In [184]:

```
from lda2vec import preprocess, Corpus
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

try:
    import seaborn
except:
    pass
```

You must be using a very recent version of pyLDAvis to use the lda2vec outputs. As of this writing, anything past Jan 6 2016 or this commit `14e7b5f60d8360eb84969ff6` can do this quickly by installing it directly from master like so:

In [ ]:

```
# pip install git+https://github.com/bmabey/pyLDAvis.git@master#egg=pyLDAvis
```

In [11]:

```
import pyLDAvis
pyLDAvis.enable_notebook()
```

Reading in the saved model topics

After runnning `lda2vec_run.py` script in `examples/twenty_newsgroups/lda2vec` directory a `topics.pyldavis.npz` will be created that contains the topic-to-word left is to visualize and label each topic from the it's prevalent words.

In [157]:

```
npz = np.load(open('topics.pyldavis.npz', 'r'))
dat = {k: v for (k, v) in npz.iteritems()}
dat['vocab'] = dat['vocab'].tolist()
# dat['term_frequency'] = dat['term_frequency'] * 1.0 / dat['term_frequency'].sum()
```

In [189]:

```
top_n = 10
topic_to_topwords = {}
for j, topic_to_word in enumerate(dat['topic_term_dists']):
    top = np.argsort(topic_to_word)[::-1][:top_n]
    msg = 'Topic %i '  % j
    top_words = [dat['vocab'][i].strip()[:35] for i in top]
    msg += ' '.join(top_words)
    print msg
    topic_to_topwords[j] = top_words
```

```
Topic 0 x11r5 xv window xterm server motif font xlib // sunos
Topic 1 jesus son father matthew sin mary g'd disciples christ sins
Topic 2 s1 nsa s2 clipper chip administration q escrow private sector serial number encryption technology
Topic 3 leafs games playoffs hockey game players pens yankees bike phillies
Topic 4 van - 0 pp en 1 njd standings 02 6
Topic 5 out_of_vocabulary out_of_vocabulary anonymity hiv homicide adl ripem bullock encryption technology eff
Topic 6 hiv magi prof erzurum venus van 2.5 million ankara satellite launched
Topic 7 nsa escrow clipper chip encryption government phones warrant vat decrypt wiretap
Topic 8 mac controller shipping disk printer mb ethernet enable os/2 port
Topic 9 leafs cooper weaver karabagh myers agdam phillies flyers playoffs fired
Topic 10 obfuscated = ciphertext jesus gentiles matthew judas { x int
Topic 11 jesus ra bobby faith god homosexuality bible sin msg islam
Topic 12 jesus sin scripture matthew christ islam god sins prophet faith
Topic 13 mac i thanks monitor apple upgrade card connect using windows
Topic 14 i quadra monitor my apple duo hard drive mac mouse thanks
Topic 15 { shipping } + mac mb os/2 $ 3.5 manuals
Topic 16 playoffs morris yankees leafs // pitching players } team wins
Topic 17 :> taxes guns flame .. clinton kids jobs hey drugs
Topic 18 revolver tires pitching saturn ball trigger car ice team engine
Topic 19 stephanopoulos leafs mamma karabagh mr. koresh apartment fired myers sumgait
```

