Assignment 4

Clustering Languages

Verena Blaschke

July 04, 2018

Assignment 4

I: Feature extraction

II: K-means clustering

II: K-means clustering

III: Principal component analysis

IV: Evaluation with gold-standard labels

V: Calculating distances

VI: Hierarchical clustering

I: Feature extraction

fin	s i l m æ
fin	k ɔ r ʋ ɑ
fin	n ɛ n æ
fin	s u u
...	
cmn	t^h ʊ ŋ t ʂ ɨ
cmn	ɻ ə n n a ɪ

I: Feature extraction

fin	s i l m æ
fin	k ɔ r ʋ ɑ
fin	n ɛ n æ
fin	s u u
...	
cmn	tʰ ʊ ŋ t ʂ ɨ
cmn	ɻ ə n n a ɪ

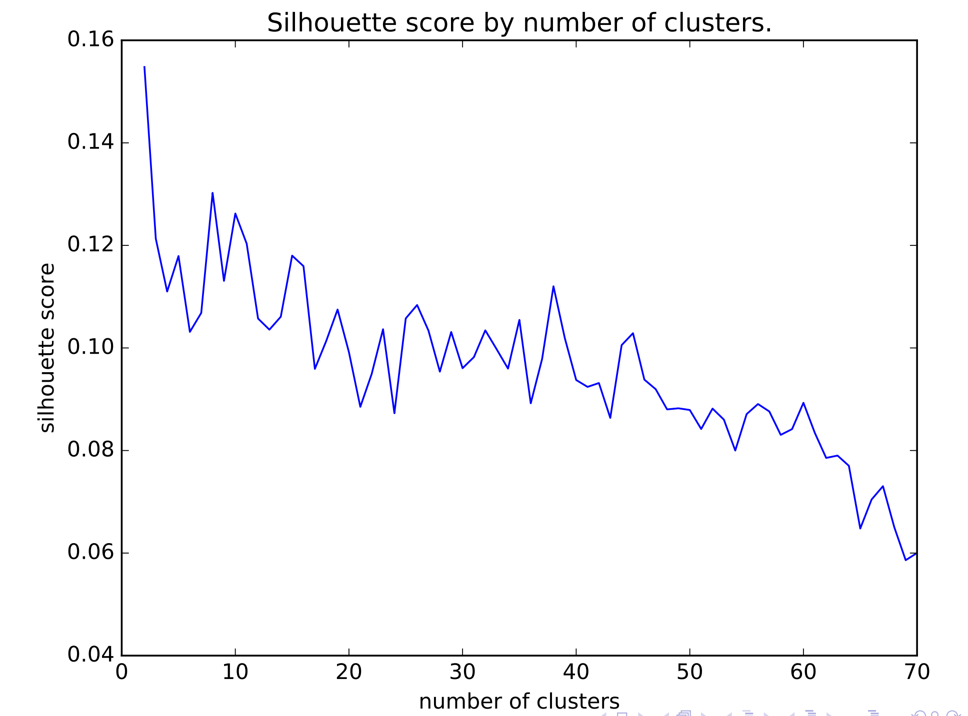
- 80 languages × 272 IPA segments

II: K-means clustering

- k-means clustering for k in $[2, 70]$
- silhouette coefficient

II: K-means clustering

- k-means clustering for k in $[2, 70]$
- silhouette coefficient
  - How close (=similar) is each data point to other points from its own cluster compared to other clusters?
  - $[-1, +1]$, higher scores are better

