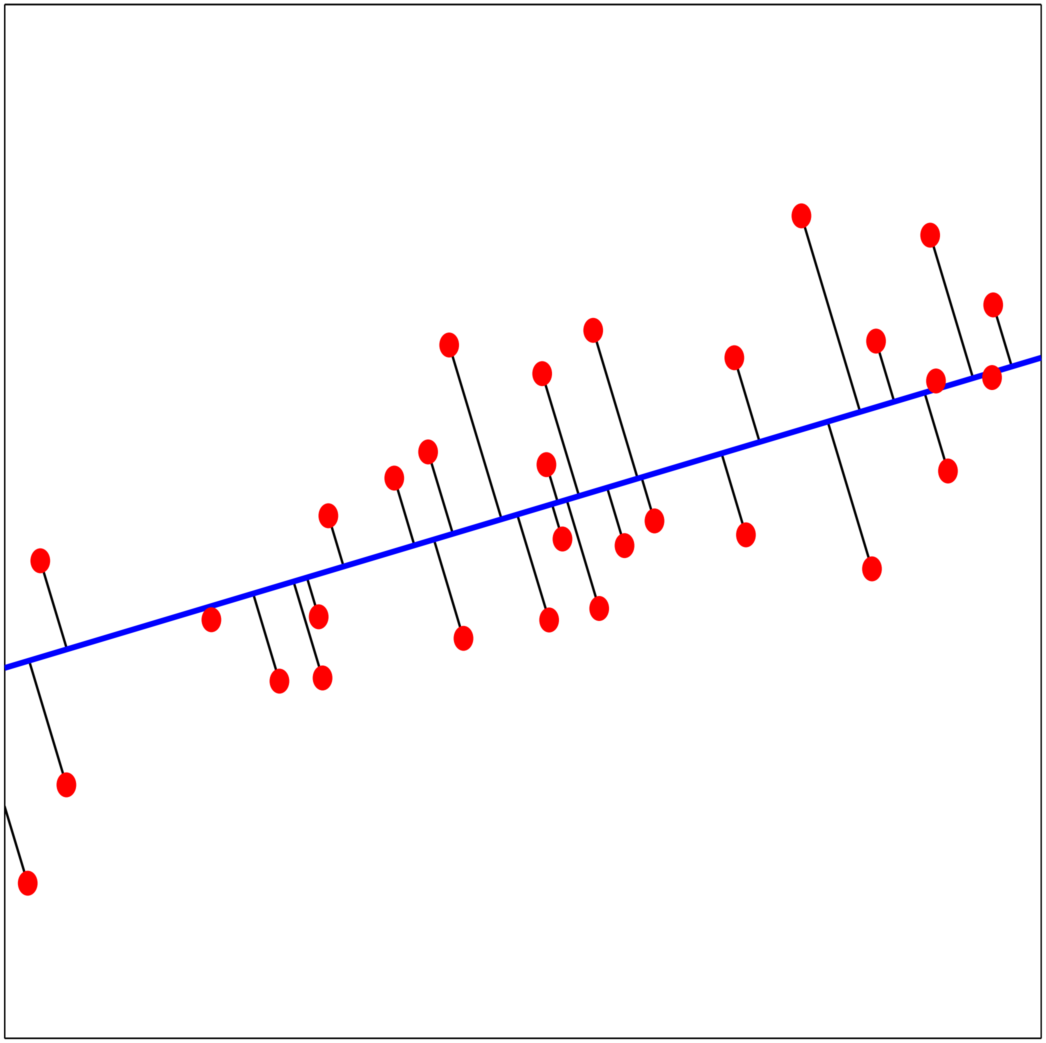
So $\hat{x}_i = b + v \cdot (x_i - b)$.

- Intuitively:
  - The line goes through the centroid of the data.
  - Data points are mapped to the point on the line closest to them in Euclidean distance. (They are *projected onto* the line.)

