Stack Overflow User Classification

by: Accever Mendoza

Executive Summary

Stack Overflow (https://stackoverflow.com (https://stackoverflow.com)) is a platform where developers learn and also share their programming knowledge. From the data of Stack Overflow stored on jojie (supercomputer), we look into the posts and comments in the website. The goal is to classify the users based on where they post and the tags used.

We parse the data using lxml tree parser to get the data on tags used and the user comments for the posts. Given that Stack Overflow has about 80 gigabytes of data, random sampling was done.

We able to cluster by using Mini-Batch k-means which is alternative to the k-means algorithm from scikit-learn. Using internal validation criteria, we are able to pinpoint optimal k at k = 4. The clusters were formed generally by the different programming languages.

Parsing the Data

In [1]:

```
from lxml import etree
comms = '/mnt/data/public/stackoverflow/Comments.xml'
posts = '/mnt/data/public/stackoverflow/Posts.xml'
```

In [3]:

```
temp = etree.iterparse(posts)
l = next(temp)
l[1].attrib
```

Out[3]:

```
{'Id': '4', 'PostTypeId': '1', 'AcceptedAnswerId': '7', 'CreationDat
e': '2008-07-31T21:42:52.667', 'Score': '543', 'ViewCount': '34799',
'Body': "<p>I want to use a track-bar to change a form's opacity.</p
>\n\n<p>This is my code:</p>\n\n<pre><code>decimal trans = trackBar1
.Value / 5000;\nthis.Opacity = trans;\n</code></pre>\n\n<p>When I bu
ild the application, it gives the following error:</p>\n\n<blockquot
e>\n  <p>Cannot implicitly convert type <code>'decimal'</code> to <c
ode>'double'</code>.</p>\n</blockquote>\n\n<p>I tried using <code>tr
ans</code> and <code>double</code> but then the control doesn't work
. This code worked fine in a past VB.NET project.</p>\n", 'OwnerUser
Id': '8', 'LastEditorUserId': '3151675', 'LastEditorDisplayName': 'R
ich B', 'LastEditDate': '2017-09-27T05:52:59.927', 'LastActivityDate
': '2018-02-22T16:40:13.577', 'Title': 'While applying opacity to a
form, should we use a decimal or a double value?', 'Tags': '<c#><win
forms><type-conversion><decimal><opacity>', 'AnswerCount': '13', 'Co
mmentCount': '1', 'FavoriteCount': '39', 'CommunityOwnedDate': '2012
-10-31T16:42:47.213'}
```

In [4]:

```
i = 0
N_iter = 500000
comm_postID = []
comm_userID = []

post_ID = []
post_tags = []

for ((event1, element1), (event2, element2)) in zip(etree.iterparse(comms),
                                                    etree.iterparse(posts)):

    if {'PostId', 'UserId'}.issubset(set(element1.attrib.keys())):
        comm_postID.append(element1.attrib['PostId'])
        comm_userID.append(element1.attrib['UserId'])

    if {'Tags', 'Id', }.issubset(set(element2.attrib.keys())):
        post_ID.append(element2.attrib['Id'])
        post_tags.append(element2.attrib['Tags'])

    i += 1
    if i >= N_iter:
        break
```

In [5]:

```
import pandas as pd
import numpy as np

df_comm = pd.DataFrame({'comm_postID' : comm_postID,
              'comm_userID' : comm_userID})

df_post = pd.DataFrame({'post_ID' : post_ID,
              'post_tags' : post_tags})
```

In [6]:

```
# Write parsing results to csv file for retrieval

df_post.to_csv('posts.csv')
df_comm.to_csv('comm.csv')
```

In [30]:

```
# Reload the csv files for clustering

import pandas as pd
import numpy as np

df_post = pd.read_csv('posts.csv')
df_comm = pd.read_csv('comm.csv')
```

In [31]:

```
display(df_post.head())
display(df_comm.head())
```

	Unnamed: 0	post_ID	post_tags
0	0	4	<c#><winforms><type-conversion><decimal><opacity>
1	1	6	<html><css><css3><internet-explorer-7>
2	2	9	<c#><.net><datetime>
3	3	11	<c#><datetime><time><datediff><relative-time-s...
4	4	13	<javascript><html><browser><timezone><timezone...

	Unnamed: 0	comm_postID	comm_userID
0	0	35314	1
1	1	35314	3
2	2	35195	380
3	3	47239	4550
4	4	45651	242

In [32]:

```
# Drop irrelevant columns and reemove separator for the tags

df_post.drop(columns=['Unnamed: 0'], inplace=True)
df_comm.drop(columns=['Unnamed: 0'], inplace=True)

lst = df_post['post_tags'].tolist()
df_post['post_tags'] = [s.replace('><', ' ').strip('<').strip('>') for s in lst]
```

In [33]:

```
# Join the post and comments dataframe with the same Post_ID

df_comm.columns = ['post_ID', 'user_ID']
df_post_comm = pd.merge(df_post, df_comm, on='post_ID')
```

In [34]:

```
# Create the dataframe for users ang tags used on the posts

df_user = pd.DataFrame(columns=['userID', 'post_tags'])

df_user['userID'] = df_post_comm.user_ID.unique()
```

