Summary

Otte Heinävaara

December 18, 2015

1 Introduction

The purpose of this text is to go through the main aspects that came up during the project in the summer and the fall.

2 The Main Problem

The main problem is to investigate how gradient descent based optimization methods behave on certain variational bounds applied to Latent Dirichlet Allocation (LDA) and Mixture of Gaussians (MOG) with the hope of seeing general phenomena relating the performance to the geometry of the method.

3 Implementation

Implementation [INSERT REFERENCE HERE] is based on: [INSERT REFERENCE HERE]. Further modifications made:

3.1 Additions

- Calculation of the gradient and the hessian of the bound with automatic differentiation (Theano).
- (Standard) calculation of the Hessian of the bound of MOG.
- Calculation of the eigenvalue(-like) data from the hessian.

3.2 Findings about modifications

- The calculation of the Hessian turned out to be surprisingly imprecise, with both methods: especially the part with logarightmic determinant. The reason for which is not clear. Possible causes might be: (1) Bug in the code, (2) imprecision in the Theano evaluation AND standard evaluation; both of which sound weird.
- Ignoring precision errors, calculation naturally takes significant amount of time, much smaller than with...
- Eigenvalue calculation, biggest bottleneck.

4 Ways to measure

Ways to pinpoint the behaviour have been devised.

4.1 Findings about ways to measure

- Calculating all the eigenvalues is extremely slow (how slow?) even in the moderately small cases, so that can't be applied to the interesting big cases.
- Estimation of the biggest eigenvalue with power iteration style methods is doable to some extent (to what extent?), but still, eigenvalues being close to zero raise the question whether the hessian was accurate enough.
- In LDA, for which the Hessian calculation works better than MOGs, only small amount f bad eigenvalues were found, leading to the conclusion that the idea of saddle points might be off.
- Estimation of the index via Chebychev polynomials (with the idea of estimating eigenprojection with polynomials) didn't seem to be good idea, since the eigenvalues are clustered around discontinuity, for which the approximation is bad (how bad?).
- The same can be said about naive projections, approximation is very bad, especially with the edge cases (how bad?).
- Checking whether some components wither: for some reason, this seemed to happen in the easier cases (where the clusters are separated), but in the harder cases, all the components remain, with

varying performance. Cavaet: the relation of difficulty and withering was investigated in relatively small cases (in which difficulty could be grasped visually), leading to question whether the observed behaviour is really illustrative.

- How are the found solutions distributed globally? With index shuffling, the solutions showed small amount of clusters.
- Jumping was tested: Converge, jump, converge... with the idea of revealing how locally are the solutions optimal. The tests showed no significant findings: with small jumps, we don't get to better results, with moderately long the variance decreases, until far away we converge mostly to bad results.

5 Data used

Data used was

- Data originally in demos: (1) Data from papers of nips 2011 (2) Original data creators.
- Data creation based on gaussians with random structured covariance matrix.