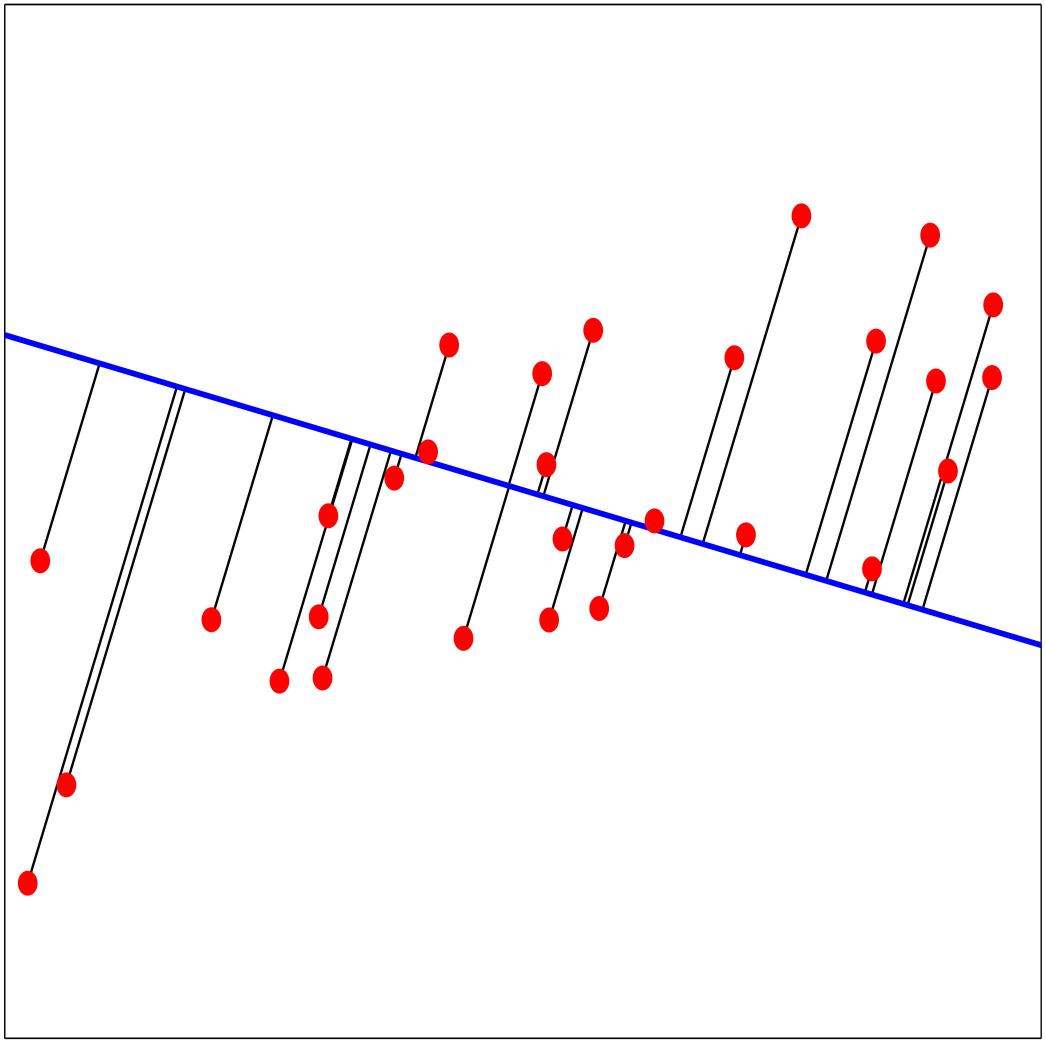
Example data



Example with $v \propto (1, 0.3)$



Example with $v \propto (1, -0.3)$



Minimizing reconstruction error: the scatter matrix

- Substituting back into the formula for R shows v should maximize

$$v^T S v \ ,$$

where S is an $n \times n$ matrix with

$$S(k,l) = \sum_{i=1}^{m} (x_i(k) - b(k))(x_i(l) - b(l))$$

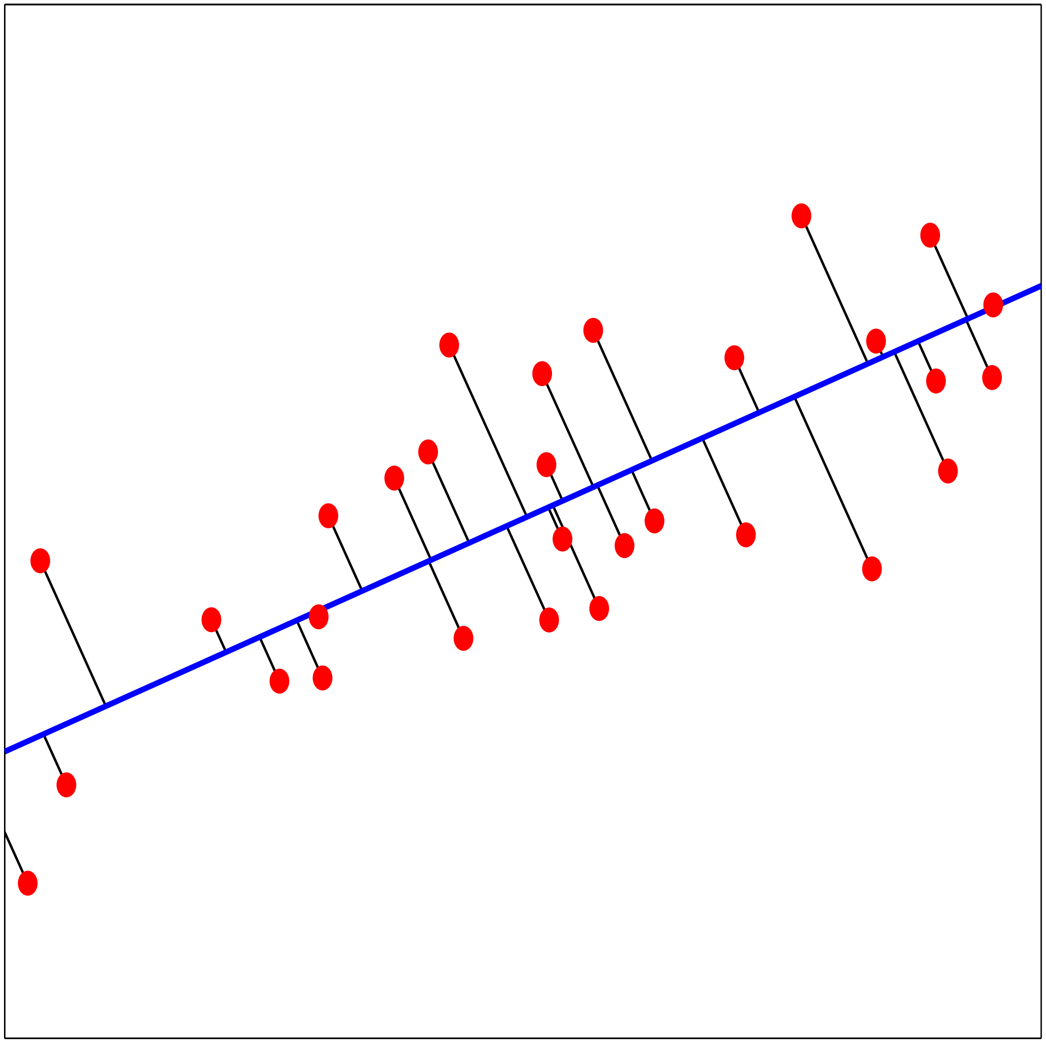
- S(k,l) is proportional to the estimated covariance between element k and element l in the data.

- S is the *scatter matrix*.

Optimal choice of v

- Recall: an *eigenvector* u of a matrix A satisfies $Au = \lambda u$, where $\lambda \in \Re$ is the *eigenvalue*.
- Fact: the scatter matrix, S, has n non-negative eigenvalues and n orthogonal eigenvectors.
- The v that maximizes $v^T S v$ is the eigenvector of S with the largest eigenvalue.