In [35]:

```
# Determine sample size needed for Confidence Interval=95% and Error Margin of 3
%

from scipy import stats

CI =0.95 # confidence interval
ERR=0.03 # error margin

Z=stats.norm.ppf(1-((1-CI)/2))
n=Z**2/(4*(ERR**2))

print("sample n needed= {}".format(n))
```

```
sample n needed= 1067.0718946372572
```

In [36]:

```
# Perform Random sampling for the user IDs
df_user = df_user.sample(1100, random_state=7) #sample size ~1100
```

In [37]:

```
df_user.head()
```

Out[37]:

	userID	post_tags
2926	13342	NaN
1435	21364	NaN
7733	72686	NaN
2338	22970	NaN
305	5197	NaN

In [38]:

```
def get_post_tags(userID):
    '''
    Parameter
    ---------
    userID: int
        the user ID

    Returns
    -------
    s: string
        all post_tags for all posts commented by the given user
    '''
    s = ' '.join(df_post_comm.loc[df_post_comm['user_ID']==userID]['post_tags'].
tolist())
    return(s)

get_post_tags(3)
```

Out[38]:

```
'asp.net-mvc throttling asp.net asp.net-mvc performance webforms vim
emacs viper vimpulse c# design events asp.net download connection we
bclient asp.net-mvc html-helper renderpartial asp.net windows-mobile
mobile viewstate database memory footprint winapi powershell volume
mute asp.net regex http http-headers parsing compiler-construction c
# msbuild msbuild-task invalidoperationexception c# include director
y copy java jvm stack heap java database soap sybase reliability c#
.net math matrix mathdotnet jquery ajax json firefox rest linq-to-sq
l ruby-on-rails ruby helper view-helpers c# .net algorithm cocoa mac
os isight visual-studio-2008 crash add-in .net linq data-access-laye
r jquery asp.net-mvc c# asp.net datagrid css internet-explorer firef
ox font-size c# .net oracle stored-procedures plsql'
```

In [39]:

```
# Apply the new function for each user in the user dataframe

df_user['post_tags'] = df_user.apply(lambda row: get_post_tags(row.userID), axis
=1)
```

In [40]:

```
df_user.head()
```

Out[40]:

	userID	post_tags
2926	13342	java jsp import jstl java math optimization pe...
1435	21364	java reflection classloader
7733	72686	c# winforms drag-and-drop inno-setup
2338	22970	jquery cross-browser sharepoint logging moss l...
305	5197	sql-server xml xquery html internet-explorer f...

Vectorize the post_tags in the df_user table into a bag-of-words (bows)

We use **sklearn's TFIDFVectorizer class**. Here, we use the parameter: *max_features* = 2000 i.e., the number of words that will be used for each post_tags entry. It is then stored in our bows (bag-of-words) variable which is a representation of our vector matrix.

In [41]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=2000)
bows = vectorizer.fit_transform(df_user['post_tags'])

bows = pd.DataFrame(bows.todense())
bows.head()
```

Out[41]:

	0	1	2	3	4	5	6	7	8	9	...	1990	1991	1992	1993	1994	1995	19
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 2000 columns

In [42]:

```
bows.shape
```

Out[42]:

```
(1100, 2000)
```

Use PCA to Visualize in 2D and 3D

We need to visualize our data in 2D or 3D. In order to do this, we use **PCA (Principal Component Analysis)** to project our data in such a way that we get the maximum variance explained.

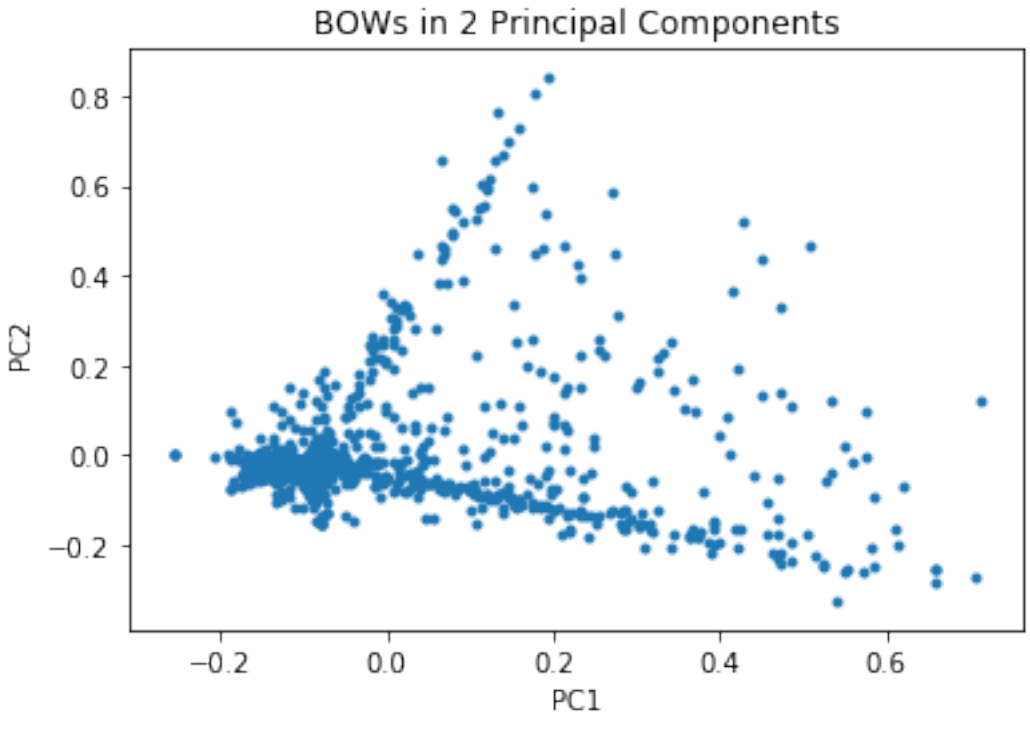
In [43]:

```
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

pca = PCA()
sk_pca = pca.fit_transform(bows)
sk_res = pd.DataFrame(sk_pca)
```

Visualize in 2D

In [44]:

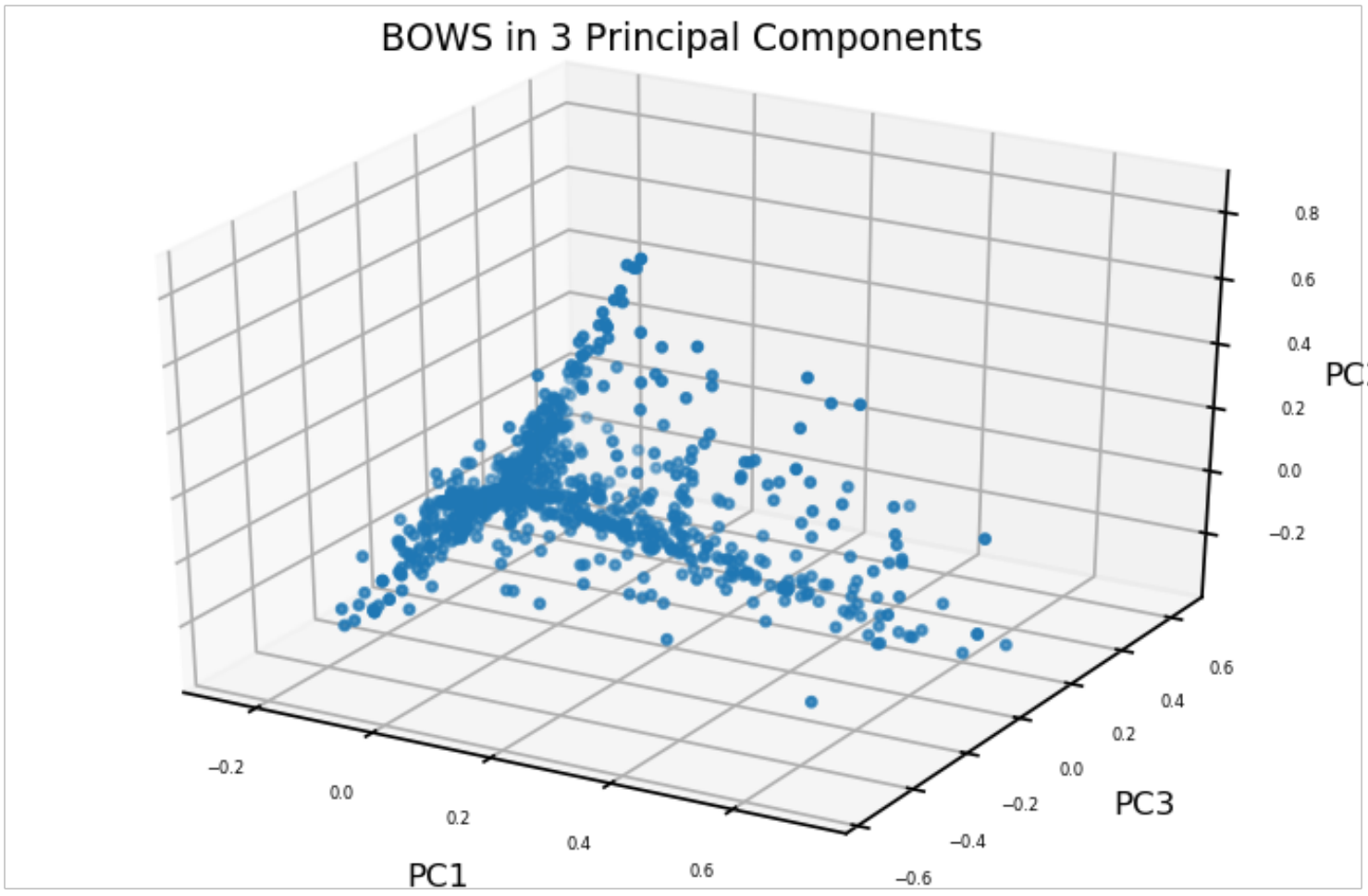
```
# fig, ax = plt.subplots(subplot_kw=dict(aspect='equal'))
fig, ax = plt.subplots()
ax.scatter(sk_res[0], sk_res[1], marker='.');
ax.set_title("BOWs in 2 Principal Components")
ax.set_xlabel('PC1')
ax.set_ylabel('PC2');
```



Visualize in 3D

In [45]:

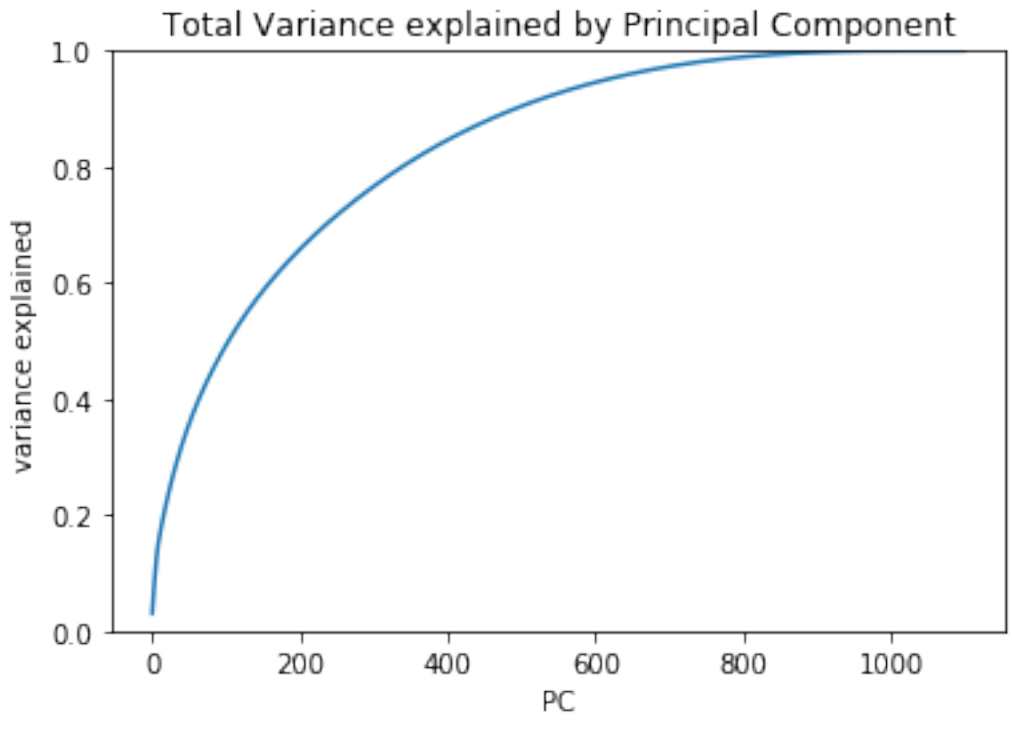
```
from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

