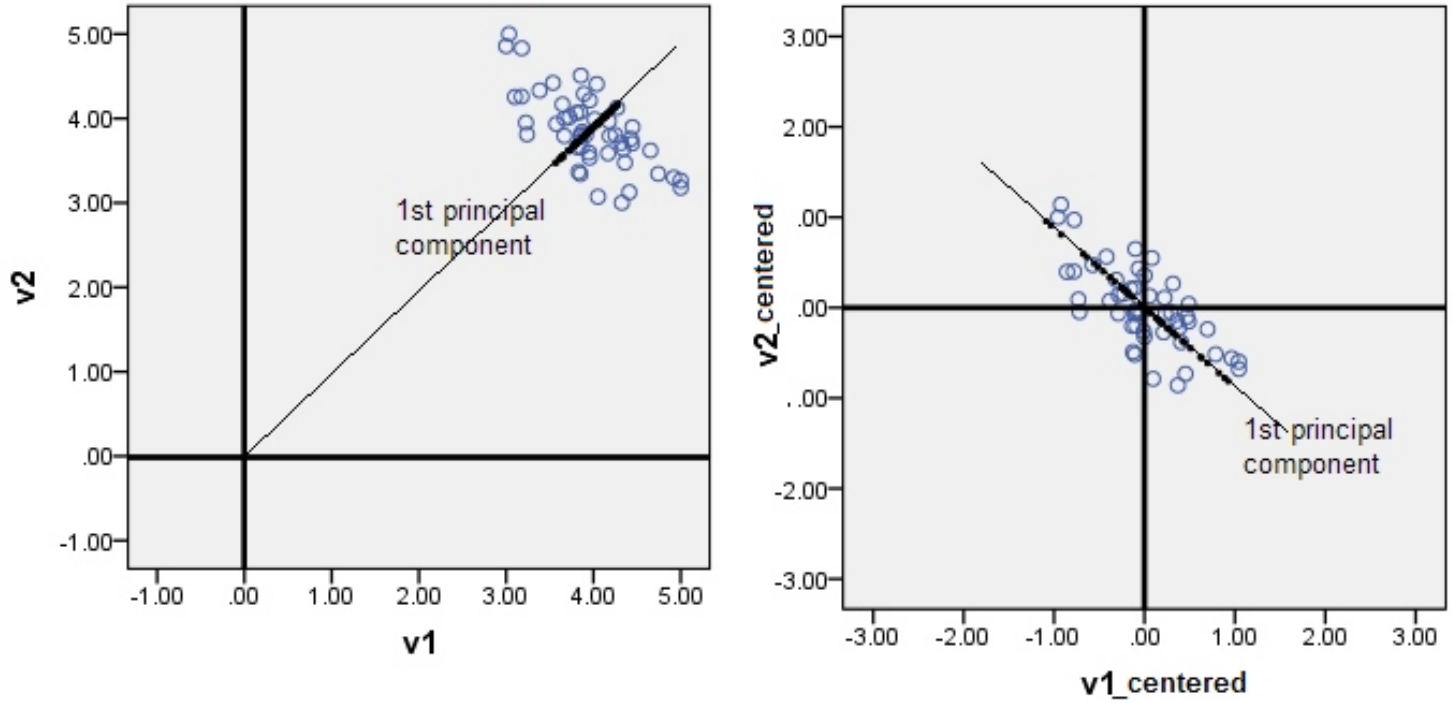
Pretprocesiranje podataka za PCA

1. Podaci moraju biti centrirani
   - Prilikom formiranja Σ, pretpostavili smo da obeležja u datom skupu podataka imaju srednju vrednost 0 ($\Sigma = X^T X$, a empirijska varijansa iz uzorka se računa $\frac{1}{N}\sum_{i=1}^{N}\left(x^{(i)} - \mu\right)^2$, dakle, pretpostavka je $\mu = 0$)



Pretprocesiranje podataka za PCA

- Postupak centriranja podataka (*mean normalization*):
  a. Dat je trening skup $T = \left\{\left(x^{(i)}, y^{(i)}\right), i \in \{1, \ldots, N\}, x^{(i)} \in \mathbb{R}^D\right\}$
  b. Za svako obeležje $d \in \{1, \ldots, D\}$ izračunati srednju vrednost: $\mu_d = \frac{1}{N}\sum_{i=1}^{N} x_d^{(i)}$
  c. Za svako obeležje d: $x_d^{(i)} \leftarrow x_d^{(i)} - \mu_d$