Example with optimal line: $b = (0.54, 0.52)$, $v \propto (1, 0.45)$



Comments

- The line $b + \alpha v$ is the *first principal component*.
- The variance of the data along the line $b + \alpha v$ is as large as along any other line.
- b, v, and the α_i can be computed in polynomial time.

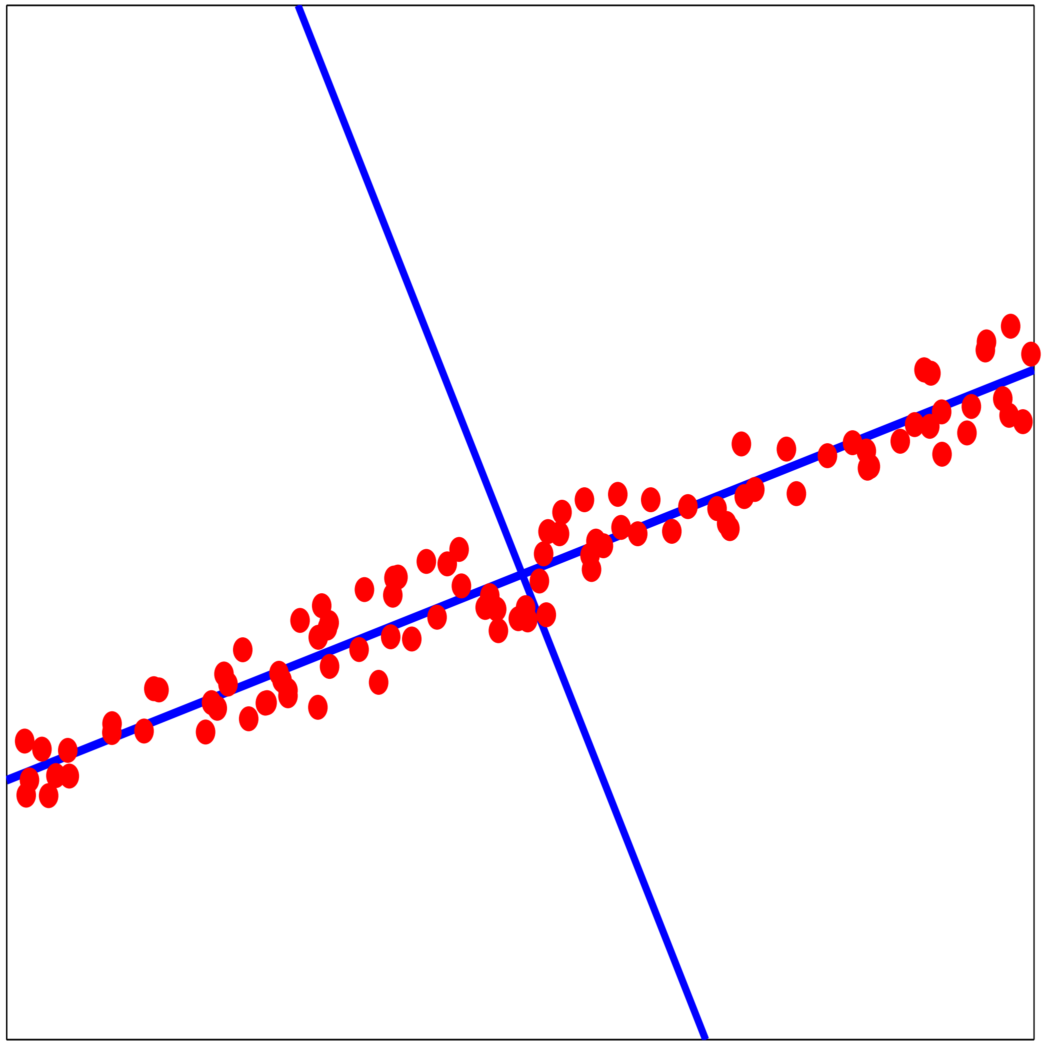
Reduction to d dimensions

- More generally, we can create a d-dimensional representation of our data by projecting our data points onto a hyperplane $b + \alpha^1 v_1 + \ldots + \alpha^d v_d$.
- If we assume the v_j are of unit length and orthogonal, then the optimal choices are:
  - b is the centroid of the data (as before)
  - The v_j are orthogonal eigenvectors of S corresponding to S's d-largest eigenvalues.
  - Each data point is assigned to the nearest (in Euclidean distance) point on the hyperplane.

