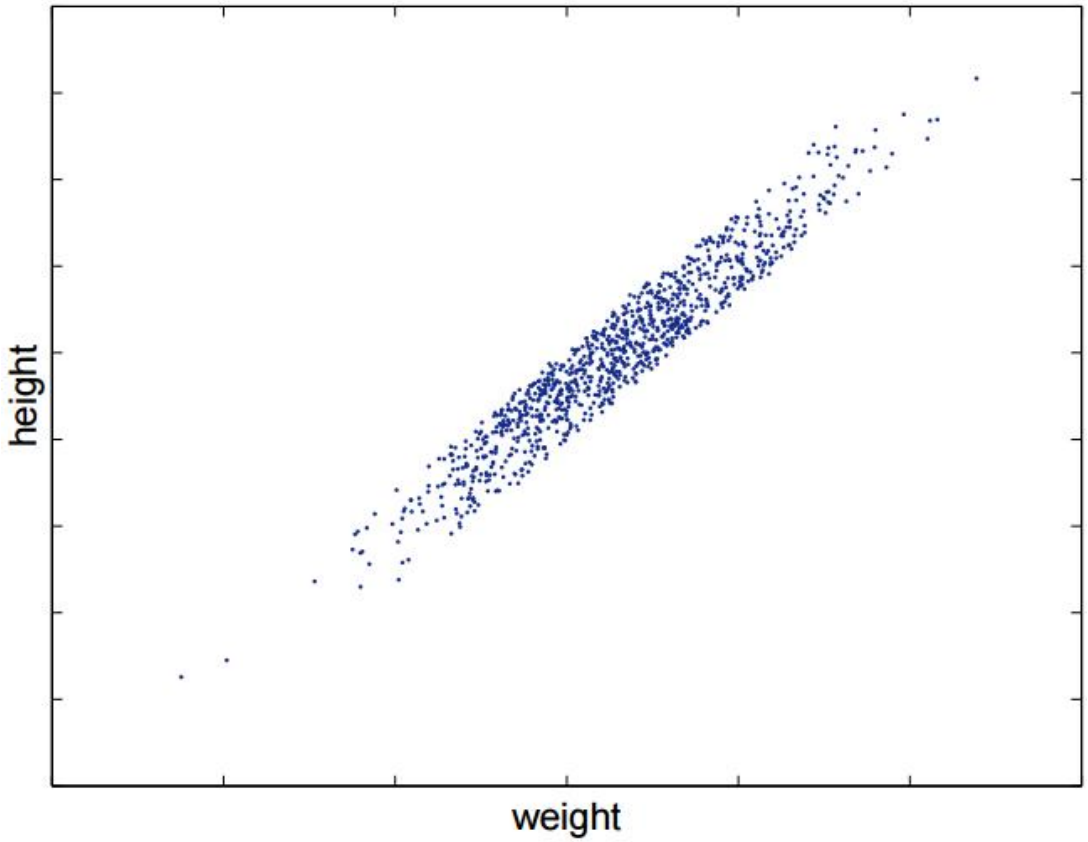
Nedostaci PCA: Interpretacija

- Obeležja dobijena pomoću PCA metode su linearne kombinacije originalnih obeležja. Ovo je često teško interpretirati