Visualize topics

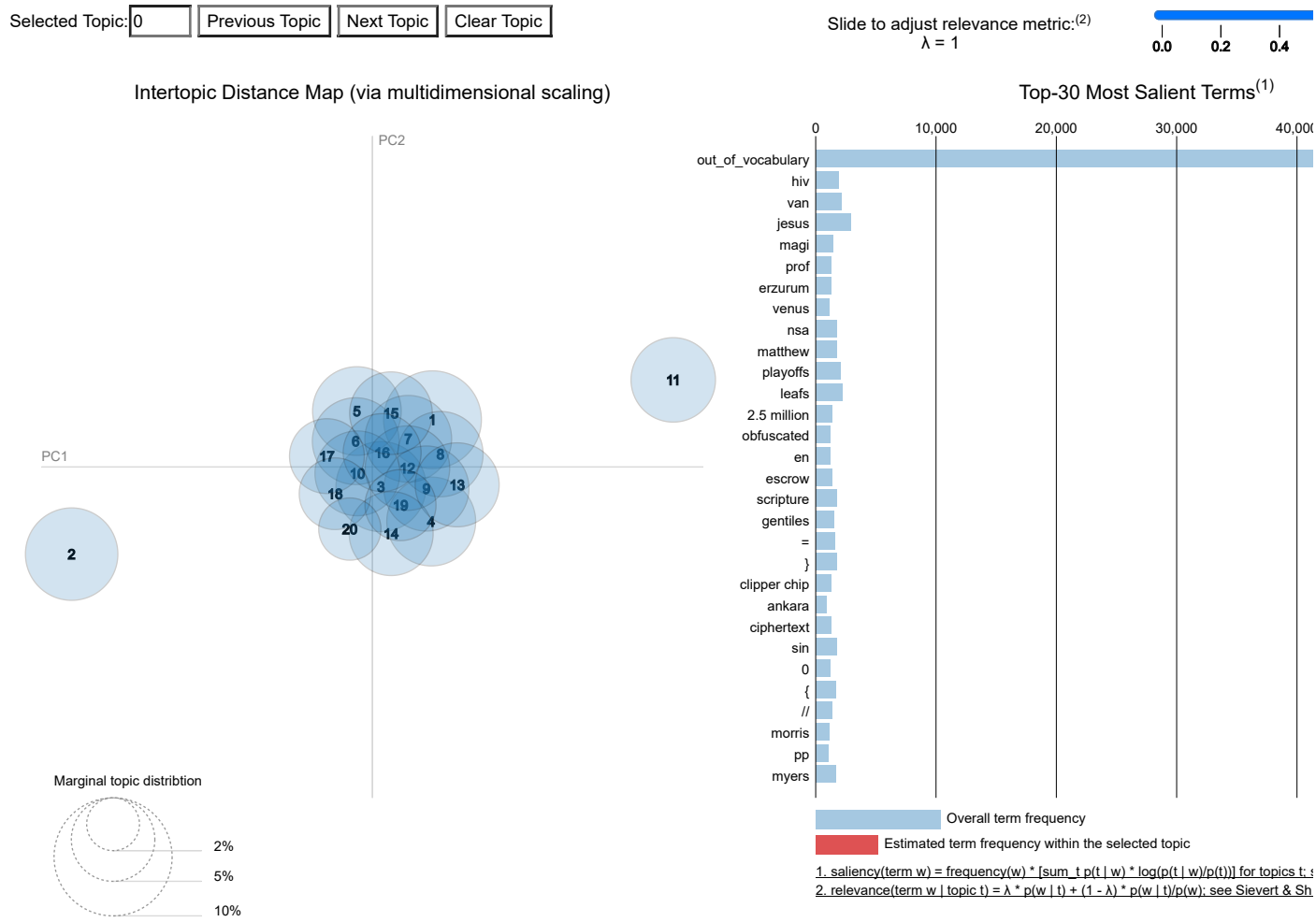
In [187]:

```
import warnings
warnings.filterwarnings('ignore')
prepared_data = pyLDAvis.prepare(dat['topic_term_dists'], dat['doc_topic_dists'],
                                 dat['doc_lengths'] * 1.0, dat['vocab'], dat['term_frequency'] * 1.0, mds='tsne')
```

In [188]:

```
pyLDAvis.display(prepared_data)
```

Out[188]:



'True' topics

The 20 newsgroups dataset is interesting because users effetively classify the topics by posting to a particular newsgroup. This lets us qualitatively check our unsupervise example, the four topics we highlighted above are intuitively close to comp.graphics, sci.med, talk.politics.misc, and sci.space.

```
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
```

```
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
sci.crypt
sci.electronics
sci.med
sci.space
misc.forsale
talk.politics.misc
talk.politics.guns
talk.politics.mideast
talk.religion.misc
alt.atheism
soc.religion.christian
```

Individual document topics

In [248]:

```
from sklearn.datasets import fetch_20newsgroups
remove=('headers', 'footers', 'quotes')
texts = fetch_20newsgroups(subset='train', remove=remove).data
```

First Example

In [249]:

```
print texts[1]
```

```
A fair number of brave souls who upgraded their SI clock oscillator have
shared their experiences for this poll. Please send a brief message detailing
your experiences with the procedure. Top speed attained, CPU rated speed,
add on cards and adapters, heat sinks, hour of usage per day, floppy disk
functionality with 800 and 1.4 m floppies are especially requested.

I will be summarizing in the next two days, so please add to the network
knowledge base if you have done the clock upgrade and haven't answered this
poll. Thanks.
```

In [250]:

```
msg = "{weight:02d}% in topic {topic_id:02d} which has top words {text:s}"
for topic_id, weight in enumerate(dat['doc_topic_dists'][1]):
    if weight > 0.01:
        text = ', '.join(topic_to_topwords[topic_id])
        print msg.format(topic_id=topic_id, weight=int(weight * 100.0), text=text)
```