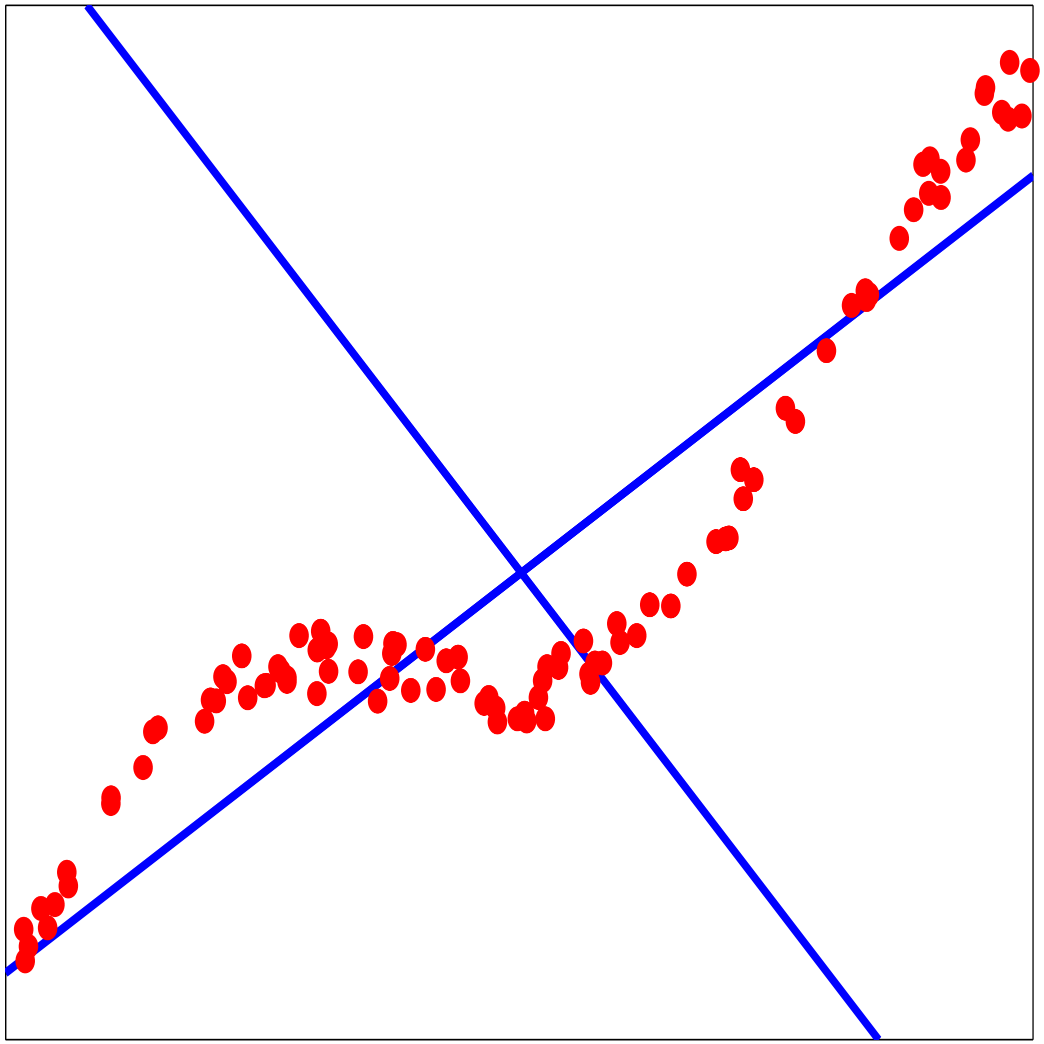
Comments

- b, the v_j (and the corresponding eigenvalues), and the projections of the data points can all be computing in polynomial time.
- The magnitude of the j^{th}-largest eigenvalue, λ_j, tells you how much variability in the data the j^{th} principal component captures — giving you feedback on how to choose d!