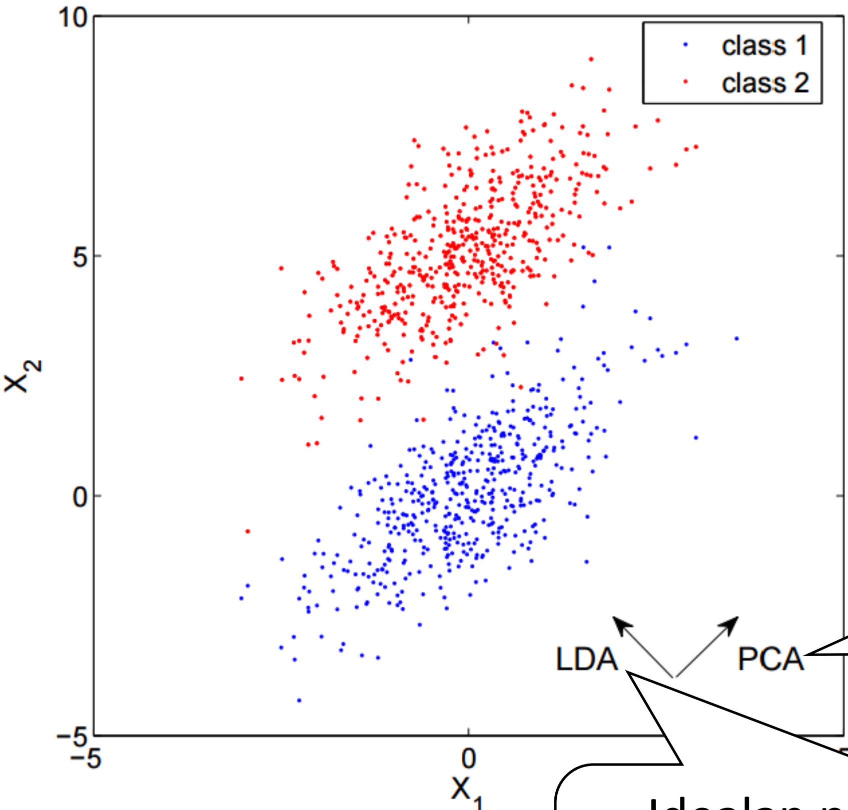


Težinu i visinu menjamo samo jednom koordinatom.

Kako interpretirati ovu novu dimenziju, odnosno, pridodati joj neko fizičko značenje?

Nedostaci PCA: Bez garancije da će pomoći

- Nema garancije da će biti korisna prilikom rešavanja problema koji pokušavamo da rešimo



Ako je naš cilj da klasifikujemo tačke, projekcija bi činila više štete nego dobra jer bi se većina crvenih i plavih tačaka preklapala u novoj dimenziji

PCA bi prvu glavnu komponentu postavio u pravcu maksimalne varijanse

Idealan pravac projekcije u ovom slučaju je normalan na prvu glavnu komponentu

LDA (Linear Discriminant Analysis)

- Još jedan način da se objasni zašto PCA ne radi u prethodnom primeru jeste što PCA ne koristi informaciju o klasnom obeležju (PCA je nenadgledana metoda)

- LDA je nadgledan metod koji služi istovremeno za redukciju dimenzionalnosti i klasifikaciju
  - Kao i PCA, LDA rezultuje linearnom projekcijom podataka
  - Za razliku od PCA, LDA prilikom projekcije koristi klasno obeležje

LDA (Linear Discriminant Analysis)

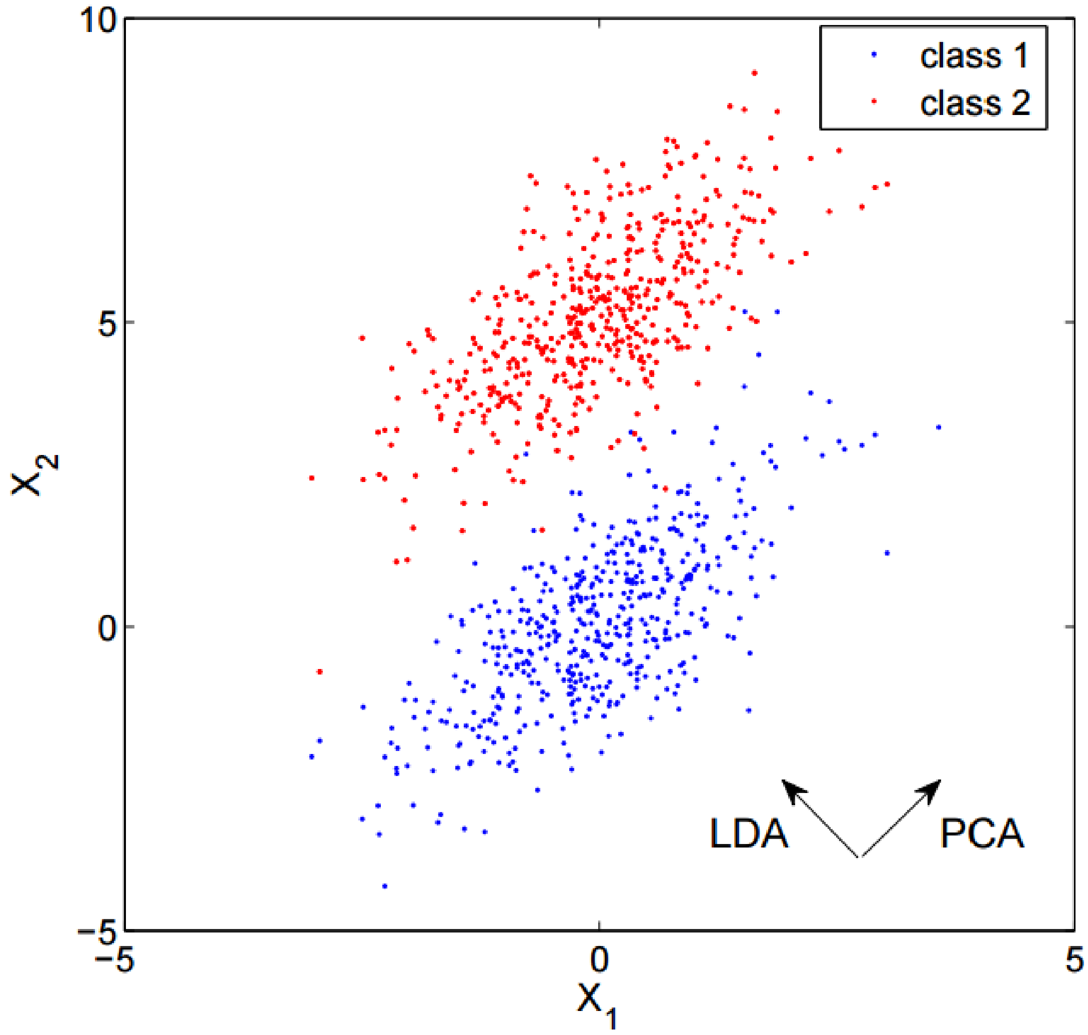
- LDA pronalazi projekciju koja maksimizuje sledeći odnos:

$$R = \frac{v^T \mathrm{C}_b v}{v^T C_i v}$$

  - v vektor projekcije,
  - $C_b = \sum_c (\mu_c - \mu)(\mu_c - \mu)^T$ - varijabilnost između klasa
  - $C_i = \sum_c \sum_{n_c} \left(x_{n_c} - \mu_c\right)\left(x_{n_c} - \mu_c\right)^T$ - varijabilnost u klasi c
  - μ_c - srednja vrednost u klasi c, a x_{n_c}- primer n iz klase c

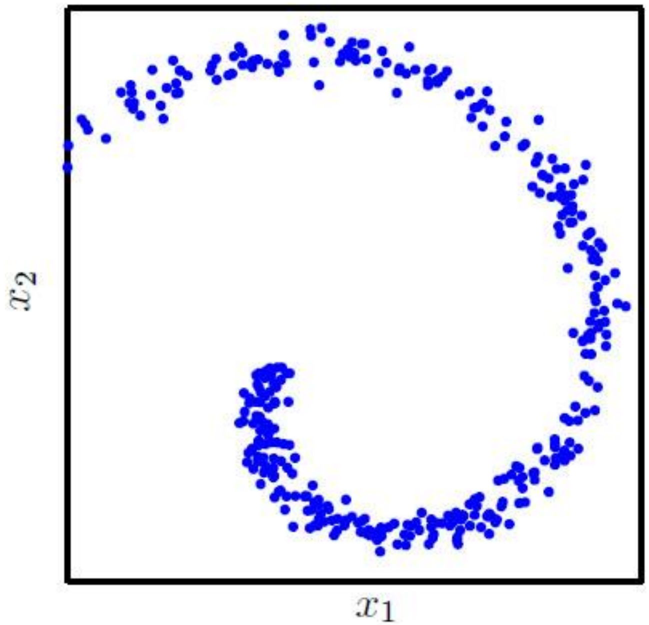
- Intuitivno, ova projekcija teži da maksimizuje razlike između klasa, a da minimizuje varijabilnost u svakoj klasi pojedinačno

LDA (Linear Discriminant Analysis)

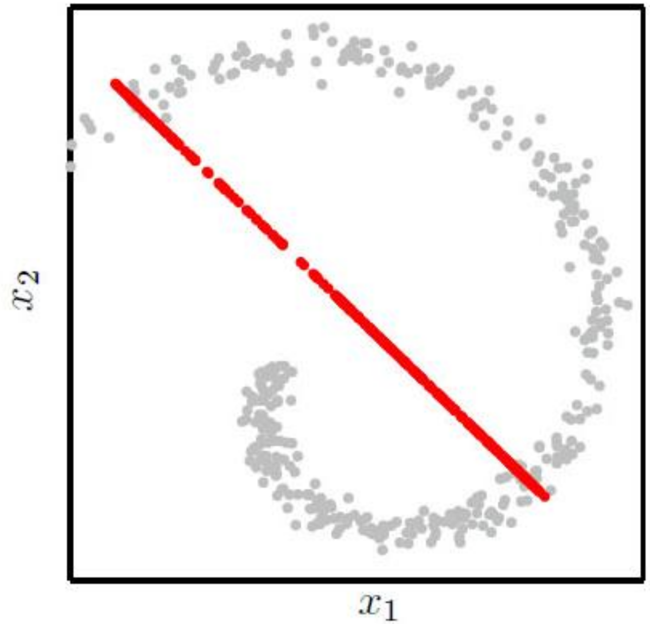


Napomena: maksimalan broj dimenzija dobijen LDA projekcijom je $C - 1$, gde je C broj klasa

Nedostaci PCA: Linearnost



(a) Data in $\mathcal{X}$ space



(b) Top-1 PCA reconstruction

- Podaci (a) približno leže na jednodimenzionoj površi (krivoj)
- Međutim, ako pokušamo da rekonstruišemo ove podatke pomoću PCA, rezultati su katastrofalni (b)
- Razlog je što kriva nije linearna, a PCA može da kreira samo linearna obeležja

Checklist za primenu PCA

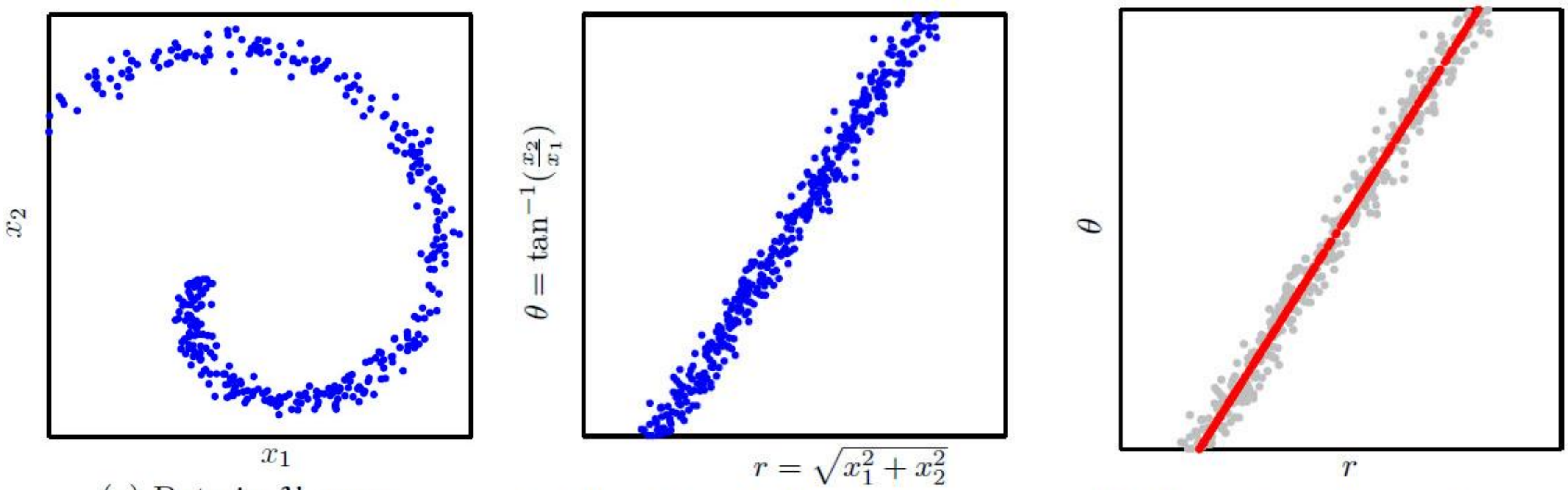
1. Da li očekujete da podaci imaju linearnu strukturu
   - Na primer, da li podaci leže u gotovo linearnom potprostoru?

2. Da li donje glavne komponente sadrže uglavnom male slučajne fluktuacije koje liče na šum i treba da budu odbačene?
   - Činjenicu da su fluktuacije male možemo odrediti pomoću greške rekonstrukcije
   - Činjenica da se radi o šumu je ne više od nagađanja

Checklist za primenu PCA

3. Da li ciljna funkcija f zavisi prvenstveno od gornjih glavih komponenti ili su male fluktuacije u donjim glavnim komponentama ključne?
   - Ako je ovo drugo slučaj, PCA neće pomoći u rešavanju datog problema
   - U praksi, teško je proveriti da li je ovo tačno
   - Validacijom možemo proveriti da li koristiti PCA ili ne
   - Obično, odbacivanje donjih komponenti ne odbacuje informacije bitne sa aspekta f
   - A ono što je odbačeno je nadoknađeno smanjenom greškom generalizacije (zbog redukovane dimenzionalnosti)

Jedno rešenje: nelinearna transformacija

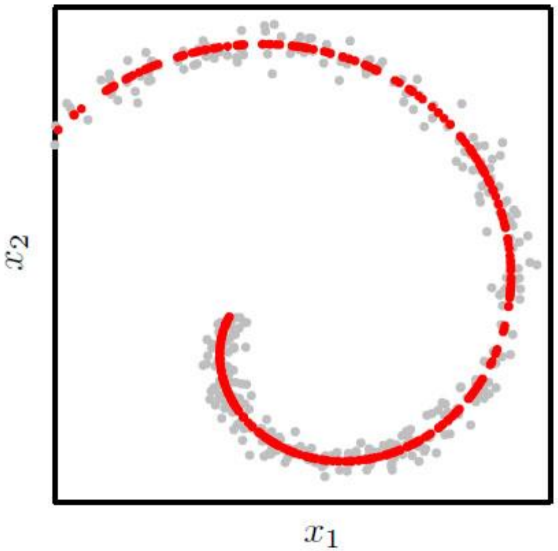


(a) Data in $\mathcal{X}$ space (a) Transformed data in $\mathcal{Z}$ space (b) Reconstruction in $\mathcal{Z}$ space

- Možemo primeniti nelinearnu transformaciju na ulazne podatke:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \xrightarrow{\Phi} \begin{bmatrix} r \\ \theta \end{bmatrix} = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ \tan^{-1}\left(\dfrac{x_2}{x_1}\right) \end{bmatrix}$$

Jedno rešenje: nelinearna transformacija



- Na slici je prikazana rekonstrukcija podataka vraćena u originalni prostor $\mathcal{X}$
- Ovo ne mora uvek biti moguće (Φ mora biti invertibilno)
- Generalno, možemo želeti da primenjujemo Φ koje nije invertibilno, poput polinoma 2. stepena

- Ali, ovo nas ne sprečava da primenjujemo PCA sa nelinearnim transformacijama – naš cilj je predikcija
  1. Transformisaćemo u Z prostor
  2. Primenićemo PCA u Z prostoru
  3. Primenićemo obučavajući algoritam koristeći gornjih K glavnih komponenti u prostoru Z
  4. Prilikom klasifikacije novog primera x, transformisaćemo ga u Z prostor i tamo klasifikovati

PCA sa nelinearnom transformacijom obeležja u obučavanju

Ulaz	• $X \in \mathbb{R}^{N \times D}$ – matrica trening podataka i $y \in \mathbb{R}^N$ vektor ciljnog obeležja • K $(0 < K \leq D)$ • Transformacija Φ
Postupak	1. Transformisati ulazne podatke: $z^{(n)} = \Phi\left(x^{(n)}\right)$ 2. Centrirati podatke: $z^{(n)} \leftarrow z^{(n)} - \bar{z}$, gde je $\bar{z} = \frac{1}{N}\sum_{n=1}^{N} z^{(n)}$ 3. Formirati centriranu matricu Z čiji su redovi $\mathrm{z}^{(n)}$ 4. Primeniti SVD na matricu Z: $[U, \Sigma, V] = \mathrm{svd}(Z)$ 5. Neka su $V_K = [v_1, \dots, v_K]$ prvih K kolona matrice V 6. Matrica sa novim (PCA) obeležjima je: $Z_K = ZV_K$ 7. Koristeći podatke (Z_K, y) naučiti finalnu hipotezu $\tilde{g}$
Izlaz	Konačna hipoteza $g(x) = \tilde{g}\left(V_K^T(\Phi(x) - \bar{z})\right)$

PCA sa nelinearnom transformacijom

- Postoje i drugi pristupi za nelinearnu redukciju dimenzionalnosti PCA metode
- Kernel-PCA: način da se kombinuje PCA sa nelinearnim transformacijama obeležja, bez da moramo posetiti Z prostor (slično kernel triku kod SVM)
- Još neki popularni pristupi: Neural-Network auto-encoder, nonlinear principal curves and surfaces
- Neparametarski pristupi: Laplacian Eigenmap, Locally Linear Embedding (LLE)