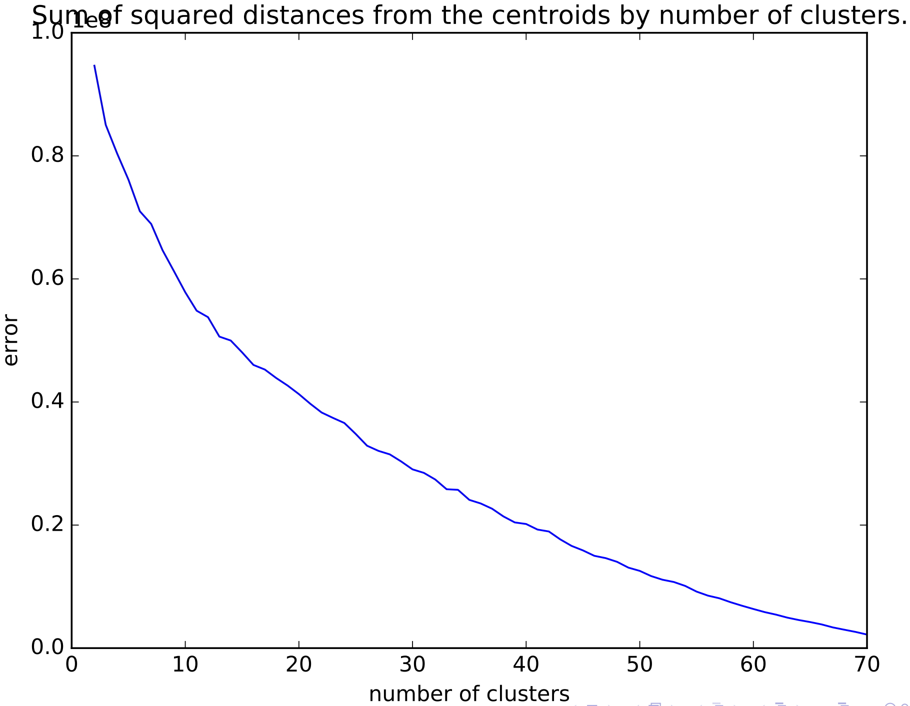

Silhouette score by number of clusters.
silhouette score
number of clusters
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0
10
20
30
40
50
60
70

II: K-means clustering

- k-means clustering for k in $[2, 70]$
- silhouette coefficient
  - How close (=similar) is each data point to other points from its own cluster compared to other clusters?
  - $[-1, +1]$, higher scores are better
- error function
  - sum of squared distances from the closest centroids `kmeans.inertia_`

II: K-means clustering

- k-means clustering for k in $[2, 70]$
- silhouette coefficient
  - How close (=similar) is each data point to other points from its own cluster compared to other clusters?
  - $[-1, +1]$, higher scores are better
- error function
  - sum of squared distances from the closest centroids `kmeans.inertia_`
  - What is a good number of clusters? → elbow method



Sum of squared distances from the centroids by number of clusters.
1e8
1.0
0.8
0.6
0.4
0.2
0.0
error
0
10
20
30
40
50
60
70
number of clusters

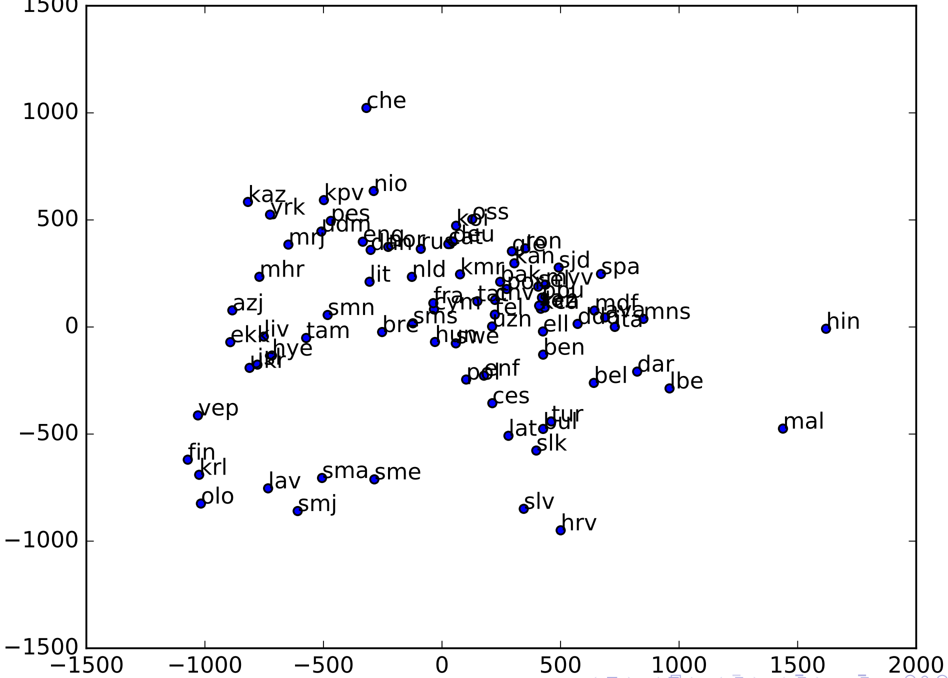
III: Principal component analysis

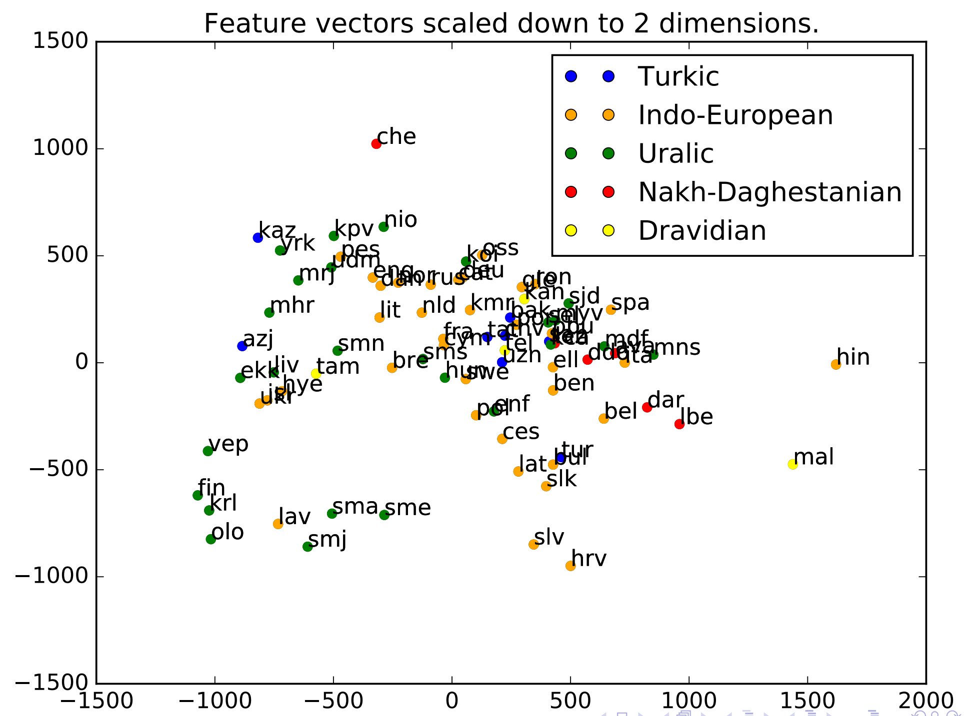
- remove redundant features

III: Principal component analysis

- remove redundant features
  - remove noise
  - train machine learning models more quickly

Feature vectors scaled down to 2 dimensions.




Feature vectors scaled down to 2 dimensions.
Turkic
Indo-European
Uralic
Nakh-Daghestanian
Dravidian
che
kaz
yrk
kpv
nio
pes
udm
mrj
eng
dan
rus
oss
koi
mhr
lit
nld
kmr
sjd
spa
azj
smn
tam
bre
sms
hun
swe
ekk
liv
hye
ukr
ell
ben
hin
mdf
mns
dar
lbe
bel
pol
ces
vep
tur
bul
lat
slk
mal
fin
krl
sma
sme
lav
olo
smj
slv
hrv
−1500
−1000
−500
0
500
1000
1500
2000

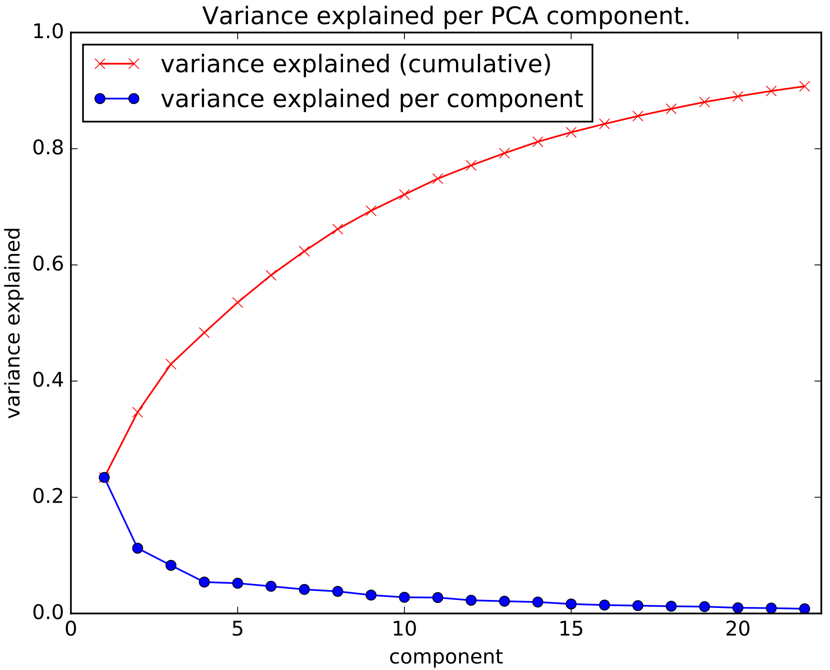
III: Principal component analysis

```
pca = PCA(features.shape[1])
d = 0
var_explained = 0
while var_explained < 0.9:
    var_explained += pca.explained_variance_ratio_[d]
    d += 1


featuresPCA = PCA(d).fit_transform(features)
```

III: Principal component analysis

```
pca = PCA(features.shape[1])
d = 0
var_explained = 0
while var_explained < 0.9:
    var_explained += pca.explained_variance_ratio_[d]
    d += 1


featuresPCA = PCA(d).fit_transform(features)
```

```
pca = PCA(0.9)
print(pca.n_components_)
```



Variance explained per PCA component.
variance explained (cumulative)
variance explained per component
variance explained
component
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20

IV: Evaluation with gold-standard labels

```
n_fam = len(set(family))
pred_all = KMeans(n_fam).fit_predict(features)
pred_pca = KMeans(n_fam).fit_predict(featuresPCA)
```

IV: Evaluation with gold-standard labels

```
n_fam = len(set(family))
pred_all = KMeans(n_fam).fit_predict(features)
pred_pca = KMeans(n_fam).fit_predict(featuresPCA)
```

```
lang    all    pca    family
--------------------------------------
kan     2      4      Dravidian
tam     3      0      Dravidian
tel     4      0      Dravidian
mal     4      2      Dravidian
bul     0      1      Indo-European
ces     0      1      Indo-European
...
```

IV: Evaluation with gold-standard labels

- **Homogeneity**: Each cluster contains data points of the same gold-standard class.

IV: Evaluation with gold-standard labels

- **Homogeneity**: Each cluster contains data points of the same gold-standard class.
- **Completeness**: All members of a gold-standard class are in the same cluster.

IV: Evaluation with gold-standard labels

- **Homogeneity**: Each cluster contains data points of the same gold-standard class.
- **Completeness**: All members of a gold-standard class are in the same cluster.
- **V-measure**: Harmonic mean of homogeneity and completeness.
- $[0, 1]$ higher is better

IV: Evaluation with gold-standard labels

- **Homogeneity**: Each cluster contains data points of the same gold-standard class.
- **Completeness**: All members of a gold-standard class are in the same cluster.
- **V-measure**: Harmonic mean of homogeneity and completeness.
- $[0,1]$ higher is better

```
all   H: 0.1707   C: 0.1461   V: 0.1575
PCA   H: 0.1728   C: 0.1572   V: 0.1646
```

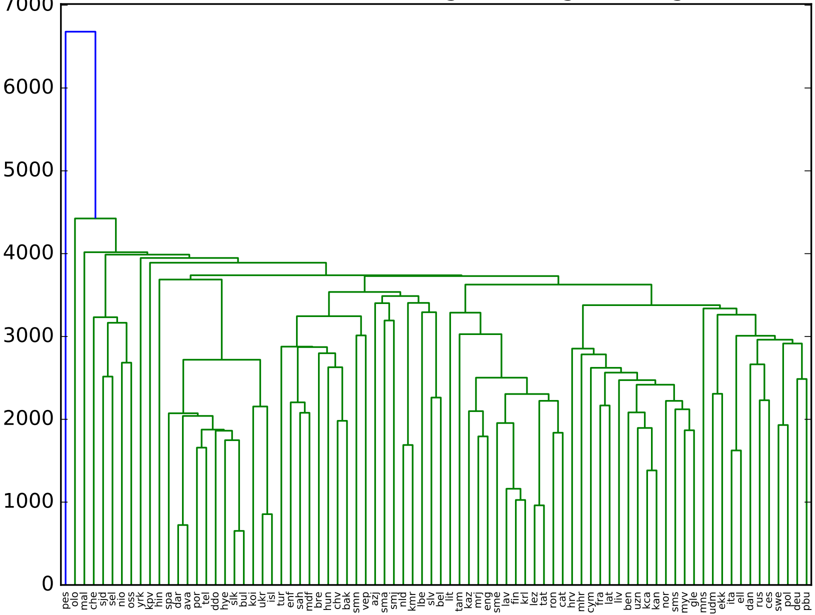
V: Calculating distances

	A	B	C	D	E
A		123	452	10	572
B			342	370	908
C				127	754
D					23
E					

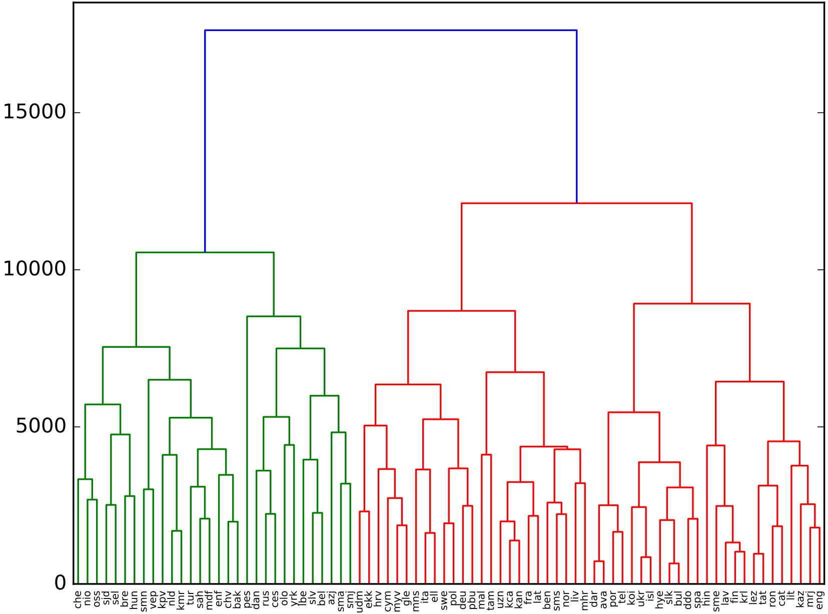
VI: Hierarchical clustering

```
for m in ['single', 'complete', 'average']:
    fig, ax = plt.subplots()
    z = scipy.cluster.hierarchy.linkage(dist, method=m)
    scipy.cluster.hierarchy.dendrogram(z, labels=languages)
    fig.savefig('dendrogram-{}.pdf'.format(method))
```

Hierarchical clustering with single linking



Hierarchical clustering with complete linking



Hierarchical clustering with average linking