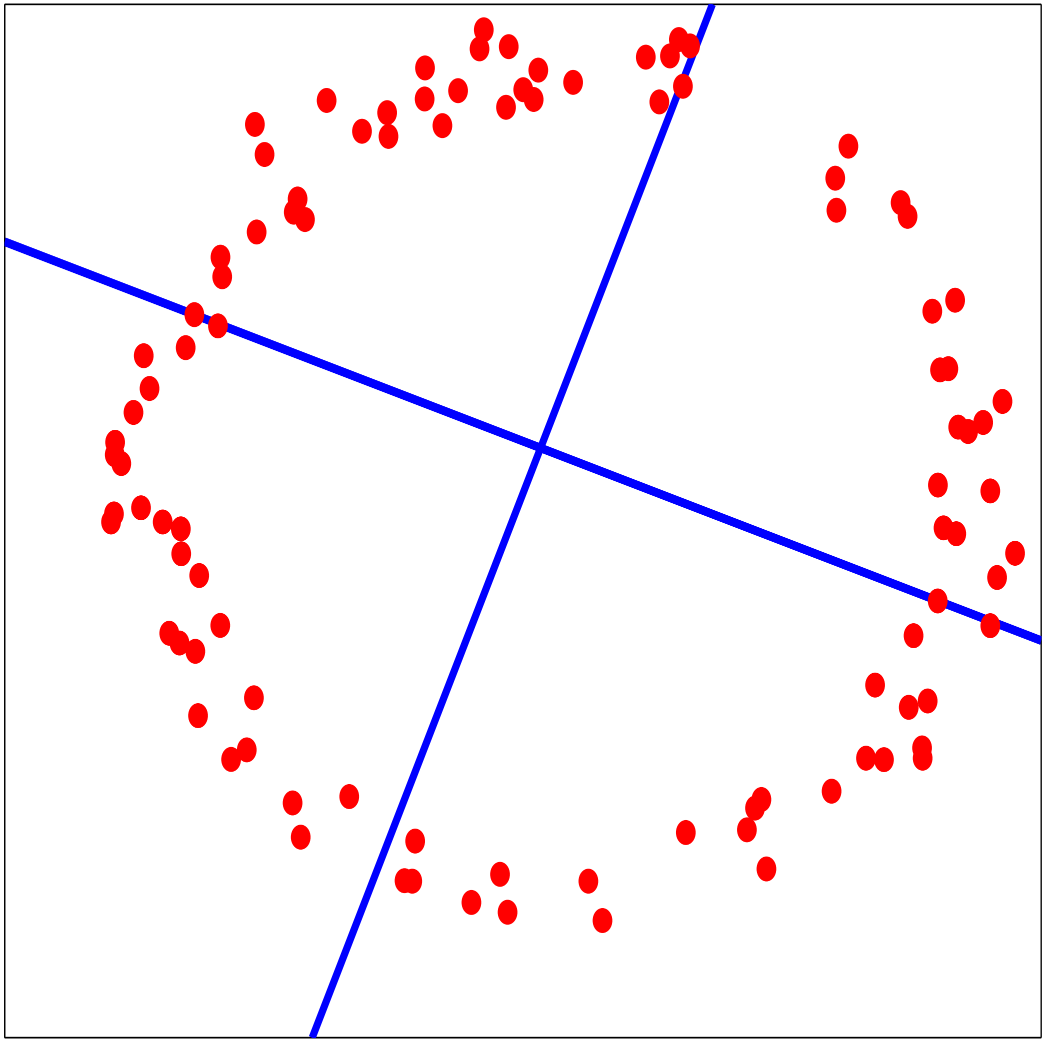
$\underline{\lambda_1 = 0.0938,\ \lambda_2 = 0.0007}$



$\lambda_1 = 0.1260, \lambda_2 = 0.0054$



$\lambda_1 = 0.0884, \lambda_2 = 0.0725$



$$\lambda_1 = 0.0881, \lambda_2 = 0.0769$$