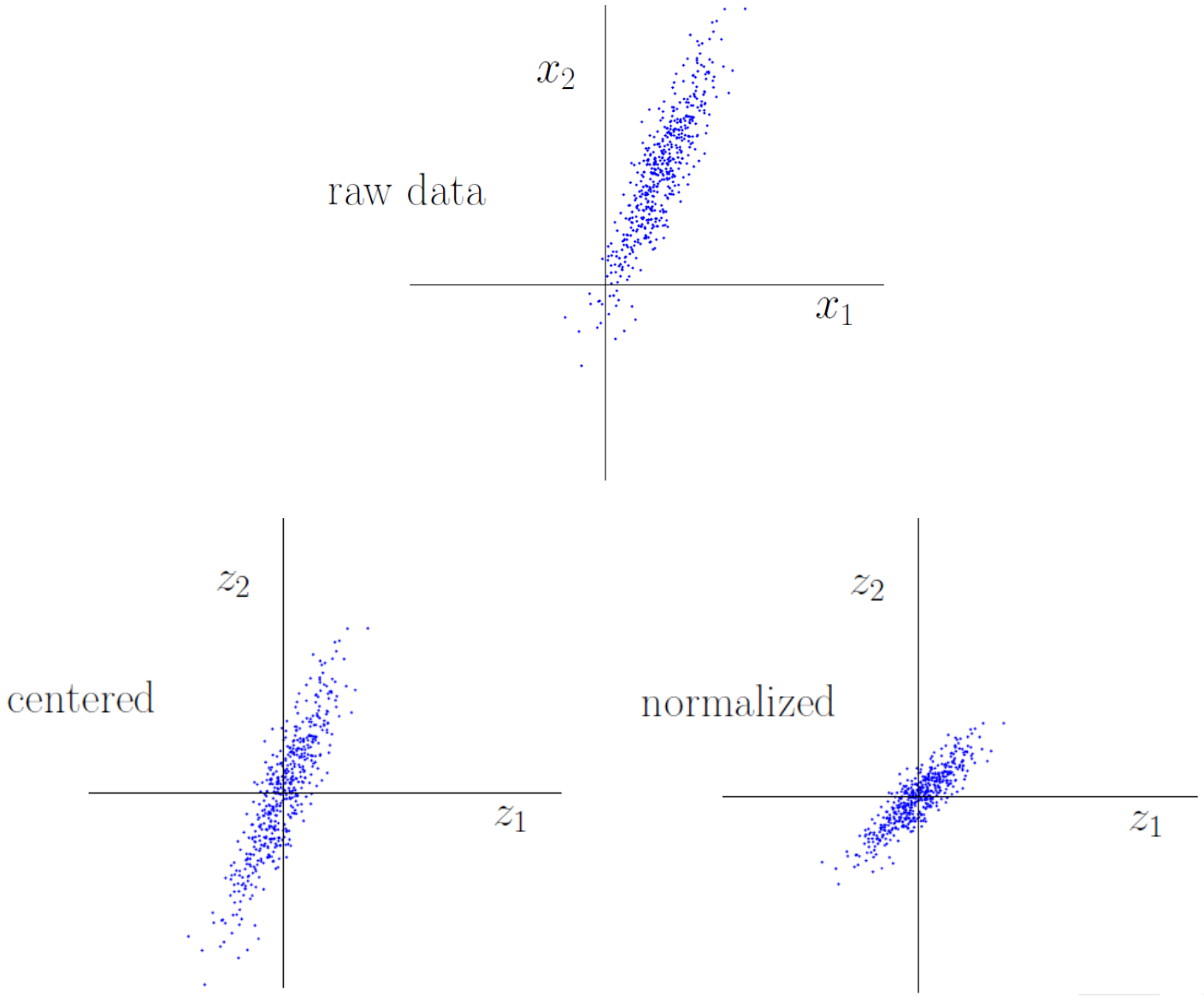
Pretprocesiranje podataka za PCA

2. Normalizovati podatke (opciono)
   - Ako se opsezi vrednosti različitih obeležja veoma razlikuju, obeležja treba skalirati tako da se kreću u približno istom opsegu, npr.