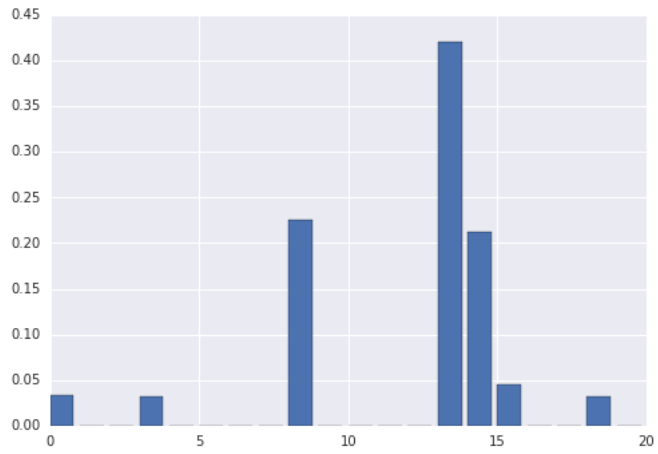
```
03% in topic 00 which has top words x11r5, xv, window, xterm, server, motif, font, xlib, //, sunos
03% in topic 03 which has top words leafs, games, playoffs, hockey, game, players, pens, yankees, bike, phillies
22% in topic 08 which has top words mac, controller, shipping, disk, printer, mb, ethernet, enable, os/2, port
41% in topic 13 which has top words mac, i, thanks, monitor, apple, upgrade, card, connect, using, windows
21% in topic 14 which has top words i, quadra, monitor, my, apple, duo, hard drive, mac, mouse, thanks
04% in topic 15 which has top words {, shipping, }, +, mac, mb, os/2, $, 3.5, manuals
03% in topic 18 which has top words revolver, tires, pitching, saturn, ball, trigger, car, ice, team, engine
```

In [251]:

```
plt.bar(np.arange(20), dat['doc_topic_dists'][1])
```

Out[251]:

```
<Container object of 20 artists>
```



Second Example

In [255]:

```
print texts[51]
```

```
I have been following this thread on talk.religion,
soc.religion.christian.bible-study and here with interest.  I am amazed at
the different non-biblical argument those who oppose the Sabbath present.

One question comes to mind, especially since my last one was not answered
from Scripture.  Maybe clh may wish to provide the first response.

There is a lot of talk about the Sabbath of the TC being ceremonial.
Answer this:

Since the TC commandments is one law with ten parts on what biblical
basis have you decided that only the Sabbath portion is ceremonial?
OR You say that the seventh-day is the Sabbath but not applicable to
Gentile Christians.  Does that mean the Sabbath commandment has been
annulled?  References please.

If God did not intend His requirements on the Jews to be applicable to
Gentile Christians why did He make it plain that the Gentiles were now
grafted into the commonwealth of Israel?

Darius
```

In [259]:

```
msg = "{weight:02d}% in topic {topic_id:02d} which has top words {text:s}"
for topic_id, weight in enumerate(dat['doc_topic_dists'][51]):
    if weight > 0.01:
        text = ', '.join(topic_to_topwords[topic_id])
        print msg.format(topic_id=topic_id, weight=int(weight * 100.0), text=text)
```

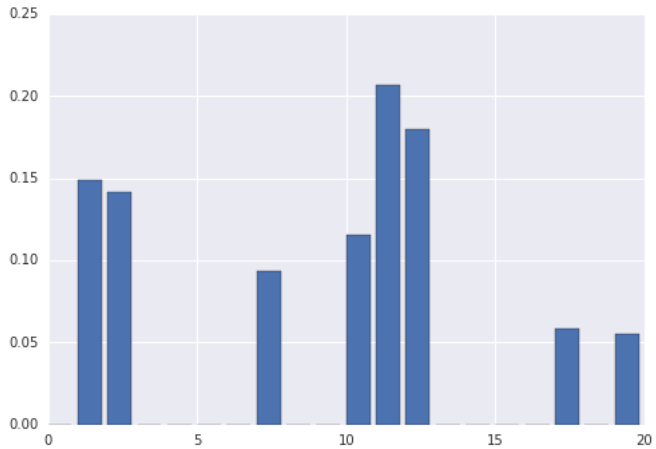
```
14% in topic 01 which has top words jesus, son, father, matthew, sin, mary, g'd, disciples, christ, sins
14% in topic 02 which has top words s1, nsa, s2, clipper chip, administration, q, escrow, private sector, serial number, encryption techr
09% in topic 07 which has top words nsa, escrow, clipper chip, encryption, government, phones, warrant, vat, decrypt, wiretap
11% in topic 10 which has top words obfuscated, =, ciphertext, jesus, gentiles, matthew, judas, {, x, int
20% in topic 11 which has top words jesus, ra, bobby, faith, god, homosexuality, bible, sin, msg, islam
17% in topic 12 which has top words jesus, sin, scripture, matthew, christ, islam, god, sins, prophet, faith
05% in topic 17 which has top words :>, taxes, guns, flame, .., clinton, kids, jobs, hey, drugs
05% in topic 19 which has top words stephanopoulos, leafs, mamma, karabagh, mr., koresh, apartment, fired, myers, sumgait
```

In [260]:

```
plt.bar(np.arange(20), dat['doc_topic_dists'][51])
```

Out[260]:

```
<Container object of 20 artists>
```



In [ ]: