Homework 4

October 23, 2015

1 Homework 4

1.1 Alex Pine, akp258

1.1.1 Question 1: Topic modeling code

```
In [ ]: # Hack to get python to look for the pip modules before the OS X versions.
        # This ensures the newest version of the 'six' library is used, which gensim requires.
        import sys
        sys.path.insert(0, '/Library/Python/2.7/site-packages')
        import gensim
```

1.a: Prepare document corpus Using the UC Irvine's "Daily Kos" weblog corpus.

```
In [65]: from gensim import corpora, models

         corpus = corpora.UciCorpus('docword.kos.txt', fname_vocab='vocab.kos.txt')
```

1.b Prepare Document Corpus Train LDA models with default parameters. gensim's LDA module defaults to 100 topics.

```
In [315]: def print_top_topics(model, num_topics):
              print 'Number of topics:', model.num_topics
              for i, topic in enumerate(default_model.print_topics(num_topics=num_topics, num_words=6)):
                  print 'Topic', str(i+1), ':', topic
```

```
In [313]: # Defaults to num_topics=100
          default_model = models.LdaModel(corpus, id2word=corpus.create_dictionary())
```

```
WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the nu
```

```
In [314]: print_top_topics(default_model, 100)
```

```
Number of topics: 100
Topic 1 : 0.013*kerry + 0.013*november + 0.007*war + 0.007*media + 0.006*bush + 0.006*senate
Topic 2 : 0.033*november + 0.015*poll + 0.010*republicans + 0.009*voting + 0.009*election + 0.009*gover
Topic 3 : 0.030*kerry + 0.027*poll + 0.014*bush + 0.010*polls + 0.008*national + 0.008*percent
Topic 4 : 0.014*kuhl + 0.009*kerry + 0.007*bills + 0.007*bush + 0.007*vice + 0.006*people
Topic 5 : 0.022*bush + 0.016*kerry + 0.008*vote + 0.007*november + 0.006*senate + 0.006*president
Topic 6 : 0.021*bush + 0.011*kerry + 0.009*states + 0.008*republican + 0.007*federal + 0.006*democratic
Topic 7 : 0.013*kerry + 0.011*push + 0.011*shares + 0.011*dean + 0.010*bush + 0.010*economists
Topic 8 : 0.013*senate + 0.010*elections + 0.009*poll + 0.008*seat + 0.007*house + 0.007*coors
Topic 9 : 0.034*bush + 0.027*kerry + 0.017*luntz + 0.009*swift + 0.009*boat + 0.007*poll
Topic 10 : 0.020*bush + 0.009*democrats + 0.009*republicans + 0.008*kerry + 0.007*democratic + 0.007*ca
Topic 11 : 0.011*texas + 0.010*time + 0.009*bush + 0.009*court + 0.007*democrats + 0.006*democratic
```

```
Topic 12 : 0.017*bush + 0.008*democratic + 0.007*jobs + 0.007*vote + 0.005*time + 0.005*senate
Topic 13 : 0.027*bush + 0.026*nader + 0.021*kerry + 0.011*general + 0.011*poll + 0.008*states
Topic 14 : 0.010*cattle + 0.009*award + 0.009*november + 0.008*oreilly + 0.008*bush + 0.008*kerry
Topic 15 : 0.018*bush + 0.015*percent + 0.014*kerry + 0.008*poll + 0.008*general + 0.006*president
Topic 16 : 0.012*senate + 0.010*bush + 0.009*kerry + 0.008*referendum + 0.008*november + 0.007*campaign
Topic 17 : 0.029*thune + 0.016*explosion + 0.013*brock + 0.012*media + 0.012*forms + 0.009*thunes
Topic 18 : 0.033*dean + 0.031*clark + 0.029*lieberman + 0.023*democratic + 0.020*dec + 0.018*poll
Topic 19 : 0.023*november + 0.011*amp + 0.010*voting + 0.010*danielua + 0.010*pride + 0.010*nprigo
Topic 20 : 0.037*bush + 0.011*percent + 0.011*president + 0.011*poll + 0.009*administration + 0.009*busl
Topic 21 : 0.056*harris + 0.021*nag + 0.013*katherine + 0.009*discover + 0.007*war + 0.007*invitation
Topic 22 : 0.014*bush + 0.013*iraq + 0.009*kerry + 0.009*war + 0.007*gear + 0.007*dumb
Topic 23 : 0.022*coburn + 0.013*coburns + 0.010*democratic + 0.009*percent + 0.008*contracts + 0.008*ker
Topic 24 : 0.026*iran + 0.021*cheney + 0.019*saddam + 0.018*hijackers + 0.014*powell + 0.013*hussein
Topic 25 : 0.015*bush + 0.011*iraq + 0.008*war + 0.007*people + 0.005*poll + 0.005*president
Topic 26 : 0.016*kerry + 0.014*bush + 0.012*dean + 0.010*iowa + 0.009*primary + 0.008*clark
Topic 27 : 0.019*party + 0.015*republican + 0.013*bush + 0.012*democratic + 0.010*states + 0.009*republi
Topic 28 : 0.018*kerry + 0.016*debate + 0.014*campaign + 0.012*democratic + 0.010*republican + 0.009*der
Topic 29 : 0.023*bush + 0.015*iraq + 0.010*war + 0.008*kerry + 0.006*republican + 0.005*president
Topic 30 : 0.017*war + 0.016*iraq + 0.012*bush + 0.008*administration + 0.007*president + 0.007*news
Topic 31 : 0.008*space + 0.007*bush + 0.007*kerry + 0.006*tnr + 0.005*smackdown + 0.004*poll
Topic 32 : 0.018*million + 0.014*states + 0.008*percent + 0.007*race + 0.007*bush + 0.007*growth
Topic 33 : 0.014*bush + 0.009*dingell + 0.008*november + 0.007*house + 0.007*guard + 0.006*president
Topic 34 : 0.020*bush + 0.011*kerry + 0.010*election + 0.010*november + 0.008*general + 0.007*state
Topic 35 : 0.012*war + 0.010*bush + 0.009*iraq + 0.008*attacks + 0.008*blogpac + 0.006*administration
Topic 36 : 0.022*delay + 0.017*bush + 0.015*million + 0.012*kerry + 0.012*campaign + 0.011*dean
Topic 37 : 0.026*bush + 0.017*november + 0.010*president + 0.009*kerry + 0.007*house + 0.007*senate
Topic 38 : 0.018*gep + 0.010*bush + 0.008*dean + 0.007*democrats + 0.007*house + 0.007*campaign
Topic 39 : 0.015*bush + 0.008*cheney + 0.006*state + 0.005*president + 0.005*general + 0.005*race
Topic 40 : 0.010*leaks + 0.010*leaking + 0.009*west + 0.008*updates + 0.008*classified + 0.007*outfit
Topic 41 : 0.011*bush + 0.010*house + 0.010*november + 0.009*campaign + 0.006*consultants + 0.006*media
Topic 42 : 0.012*kerry + 0.010*bush + 0.005*nyt + 0.005*war + 0.005*john + 0.005*general
Topic 43 : 0.057*november + 0.012*poll + 0.011*house + 0.011*account + 0.010*governor + 0.010*vote
Topic 44 : 0.019*bush + 0.011*kerry + 0.007*poll + 0.006*news + 0.006*democratic + 0.006*konop
Topic 45 : 0.009*party + 0.007*general + 0.006*media + 0.006*airport + 0.005*bush + 0.005*staged
Topic 46 : 0.017*sessions + 0.014*bin + 0.014*frost + 0.013*bush + 0.010*laden + 0.009*lakes
Topic 47 : 0.013*iraq + 0.007*kerry + 0.007*apple + 0.006*general + 0.006*vanity + 0.005*bush
Topic 48 : 0.016*november + 0.012*bush + 0.007*poll + 0.006*kerry + 0.006*republicans + 0.006*turnout
Topic 49 : 0.040*kerry + 0.036*edwards + 0.023*parenthesis + 0.022*poll + 0.021*undecided + 0.017*gephar
Topic 50 : 0.042*dean + 0.022*kerry + 0.020*clark + 0.014*campaign + 0.013*democratic + 0.013*primary
Topic 51 : 0.018*disappear + 0.016*misled + 0.015*owner + 0.012*ohio + 0.012*iraq + 0.010*stations
Topic 52 : 0.015*house + 0.010*bush + 0.007*outreach + 0.007*republican + 0.007*people + 0.006*republica
Topic 53 : 0.009*rogers + 0.008*cheney + 0.007*people + 0.007*bush + 0.006*department + 0.006*civilized
Topic 54 : 0.022*bush + 0.011*president + 0.010*kerry + 0.009*iraq + 0.007*general + 0.006*war
Topic 55 : 0.031*tax + 0.025*gotv + 0.023*bush + 0.022*kerry + 0.013*kucinich + 0.012*results
Topic 56 : 0.031*bush + 0.024*november + 0.012*poll + 0.008*kerry + 0.008*war + 0.007*house
Topic 57 : 0.010*kerry + 0.009*edwards + 0.007*democrats + 0.007*bush + 0.006*party + 0.006*democratic
Topic 58 : 0.011*iraqi + 0.008*johnson + 0.007*forces + 0.007*insurgents + 0.007*iraq + 0.007*general
Topic 59 : 0.028*bush + 0.006*kerry + 0.005*campaign + 0.005*bushs + 0.005*house + 0.005*texas
Topic 60 : 0.035*bush + 0.021*kerry + 0.008*general + 0.008*campaign + 0.007*poll + 0.006*bushs
Topic 61 : 0.021*district + 0.019*bush + 0.012*veterans + 0.011*schrock + 0.010*vietnam + 0.009*kerry
Topic 62 : 0.016*bush + 0.010*administration + 0.010*president + 0.008*ban + 0.007*officials + 0.006*hou
Topic 63 : 0.013*browser + 0.011*lightbulb + 0.011*delay + 0.011*bush + 0.010*bloggers + 0.010*arabia
Topic 64 : 0.014*bush + 0.007*tariffs + 0.005*iraq + 0.005*arnold + 0.005*president + 0.005*steel
Topic 65 : 0.019*winnable + 0.018*house + 0.012*bush + 0.009*sanctions + 0.008*daschle + 0.007*nelson
```

```
Topic 66 : 0.016*romney + 0.015*calm + 0.011*war + 0.010*generic + 0.009*republicans + 0.009*message
Topic 67 : 0.021*bush + 0.012*kerry + 0.010*media + 0.009*general + 0.008*state + 0.008*campaign
Topic 68 : 0.018*bush + 0.009*iraq + 0.008*kerry + 0.008*state + 0.007*war + 0.006*general
Topic 69 : 0.011*jenna + 0.008*card + 0.008*gotv + 0.008*time + 0.008*bush + 0.007*general
Topic 70 : 0.033*iraq + 0.013*iraqi + 0.010*military + 0.009*american + 0.008*people + 0.007*baghdad
Topic 71 : 0.013*bush + 0.011*president + 0.006*war + 0.006*kerry + 0.006*debate + 0.006*space
Topic 72 : 0.013*bush + 0.009*kerry + 0.009*war + 0.007*november + 0.007*girly + 0.007*general
Topic 73 : 0.029*gdp + 0.015*borders + 0.012*bush + 0.010*administration + 0.008*news + 0.007*quarter
Topic 74 : 0.017*carson + 0.014*coburn + 0.013*republican + 0.011*bush + 0.010*race + 0.010*campaign
Topic 75 : 0.019*debate + 0.013*bunning + 0.013*filibuster + 0.012*court + 0.010*kerry + 0.009*edwards
Topic 76 : 0.019*kerry + 0.013*poll + 0.010*bush + 0.010*dean + 0.009*november + 0.008*gephardt
Topic 77 : 0.028*iraq + 0.024*war + 0.011*bush + 0.009*melanie + 0.008*troops + 0.006*bushs
Topic 78 : 0.015*kerry + 0.013*bush + 0.010*afscme + 0.010*democratic + 0.009*labor + 0.008*poll
Topic 79 : 0.010*bush + 0.007*president + 0.007*convention + 0.007*house + 0.006*dozen + 0.006*republica
Topic 80 : 0.016*kerry + 0.009*winner + 0.009*john + 0.008*numbers + 0.008*democratic + 0.007*susa
Topic 81 : 0.022*war + 0.012*debate + 0.011*bush + 0.009*debates + 0.008*iraq + 0.006*melanie
Topic 82 : 0.014*seat + 0.011*gop + 0.010*house + 0.009*candidate + 0.008*democratic + 0.008*primary
Topic 83 : 0.033*reid + 0.013*november + 0.011*bush + 0.010*threeway + 0.010*vote + 0.010*republicans
Topic 84 : 0.014*bush + 0.012*energy + 0.008*hoeffel + 0.006*kerry + 0.005*iraq + 0.005*war
Topic 85 : 0.028*dean + 0.028*seiu + 0.024*unions + 0.019*union + 0.017*gephardt + 0.015*afscme
Topic 86 : 0.055*ethics + 0.028*house + 0.025*committee + 0.021*delay + 0.016*republicans + 0.014*compla
Topic 87 : 0.012*november + 0.012*house + 0.007*poll + 0.007*democratic + 0.006*kerry + 0.005*bush
Topic 88 : 0.040*november + 0.016*republicans + 0.016*vote + 0.015*senate + 0.012*poll + 0.011*governor
Topic 89 : 0.014*bush + 0.007*act + 0.007*house + 0.007*white + 0.006*time + 0.006*social
Topic 90 : 0.020*kerry + 0.019*voters + 0.016*bush + 0.016*percent + 0.015*poll + 0.010*results
Topic 91 : 0.013*race + 0.012*percent + 0.009*poll + 0.007*senate + 0.007*bush + 0.006*republican
Topic 92 : 0.011*district + 0.010*house + 0.010*poll + 0.010*race + 0.010*bush + 0.009*democrats
Topic 93 : 0.010*intimidate + 0.010*ashamed + 0.009*kerry + 0.009*reputation + 0.008*chapter + 0.008*bus
Topic 94 : 0.011*ballot + 0.010*party + 0.008*state + 0.008*nader + 0.008*general + 0.007*republican
Topic 95 : 0.012*cheney + 0.011*iraq + 0.010*president + 0.010*bush + 0.008*percent + 0.007*environmenta
Topic 96 : 0.011*beef + 0.011*war + 0.010*bush + 0.007*time + 0.007*iraq + 0.006*military
Topic 97 : 0.036*meetup + 0.028*sanctions + 0.020*indicted + 0.014*database + 0.011*failed + 0.009*senat
Topic 98 : 0.044*november + 0.016*poll + 0.014*senate + 0.012*house + 0.011*republicans + 0.010*exit
Topic 99 : 0.030*fox + 0.017*species + 0.012*seats + 0.012*news + 0.010*monkeys + 0.010*html
Topic 100 : 0.019*november + 0.014*bush + 0.009*house + 0.008*poll + 0.007*iraq + 0.007*kerry
```

Analysis The default model finds 100 different topics. The Daily KOS is a blog about US politics, and the topics discovered by LDA reflect this. The first topic largely refers to the 2004 US presidential election ("kerry", "november", "war", "bush"). Nearly all of the other topics are political topics (e.g. one topic has the words: "war", "bush", "iraq", "attacks"). There is a great deal of overlap between these topics.

1.c Try different values for num_topics Trying out the same model with 5, 10, and 25 different topics.

```
In [316]: num_topics_list = [5, 10, 25]
          for num_topics in num_topics_list:
              model = models.LdaModel(corpus, num_topics=num_topics, id2word=corpus.create_dictionary()
              print_top_topics(model, num_topics)

WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the nu
WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the nu

Number of topics: 5
Topic 1 : 0.017*carson + 0.014*coburn + 0.013*republican + 0.011*bush + 0.010*race + 0.010*campaign
Topic 2 : 0.013*browser + 0.011*lightbulb + 0.011*delay + 0.011*bush + 0.010*bloggers + 0.010*arabia
```

```
Topic 3 : 0.012*november + 0.012*house + 0.007*poll + 0.007*democratic + 0.006*kerry + 0.005*bush
Topic 4 : 0.010*bush + 0.007*president + 0.007*convention + 0.007*house + 0.006*dozen + 0.006*republica
Topic 5 : 0.022*bush + 0.011*president + 0.010*kerry + 0.009*iraq + 0.007*general + 0.006*war
Number of topics:

WARNING:gensim.models.ldamodel:too few updates, training might not converge; consider increasing the nu

10
Topic 1 : 0.021*bush + 0.011*kerry + 0.009*states + 0.008*republican + 0.007*federal + 0.006*democratic
Topic 2 : 0.010*intimidate + 0.010*ashamed + 0.009*kerry + 0.009*reputation + 0.008*chapter + 0.008*bus
Topic 3 : 0.014*bush + 0.013*iraq + 0.009*kerry + 0.009*war + 0.007*gear + 0.007*dumb
Topic 4 : 0.033*dean + 0.031*clark + 0.029*lieberman + 0.023*democratic + 0.020*dec + 0.018*poll
Topic 5 : 0.010*cattle + 0.009*award + 0.009*november + 0.008*oreilly + 0.008*bush + 0.008*kerry
Topic 6 : 0.019*winnable + 0.018*house + 0.012*bush + 0.009*sanctions + 0.008*daschle + 0.007*nelson
Topic 7 : 0.044*november + 0.016*poll + 0.014*senate + 0.012*house + 0.011*republicans + 0.010*exit
Topic 8 : 0.022*coburn + 0.013*coburns + 0.010*democratic + 0.009*percent + 0.008*contracts + 0.008*ker
Topic 9 : 0.019*party + 0.015*republican + 0.013*bush + 0.012*democratic + 0.010*states + 0.009*republi
Topic 10 : 0.011*iraqi + 0.008*johnson + 0.007*forces + 0.007*insurgents + 0.007*iraq + 0.007*general
Number of topics: 25
Topic 1 : 0.018*disappear + 0.016*misled + 0.015*owner + 0.012*ohio + 0.012*iraq + 0.010*stations
Topic 2 : 0.013*senate + 0.010*elections + 0.009*poll + 0.008*seat + 0.007*house + 0.007*coors
Topic 3 : 0.020*kerry + 0.019*voters + 0.016*bush + 0.016*percent + 0.015*poll + 0.010*results
Topic 4 : 0.019*party + 0.015*republican + 0.013*bush + 0.012*democratic + 0.010*states + 0.009*republi
Topic 5 : 0.010*intimidate + 0.010*ashamed + 0.009*kerry + 0.009*reputation + 0.008*chapter + 0.008*bus
Topic 6 : 0.019*debate + 0.013*bunning + 0.013*filibuster + 0.012*court + 0.010*kerry + 0.009*edwards
Topic 7 : 0.031*bush + 0.024*november + 0.012*poll + 0.008*kerry + 0.008*war + 0.007*house
Topic 8 : 0.011*jenna + 0.008*card + 0.008*gotv + 0.008*time + 0.008*bush + 0.007*general
Topic 9 : 0.040*kerry + 0.036*edwards + 0.023*parenthesis + 0.022*poll + 0.021*undecided + 0.017*gephar
Topic 10 : 0.013*kerry + 0.013*november + 0.007*war + 0.007*media + 0.006*bush + 0.006*senate
Topic 11 : 0.011*iraqi + 0.008*johnson + 0.007*forces + 0.007*insurgents + 0.007*iraq + 0.007*general
Topic 12 : 0.019*kerry + 0.013*poll + 0.010*bush + 0.010*dean + 0.009*november + 0.008*gephardt
Topic 13 : 0.018*gep + 0.010*bush + 0.008*dean + 0.007*democrats + 0.007*house + 0.007*campaign
Topic 14 : 0.018*million + 0.014*states + 0.008*percent + 0.007*race + 0.007*bush + 0.007*growth
Topic 15 : 0.022*bush + 0.016*kerry + 0.008*vote + 0.007*november + 0.006*senate + 0.006*president
Topic 16 : 0.014*seat + 0.011*gop + 0.010*house + 0.009*candidate + 0.008*democratic + 0.008*primary
Topic 17 : 0.016*romney + 0.015*calm + 0.011*war + 0.010*generic + 0.009*republicans + 0.009*message
Topic 18 : 0.028*dean + 0.028*seiu + 0.024*unions + 0.019*union + 0.017*gephardt + 0.015*afscme
Topic 19 : 0.010*leaks + 0.010*leaking + 0.009*west + 0.008*updates + 0.008*classified + 0.007*outfit
Topic 20 : 0.013*bush + 0.011*president + 0.006*war + 0.006*kerry + 0.006*debate + 0.006*space
Topic 21 : 0.012*kerry + 0.010*bush + 0.005*nyt + 0.005*war + 0.005*john + 0.005*general
Topic 22 : 0.034*bush + 0.027*kerry + 0.017*luntz + 0.009*swift + 0.009*boat + 0.007*poll
Topic 23 : 0.019*november + 0.014*bush + 0.009*house + 0.008*poll + 0.007*iraq + 0.007*kerry
Topic 24 : 0.030*fox + 0.017*species + 0.012*seats + 0.012*news + 0.010*monkeys + 0.010*html
Topic 25 : 0.013*browser + 0.011*lightbulb + 0.011*delay + 0.011*bush + 0.010*bloggers + 0.010*arabia
```

Analysis As the number of topics increases, they seem to become more coherent. The model with only 5 topics has some topics that make no intuitive sense to me. For example, What do "browser", "lightbulb", "delay,"bush","bloggers", and"arabia" have to do with each other? The 10 topic model is more coherent, but still has some strange topic groups: "cattle", "award", "november", "oreilly", "bush", "kerry"? The last four words are about the 2004 presidential election, but I have no idea what the first two words have to do with that topic. Finally, the topics generated from the 25 topic model look very similar to the 100 topic model, suggesting the actual number of topics is somewhere between 25 and 100.

1.2 Question 2 and Question 3

Question 2 and question 3 can be found in a seperate PDF file that was submitted alongside this PDF.

1.3 Question 4

I collaborated with Israel Malkin, Maya Rotmensch, Charlie Guthrie, Peter Li, and Justin Mao-Jones on this problem.

In [224]:
```
# Code that reads in data files for question 4

import os

class Doc:
    def __init__(self, num_topics, topic_priors, word_priors):
        self.num_topics = num_topics
        self.topic_priors = topic_priors  # alpha.
        self.word_priors = word_priors  # beta


def parse_input_file(filename):
    num_topics = 0
    # Dirichlet hyperparams, aka alphas
    topic_priors = []
    # Beta prior for this document, words are rows, topic probabilities are columns
    word_priors = {}

    with open(filename, 'r') as f:
        lines = [line for line in f]
        num_topics = int(lines[0])
        assert(num_topics > 0)
        topic_priors = [float(tok.strip()) for tok in lines[1].split()]
        assert(len(topic_priors) == num_topics)
        for word_index, line in enumerate(lines[2:]):
            tokens = line.split()
            word = tokens[0].strip()  # not used
            word_probs = [float(tok.strip()) for tok in tokens[1:]]
            assert(len(word_probs) == num_topics)
            word_priors[word_index] = word_probs
    return num_topics, topic_priors, word_priors


doc = Doc(*parse_input_file('ps4_data/abstract_nips21_NIPS2008_0517.txt.ready'))
```

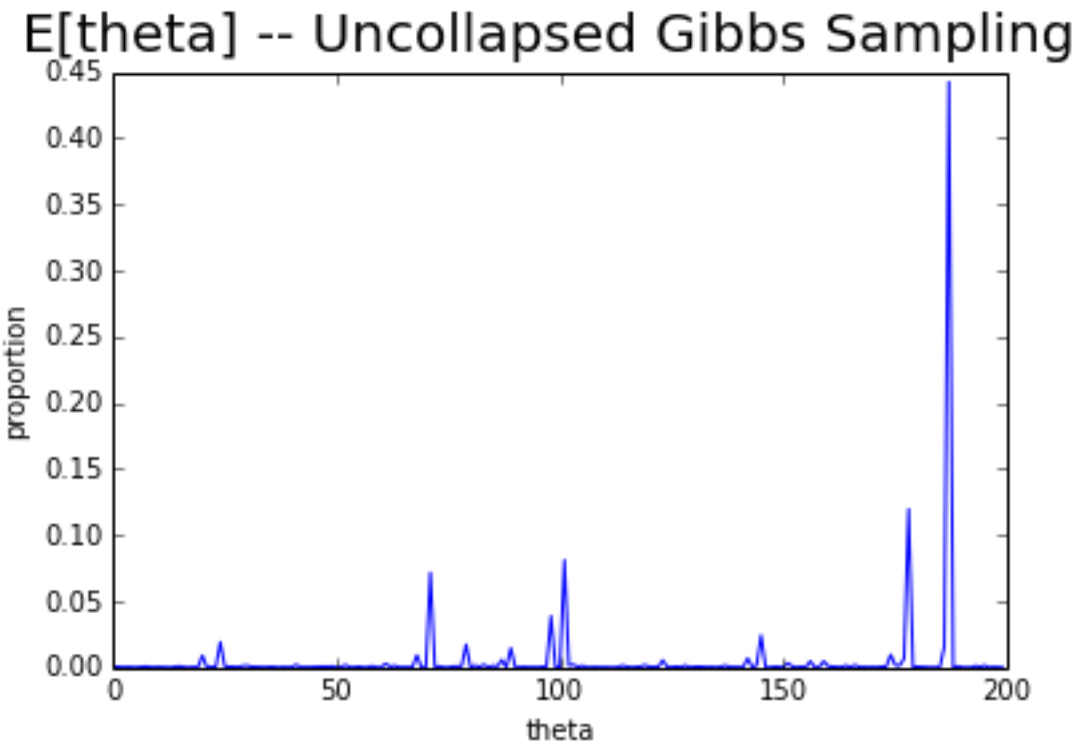
In [307]:
```
%matplotlib inline

import matplotlib
import matplotlib.pyplot as plt
import numpy as np
from numpy.random import mtrand

# Sample a topic probability (theta) for the uncollapsed sampler.
def sample_topic_dist(topic_priors, topics):
    topic_counts = np.bincount(topics, minlength=len(topic_priors))
    posterior_topic_priors = [prior + count
```

```
                                    for prior, count in zip(topic_priors, topic_counts)]
        return mtrand.dirichlet(posterior_topic_priors)


    # Create the posterior probabilities for topics (z) for the uncollapsed sampler.
    def sample_posterior_topic(word_index, word_priors, topic_dist):
        posterior_topic_probs = []
        denominator = 0.0
        word_prior_list = word_priors[word_index]
        for topic_index in range(len(topic_dist)):
            numerator = word_prior_list[topic_index] * topic_dist[topic_index]
            posterior_topic_probs.append(numerator)
            denominator += numerator
        posterior_topic_probs = [prob/denominator for prob in posterior_topic_probs]
        topic_counts = mtrand.multinomial(1, posterior_topic_probs)
        for topic_index, sample_value in enumerate(topic_counts):
            if sample_value == 1:
                return topic_index
        raise Exception('Error occured while sampling topic')


    # Returns an array of topic distribution samples
    def uncollapsed_gibbs_sampler(doc, num_iterations):
        # Initialize the topic_dist and topics to dummy values to start.
        initial_topic_dist = [1.0/doc.num_topics]*num_topics
        initial_topics = [1]*len(doc.word_priors)
        topic_dist_samples = [initial_topic_dist]
        topic_samples = [initial_topics]

        for iteration in range(num_iterations):
            prev_topics = topic_samples[-1]
            # Sample topic distribution (theta)
            topic_dist_sample = sample_topic_dist(doc.topic_priors, prev_topics)
            # Initialize the topic sample to be the sample as the last one
            topics_sample = list(prev_topics)
            for i in range(len(topics_sample)):
                # Sample each topic instantiation (z_{mn})
                topics_sample[i] = sample_posterior_topic(i, doc.word_priors,
                                                          topic_dist_sample)
            topic_dist_samples.append(topic_dist_sample)
            topic_samples.append(topics_sample)
        # Remove the 'burn' samples
        topic_dist_samples = topic_dist_samples[50:]
        return np.array(topic_dist_samples)


    def uncollapsed_expected_topic_dist(samples):
        return np.mean(samples, axis=0)
```

In [290]:
```
    # Uncollapsed topic distribution samples
    u_topic_dist_samples = uncollapsed_gibbs_sampler(doc, 10000)
```

In [292]:
```
    u_topic_dist = uncollapsed_expected_topic_dist(u_topic_dist_samples)
    fig = plt.figure()
    fig.suptitle('E[theta] -- Uncollapsed Gibbs Sampling', fontsize=20)
```

```
plt.xlabel('theta')
plt.ylabel('proportion')
plt.plot(range(len(u_topic_dist)), u_topic_dist)
plt.show()
```

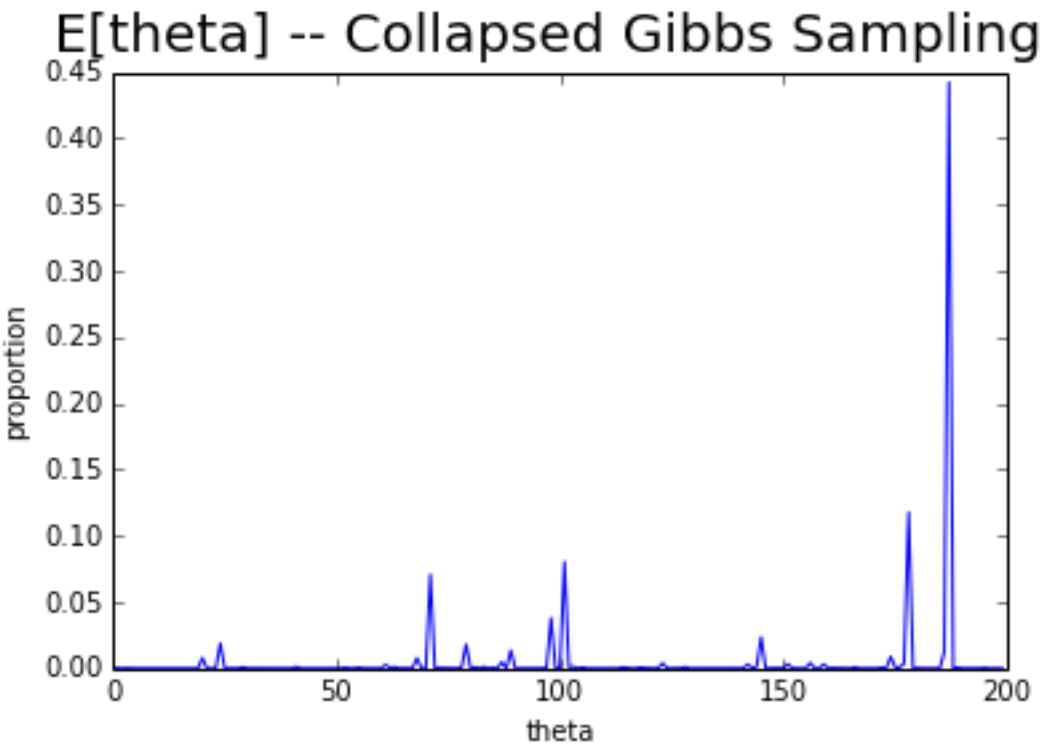


In [254]:
```
# Collapsed Gibbs Sampling

# Conditional probability of
def sample_posterior_topic_collapsed(word_index, topic_sample, word_priors, topic_priors):
    # Bucket topic samples, excluding the current topic sample
    topic_counts = [0]*len(topic_priors)
    for i, topic in enumerate(topic_sample):
        if i != word_index:
            topic_counts[topic] += 1
    # Compute each posterior topic probability
    posterior_topic_probs = []
    for topic_index in range(len(topic_priors)):
        word_prior = word_priors[word_index][topic_index]
        topic_prior = topic_priors[topic_index]
        topic_count = topic_counts[topic_index]
        prob = word_prior * (topic_prior + topic_count)
        posterior_topic_probs.append(prob)
    normalizer = sum(posterior_topic_probs)
    posterior_topic_probs = [prob/normalizer for prob in posterior_topic_probs]
    # Sample from the distribution
    sample = mtrand.multinomial(1, posterior_topic_probs)
```

```
        for topic_index, sample_value in enumerate(sample):
            if sample_value == 1:
                return topic_index
        raise Exception('Error occured while sampling topic')


    # Returns an array of topic samples
    def collapsed_gibbs_sampler(doc, num_iterations):
        # Initialize the topics to dummy values to start.
        initial_topics = [1]*len(doc.word_priors)
        topic_samples = [initial_topics]
        for iteration in range(num_iterations):
            topic_sample = list(topic_samples[-1])
            for i in range(len(topic_sample)):
                # Sample each topic instantiation (z_{mn})
                topic_sample[i] = sample_posterior_topic_collapsed(
                    i, topic_sample, doc.word_priors, doc.topic_priors)
            topic_samples.append(topic_sample)
        # Remove the 'burn' samples
        topic_samples = topic_samples[50:]
        return np.array(topic_samples)


    # Returns the expected value of the topic distribution (theta).
    def collapsed_expected_topic_dist(topic_samples, topic_priors):
        T = len(topic_samples)
        topic_dist = np.zeros(len(topic_priors))
        for topic_sample in topic_samples:
            topic_dist += np.bincount(topic_sample, minlength=len(topic_priors))
        N = len(topic_samples[0])
        topic_dist += np.array([N*topic_prior for topic_prior in topic_priors])
        topic_dist /= T * (sum(topic_priors) + N)
        return topic_dist
```

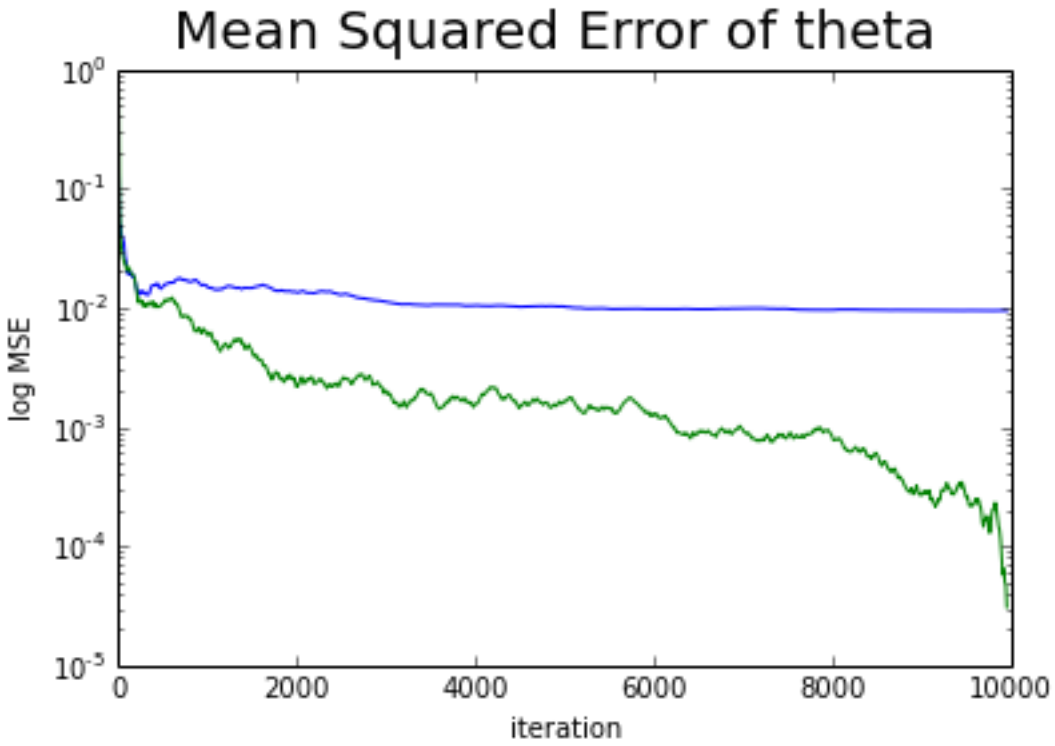
In [260]:
```
    # Collapsed topic distribution samples
    c_topic_samples = collapsed_gibbs_sampler(doc, 10000)
```

In [288]:
```
    c_topic_dist = collapsed_expected_topic_dist(c_topic_samples, doc.topic_priors)
    fig = plt.figure()
    fig.suptitle('E[theta] -- Collapsed Gibbs Sampling', fontsize=20)
    plt.xlabel('theta')
    plt.ylabel('proportion')
    plt.plot(range(len(c_topic_dist)), c_topic_dist)
    plt.show()
```



In [312]:
```
# Plotting L2 errors of uncollapsed and collasped Gibbs sampling.

import math
# u_samples are thetas and c_samples are z's.
def plot_error(u_samples, c_samples, topic_priors):
    assert len(u_samples) == len(c_samples)
    # Using collapsed sampler as ground truth
    gt_topic_dist = collapsed_expected_topic_dist(c_samples, topic_priors)
    xaxis = range(1, len(c_samples), 10)
    u_errors = []
    c_errors = []
    for i in xaxis:
        tmp_u_topic_dist = uncollapsed_expected_topic_dist(u_samples[:i])
        u_error = math.sqrt(sum((gt_topic_dist - tmp_u_topic_dist)**2))
        u_errors.append(u_error)
        tmp_c_topic_dist = collapsed_expected_topic_dist(c_samples[:i], topic_priors)
        c_error = math.sqrt(sum((gt_topic_dist - tmp_c_topic_dist)**2))
        c_errors.append(c_error)
    fig = plt.figure()
    fig.suptitle('Mean Squared Error of theta', fontsize=20)
    plt.xlabel('iteration')
    plt.ylabel('log MSE')
    plt.plot(xaxis, u_errors)
    plt.plot(xaxis, c_errors)
    plt.yscale('log')
    plt.show()
```

```
plot_error(u_topic_dist_samples, c_topic_samples, doc.topic_priors)
```



Analysis The blue line charts the mean squared error (MSE) of the uncollapsed Gibbs sampler, while the green line charts the MSE of the collapsed Gibbs sampler. The MSE is shown on a logarithmic scale. The collaspsed sampler is able to continue to decrease its error as the number of iterations increases, while the uncollapsed sampler plateaus after about 3000 iterations.