$$x_d^{(i)} \leftarrow \frac{x_d^{(i)}}{\sigma_d}$$

   - Velike razlike u opsezima varijabli koje potiču iz (proizvoljnog) odabira jedinice u kojima ih izražavamo su problem za PCA

Pretprocesiranje podataka za PCA



PCA algoritam

Ulaz	• $X \in \mathbb{R}^{N \times D}$ – matrica trening podataka (u redovima se nalaze N instanci, a u kolonama D obeležja) • K $(0 < K \leq D)$
Postupak	1. Centrirati i (opciono) normalizovati X 2. Primeniti SVD na X: $[U, S, V] = \mathrm{svd}(X)$ 3. Neka su $V_K = [v_1, v_2, \ldots, v_K]$ prvih K kolona matrice V 4. Matrica sa novim (PCA) obeležjima je: $Z = XV_K$, a rekonstrukcija je $\hat{X} = XV_K V_K^T$
Izlaz	$Z \in \mathbb{R}^{N \times K}$

Važne napomene

- Obeležja dobijena pomoću PCA metode su linearne kombinacije originalnih obeležja

- Svi koraci primenjeni u PCA, uključujući korake pretprocesiranja (centriranje i normalizaciju) su nenadgledani (ne zavise od izlaza y)

- Glavne komponente (sortirane u opadajućem redosledu prema odgovarajućim sopstvenim vrednostima) kumulativno objašnjavaju varijansu u podacima

Primena PCA: ubrzavanje treniranja modela

Imamo trening skup $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$

1. Neanotirani skup: $x^{(1)}, x^{(2)}, \ldots, x^{(N)}, x \in \mathbb{R}^D$
2. Primenimo PCA: $z^{(1)}, z^{(2)}, \ldots, z^{(N)}, z \in \mathbb{R}^K, K < D$
3. Treniramo model na novom trening skupu: $T' = \{(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \ldots, (z^{(N)}, y^{(N)})\}$

Važna napomena!

- Mapiranje $x^{(i)} \rightarrow z^{(i)}$ treba da bude realizovano primenom PCA samo nad trening skupom

- Razlog je što moramo primeniti centriranje i normalizaciju podataka. Ako bi test podaci bili uključeni u ovo, to bi bio *data snooping*

- Tek kada pronađemo mapiranje $x^{(i)} \rightarrow z^{(i)}$, identičnu transformaciju možemo primeniti na validacioni i test skup

Kako odabrati broj novih obeležja *K*?

- Ako nam je cilj vizuelizacija, odabraćemo $K = 2$ ili $K = 3$
- Ako nam je cilj da ubrzamo obučavajući algoritam, obično ćemo *K* odabrati tako da zadržimo određeni procenat varijanse (tipično 99%, 95% ili 90%)

Odabir *K* – zadržavanje % varijanse

- Zadržani procenat varijanse u podacima (želimo da bude što bliže 1):

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_K}{\lambda_1 + \lambda_2 + \cdots + \lambda_D}$$

- Procedura:
  1. $[U, S, V] = \text{svd}(X)$
  2. S je dijagonalna matrica: $\begin{bmatrix} \sqrt{\lambda_1} & & & \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ & & & \sqrt{\lambda_D} \end{bmatrix}$
  3. Ako želimo da sačuvamo 99% varijanse u podacima, odabrati *K* tako da:

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_K}{\lambda_1 + \lambda_2 + \cdots + \lambda_D} \geq 0.99$$

Loše upotrebe PCA – prevencija overfittinga

- Ideja: manje obeležja – manje šanse za overfitting
- Ova ideja nije suluda, i može se desiti da dobijemo dobre performanse, ali ovo nije dobar način da adresiramo problem overfitting-a
  - PCA ne uzima u obzir labele y, a odbacuje deo informacija
  - Može se desiti da je deo koji odbacimo veoma važan
- Mnogo bolji način da se adresira overfitting jeste regularizacija
  - Radiće isto ili bolje nego PCA za ovu primenu

Loše upotrebe PCA

- Hipotetički dizajn ML sistema:
  1. Prikupiti trening skup
  2. Primeniti PCA za redukciju dimenzionalnosti
  3. Trenirati model
  4. Testirati na test skupu

- Pre ovog „komplikovanog“ plana – probati primenu algoritma bez PCA na originalnim podacima
- Tek ako to ne uspe (algoritam je previše spor, zahteva previše memorije,...), probati PCA