fig = plt.figure(dpi=150)
ax = fig.add_subplot(111, projection='3d')
ax.scatter(sk_res[0], sk_res[2], sk_res[1], marker='.')
ax.tick_params(axis='both', which='major', labelsize=4)
ax.set_title("BOWS in 3 Principal Components", fontsize=9)
ax.set_xlabel('PC1', fontsize=8)
ax.set_ylabel('PC3', fontsize=8)
ax.set_zlabel('PC2', fontsize=8, labelpad=-1);
```



We use a scree plot to verify the **variance explained** and to be able to reduce dimensionality.

In [46]:

```
#cumulative variance explained
plt.plot(pca.explained_variance_ratio_.cumsum())
plt.ylim(0,1)
plt.title("Total Variance explained by Principal Component")
plt.xlabel('PC')
plt.ylabel('variance explained');
```



In [47]:

```
sk_res = sk_res.iloc[:, 0:-300]
```

Using K-Means Clustering

Clustering partitions data points into groups of similar points. Representantive-based clustering methods is a family of clustering methods wherein a representative point, is assigned for each cluster. Data points are then assigned to the nearest representative after which a new representative is selected for the cluster. The alternating steps of assigning points to a cluster and selecting a representative is repeated until it converges. This approach is similar to the prototype method in k-NN classification.

The most common representative-based clustering method is k-means clustering. In this method, the distance function is the L2-norm or the squared sum error (SSE), which is just the square of the L2 norm. The representative is chosen to be the mean of the points in the cluster. We will be using Sklearn's **MiniBatchKMeans** is an alternative to K-means but is faster.

In [48]:

```
from sklearn.cluster import KMeans, MiniBatchKMeans
from scipy.spatial.distance import euclidean
from sklearn.metrics import silhouette_score
```

There are two types of validation method to determine goodness of the cluster, internal and external validation criteria. External validation would be needing a ground truth.

For internal, there are 3 general **Cluster Validation Criteria** that we will use in order to determine the quality of the clustering formed.

- **Sum of Squares Distances to centroids**: This is also referred to later as **SSE**. Not to sure but I think this stands for (Sum of Squares using Euclidean distances). This metric measures the distances of data points from the cluster centroids. **The smaller the distances suggest that the data points are more clumped. The smaller the value to this, the better.**
- **Intracluster to intercluster distance ratio**: This refers to the ratio between the average distance of 2 data points within the same cluster versus the average distance between 2 data points belonging to another cluster. Similar to SSE, **the smaller the value to this, the better.**
- **Silhouette coefficient**: can be interpretted as **how separate the clusters are to each other**. **Values range from -1 to 1 where 1 means there is good separation between clusters and -1 indicates some level of "mixing" or "overlapping" between clusters.**

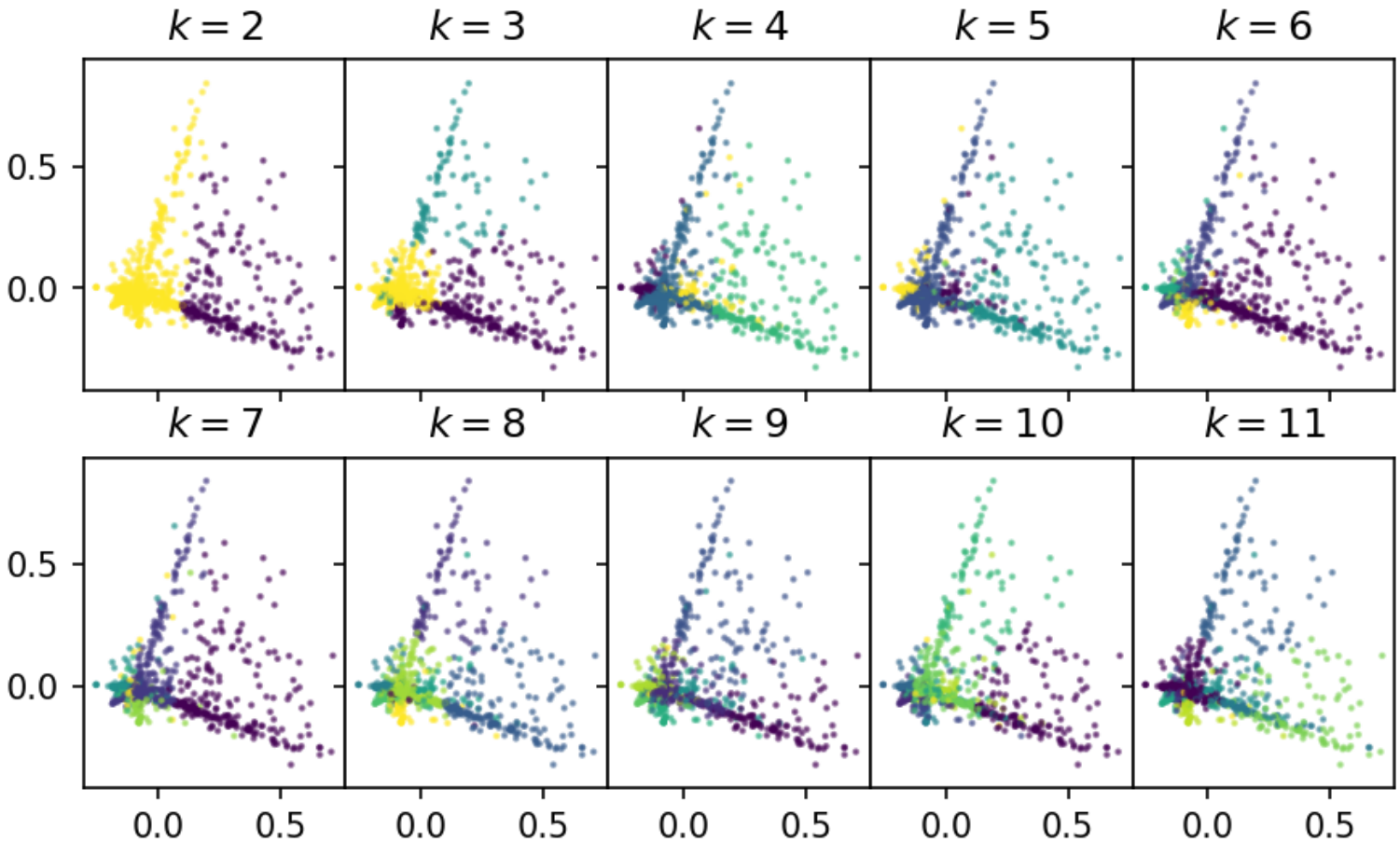
In [49]:

```
def intra_to_inter(X, y, dist, r):
    """Compute intracluster to intercluster distance ratio

    Parameters
    ----------
    X : array
        Data matrix with each row corresponding to a point
    y : array
        Class label of each point
    dist : callable
        Distance between two points. It should accept two arrays, each
        corresponding to the coordinates of each point
    r : integer
        Number of pairs to sample

    Returns
    -------
    ratio : float
        Intracluster to intercluster distance ratio
    """
    dist_P = []
    dist_Q = []
    np.random.seed(11)
    for i, j in np.random.randint(low=0, high=len(y), size=[r,2]):
        # just skip the pair even if we end up having pairs less than r
        if i == j:
            continue
        # intracluster
        elif y[i] == y[j]:
            dist_P.append(dist(X[i], X[j]))
        # intercluster
        else:
            dist_Q.append(dist(X[i], X[j]))
    intra = np.sum(dist_P) / len(dist_P)
    inter = np.sum(dist_Q) / len(dist_Q)
    ratio = intra / inter
    return ratio
```

Clustering

In [50]:

```
fig, ax = plt.subplots(2, 5, dpi=150, sharex=True, sharey=True, figsize=(7,4),
                       subplot_kw=dict(aspect='equal'),
                       gridspec_kw=dict(wspace=0.0))
inertias = []
iidrs = []
scs = []

for i in range(2, 12):
#     kmeans = KMeans(n_clusters=i, random_state=1337)
    kmeans = MiniBatchKMeans (n_clusters=i, random_state=1337)
    y = kmeans.fit_predict(sk_res)

    inertias.append(kmeans.inertia_)
    iidrs.append(intra_to_inter(sk_res.values, y, euclidean, 50))
    scs.append(silhouette_score(sk_res, y))

    if i < 7:
        ax[0][i%7-2].scatter(sk_res[0], sk_res[1], s=5, c=y, alpha=0.5, marker='
.')
        ax[0][i%7-2].set_title('$k=%d$'%i)
    else:
        ax[1][i%7].scatter(sk_res[0], sk_res[1], s=5, c=y, alpha=0.5, marker='.'
)
        ax[1][i%7].set_title('$k=%d$'%i)
```

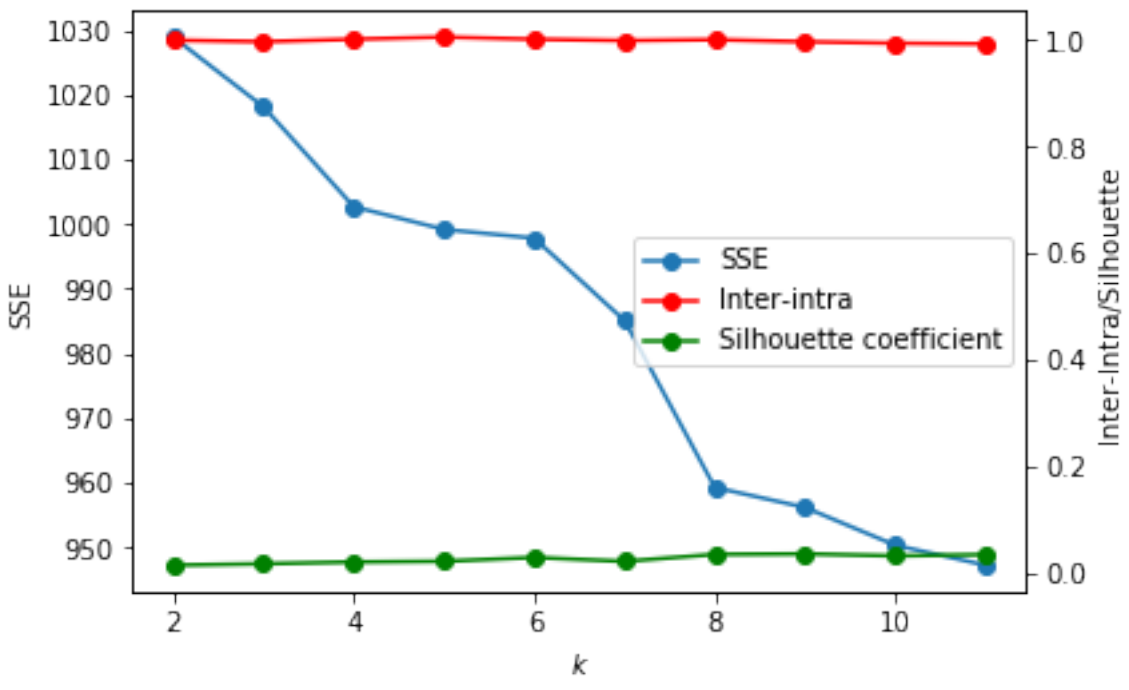


Interpreting the Results

As mentioned, we now use the **Internal Validation Criteria**. We plot these criteria for each k we did above.

In [51]:

```
plt.plot(np.arange(2,12), inertias, '-o', label='SSE')
plt.xlabel('$k$')
plt.ylabel('SSE')
lines, labels = plt.gca().get_legend_handles_labels()
plt.twinx()
plt.plot(np.arange(2,12), iidrs, '-ro', label='Inter-intra')
plt.plot(np.arange(2,12), scs, '-go', label='Silhouette coefficient')


plt.ylabel('Inter-Intra/Silhouette')
lines2, labels2 = plt.gca().get_legend_handles_labels()
plt.legend(lines+lines2, labels+labels2);
```



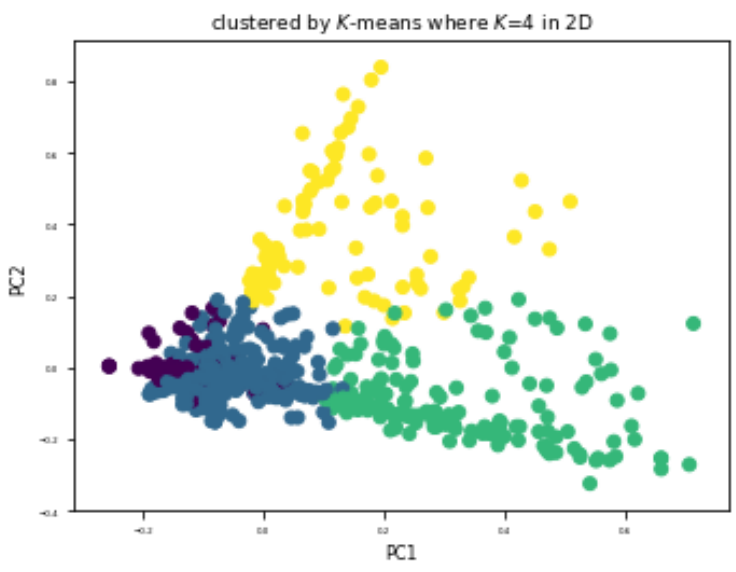
Best k value

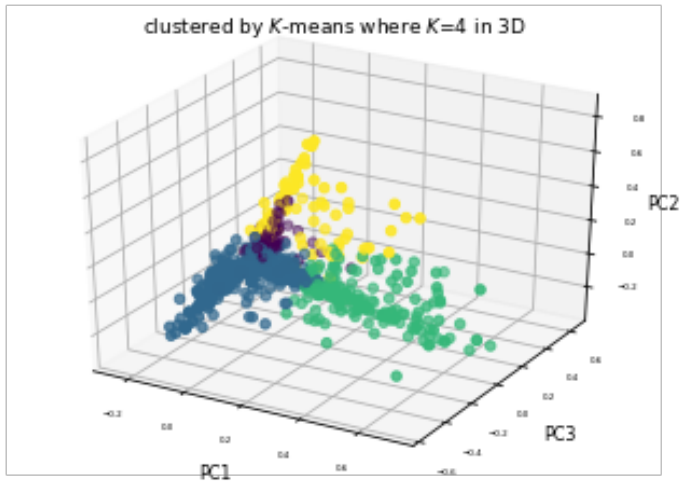
The silhouette coefficient and intercluster to intracluster ratio are fairly constant althroughout. Candidates for k values are 4 and 8, elbow was created in the plot of the mentioned k. But for this purpose, we choose k = 4 and we could see that the clusters formed below make sense.

Replotting our clustered data-set according to k value decided:

In [63]:

```
kmeans = KMeans(n_clusters=4, random_state=1337)
y_predict = kmeans.fit_predict(sk_res)

fig = plt.figure(figsize=(12,4))
fig.tight_layout()

ax1 = fig.add_subplot(121)
ax1.scatter(sk_res[0], sk_res[1], c=y_predict)
ax1.tick_params(axis='both', which='major', labelsize=4)
ax1.set_title("clustered by $K$-means where $K$=4 in 2D", fontsize=9)
ax1.set_xlabel('PC1', fontsize=8)
ax1.set_ylabel('PC2', fontsize=8)

ax2 = fig.add_subplot(122, projection='3d')
ax2.scatter(sk_res[0], sk_res[2], sk_res[1], c=y_predict)
ax2.tick_params(axis='both', which='major', labelsize=4)
ax2.set_title("clustered by $K$-means where $K$=4 in 3D", fontsize=9)
ax2.set_xlabel('PC1', fontsize=8)
ax2.set_ylabel('PC3', fontsize=8)
ax2.set_zlabel('PC2', fontsize=8, labelpad=-1);
```





We now have a look at the clusters formed.

In [64]:

```
df_user['Cluster'] = y_predict
df_user.head()
```

Out[64]:

	userID	post_tags	Cluster
2926	13342	java jsp import jstl java math optimization pe...	0
1435	21364	java reflection classloader	0
7733	72686	c# winforms drag-and-drop inno-setup	1
2338	22970	jquery cross-browser sharepoint logging moss l...	1
305	5197	sql-server xml xquery html internet-explorer f...	3

We create separate dataframes for each of the clusters for convenience.

In [65]:

```
df_cluster0 = df_user.loc[df_user['Cluster'] == 0]
df_cluster1 = df_user.loc[df_user['Cluster'] == 1]
df_cluster2 = df_user.loc[df_user['Cluster'] == 2]
df_cluster3 = df_user.loc[df_user['Cluster'] == 3]
```

In [66]:

```
df_cluster0.head(20)

##java
```

Out[66]:

	userID	post_tags	Cluster
2926	13342	java jsp import jstl java math optimization pe...	0
1435	21364	java reflection classloader	0
3649	1448983	java castor java svn continuous-integration cr...	0
3894	41362	java excel apache-poi	0
3925	38696	java data-structures map hashmap treemap	0
491	17205	javascript php json extjs java interface stati...	0
2756	706	audio java-me mobile	0
2610	14326	development-environment vmware virtualization ...	0
556	4945	c# java oop global-variables java autocomplete	0
2383	8217	java unit-testing testing junit	0
2531	939	java dependencies dependency-management java s...	0
3950	23051	java arrays io	0
3777	4308	java file file-io filesystems python regex	0
8235	71522	geocoding java unit-testing testing junit	0
488	18796	java xml soap namespaces axis java jsf tomahaw...	0
8734	81193	mysql foreign-keys innodb self-join phpmyadmin...	0
896	17398	c++ c cross-platform gui-toolkit emacs lisp sc...	0
5694	12652	java c# c++ constructor	0
5220	1440720	java swing baseline java swing baseline java s...	0
2242	2819	hibernate algorithm traversal subtree java lis...	0

In [67]:

```
df_cluster1.head(20)

## language agnostic
```

Out[67]:

	userID	post_tags	Cluster
7733	72686	c# winforms drag-and-drop inno-setup	1
2338	22970	jquery cross-browser sharepoint logging moss l...	1
796	3798	c# .net process ipc c# .net wpf xaml animation...	1
3659	32232	java xsd schema xmlbeans multiple-schema	1
7337	66798	python	1
6355	36805	javascript html css textarea php session sessi...	1
3207	25965	c++ primes sieve c++ primes sieve networking p...	1
8833	46428	c++ c wrapping ambiguity	1
3732	6247	c#	1
3700	7280	.net wcf security wcf-binding wcf configuratio...	1
3511	24079	javascript html	1
4032	39575	xaml silverlight-2.0	1
2397	8945	c stack c integer-overflow size-t c boost c# r...	1
7586	25020	winforms user-interface controls excel pivot c...	1
663	6068	sql sqlite clojure clojure lisp common-lisp py...	1
112	12983	language-agnostic oop java java database servl...	1
2282	6062	mysql database architecture computer-science h...	1
4421	26394	javascript php drupal drupal-views drupal-5	1
2788	25007	c++ algorithm cstring	1
6689	49318	html css	1

In [68]:

```
df_cluster2.head(20)

## .net
```

Out[68]:

	userID	post_tags	Cluster
1228	2947	c# .net operators asp.net sql sql-server secur...	2
7234	68300	asp.net-mvc date-formatting	2
7277	68679	vb.net .net vb.net dynamic-controls vb.net	2
4908	46072	c# asp.net gridview sql dataset sqldatareader ...	2
1113	634	c# .net ado.net timeout	2
3505	15233	c# vb.net casting .net xml xmldocument	2
8500	46769	c# javascript asp.net-2.0	2
4352	29411	.net visual-studio visual-studio-2008 windows-...	2
3759	27907	md5 c# .net asp.net vb.net xml-rpc asp.net sql...	2
8669	7720	.net winforms image-gallery .net-1.1	2
3473	23528	c++ oop visual-c++ templates c++ windows porta...	2
8079	72660	c# calendar c# calendar c# calendar c# gridvie...	2
6759	49294	c# oracle stored-procedures plsql vb.net .net-...	2
2836	29493	c# refactoring lambda .net asp.net-mvc modelbi...	2
8880	82380	c# vb.net ms-access	2
2284	35286	c# prolog integrate asp.net generics user-cont...	2
8006	77040	.net asp.net deployment archive c# asp.net coo...	2
1900	414107	asp.net html visual-studio designer photoshop	2
3856	2572	visual-studio visual-sourcesafe asp.net vb.net...	2
8008	70454	asp.net event-handling	2

In [69]:

```
df_cluster3.head(20)

## sql
```

Out[69]:

	userID	post_tags	Cluster
305	5197	sql-server xml xquery html internet-explorer f...	3
6943	65782	sql vim omnicomplete	3
232	4550	c# code-generation data-access sql sql-server ...	3
1440	11830	design design-patterns poeaa design-patterns c...	3
1083	16881	.net vb.net string c# .net algorithm sorting r...	3
1944	20703	c# dictionary locking c# dictionary locking xs...	3
2346	8454	php optimization code-generation translation s...	3
6482	9354	sql sql-server database database-design data-m...	3
805	740	sql sql-server sql-server-2005 stored-procedur...	3
6113	55020	sql-server tsql transactions	3
1741	12908	c# asp.net vb.net url-rewriting iis-6 sql-serv...	3
3647	2486	javascript jquery internet-explorer tooltip ja...	3
5509	4403	asp.net sql-server database performance design...	3
3530	28053	sql sql-server asp-classic timeout sql-server-...	3
7294	67722	sql html database jsp	3
2969	1534	sql c# wpf data-binding inotifypropertychanged	3
5774	49742	javascript jquery dom sql sql-server linq	3
5335	48478	sql-server c# email smtp java php string sql-s...	3
5242	80954	sql-server visual-studio database-edition	3
420	3201	sql sql-server sql-server-2005 .net-3.5 .net w...	3

Conclusion

After parsing the data, performing the clustering techniques we came to the conclusion of having the following classification from Stack Overflow:

Cluster 0 : Java language users

Cluster 1 : Language agnostic users

Cluster 2 : .net users

Cluster 3 : SQL users

These were from visual inspection of the tags used. For Cluster 1 however, these refer to posts that do not consider a language or probably the question is for any programming language.

Acknowledgements

I would like to thank Prof. Christian Ais, Prof. Erika Fille Legara, Patricia Manasan, Dr. Joseph Bunao and Jon Colipapa for their help in the construction of this notebook.