Scalable Machine Learning CS281B, STAT241B
http://alex.smola.org/teaching/berkeley2012
UC Berkeley

April 17, 2012
due on May 1, 2012
Alexander J. Smola

Problem Set 4 — Dynamic Programming

1 Set Algebra and Junction Trees (10)

1.1 Semiring Property

Denote by $\mathcal{X}$ a domain and let $\cup$ and $\cap$ be set union and intersection between subsets of $\mathcal{X}$ respectively. Prove that they form a semiring. Which operation corresonds to $\oplus$ and $\otimes$ respectively?

1.2 Junction Tree Generation

Denote by $G(V, E)$ a graph with vertices V and edges E. Without loss of generality assume that G is connected. We want to generate a valid junction tree from the graph. As a first step we form a spanning tree T on $G(V, E)$.

To turn this into a proper junction tree we need to attach cliques c_v with each vertex $v \in V$ such that they satisfy the running intersection property. That is, if any variable is part of two cliques then also all cliques on the connecting path must contain this variable. Your goal is to design design an algorithm that achieves this using a message passing algorithm.

1. Denote by D_i the set of variables owned by vertex i. This is the set of variables associated with the clique potential that also contains i. Moreover denote by R_{ij} the set of variables reachable to j from i. Show that R_{ij} satisfies

$$R_{ij} = D_i \cup \left[\bigcup_{k \in \mathcal{N}(i) \setminus \{j\}} R_{ki} \right]. \tag{1}$$

   Here $\mathcal{N}(i)$ denotes the neighbors of i in the spanning tree.
2. As a result of running this algorithm we obtain a set of vertices C_i associated with each vertex i with $i \in C_i$. Show that C_i also contains all vertices v with the property that $v \in R_{ji} \cap R_{j'i}$ with $j \neq j'$. That is, show that

$$C_i = D_i \cup \left[\bigcup_{k,k' \in \mathcal{N}(i) \text{ with } k \neq k'} R_{ki} \cap R_{k'i} \right] \tag{2}$$

   These sets C_i form the junction tree. Note that it is quite common that adjacent C_i are identical.
3. Show that invoking (1) repeatedly on the spanning tree will converge to the correct solution. Hint — show first that the algorithm will converge. Secondly use the semiring property of the set algebra.

Problem Set 4 — Dynamic Programming

2 Sparse Matrices (10)

Sparse matrices share some of the properties typically encountered in graphical models. This is not surprising since typical linear algebra operations arise when dealing with Gaussians. In the following we will derive a number of algorithms in sparse linear algebra.

2.1 Gaussian Markov Random Field

Assume that $x \in \mathbb{R}^n$ is drawn from a Gaussian $x \sim \mathcal{N}(\mu, \Sigma)$ with mean μ and covariance Σ. Moreover assume that there is an undirected graph $G(V, E)$ with $|V| = n$ vertices and E edges which reflect the conditional independence properties observed in x.

1. Derive conditions for sparsity of Σ or Σ^{-1} based on G (argue why you picked Σ or Σ^{-1} respectively). Hint — use the Hammersley Clifford decomposition and compare it to the terms in a Gaussian.
2. Prove that for any graph G there exists a normal distribution that satisfies the associated conditional independence conditions. Hint — give a universal example.

2.2 Banded Matrix

Assume that we are given a symmetric banded diagonal matrix with one off-diagonal row each, that is

$$M = \begin{bmatrix} m_{11} & m_{12} & & & \\ m_{21} & m_{22} & m_{23} & & \\ & m_{23} & m_{33} & m_{34} & \\ & & \ddots & \ddots & \ddots \\ & & m_{n-1,n-2} & m_{n-1,n-1} & m_{n-1,n} \\ & & & m_{n,n-1} & m_{nn} \end{bmatrix}$$

Note all off-diagonal elements beyond the ones described above are zero. Furthermore assume that M has full rank. Derive an efficient $O(n)$ algorithm for solving the equation $Mx = y$.

2.3 Ring Matrix

We now modify the banded matrix M by assuming that also $m_{1n} = m_{n1} \neq 0$, i.e. we now have

$$M = \begin{bmatrix} m_{11} & m_{12} & & & & m_{1n} \\ m_{21} & m_{22} & m_{23} & & & \\ & m_{23} & m_{33} & m_{34} & & \\ & & \ddots & \ddots & \ddots & \\ & & m_{n-1,n-2} & m_{n-1,n-1} & m_{n-1,n} & \\ m_{n1} & & & m_{n,n-1} & m_{nn} & \end{bmatrix}$$

What happens to the problem solving $Mx = y$? State the new algorithm.

2.4 Gaussian Elimination

Assume that we have $x \sim \mathcal{N}(\mu, \Sigma)$ as in Section 2.1. Now assume that we want to integrate out x_i.

1. Given an expression for the reduced distribution. Hint — the random variables are Gaussian.
2. What if we *condition* on a variable? Can you interpret the operation as solving a linear system?

Problem Set 4 — Dynamic Programming

3 Personalized Sequence Recommendation (10)

When users review movies we experience the effect of grounding. That is, the previously watched movie has considerable influence on how the current movie is being rated. For instance, after a very good movie, subsequent ones are measured against a very high standard. On the other hand, compared to *Plan 9 from Outer Space* almost any movie will be considered a masterpiece.

3.1 Independent Model

A first step is to develop an independent regression model. That is, assume that we have a latent factor per user v_u and a latent factor per movie v_m with $v_u, v_m \in \mathbb{R}^d$. Write out the negative log-likelihood for an additive Gaussian approach. It needs to incorporate the following aspects:

- We observe rating pairs (u, m, y) and want to minimize the negative log-likelihood of misprediction, i.e. we assume that

$$y \sim \mathcal{N}(\langle v_u, v_m \rangle, \sigma^2). \tag{3}$$

- We assume that both v_u and v_m are drawn from a normal distribution $v_u, v_m \sim \mathcal{N}(0, \lambda^2 \mathbf{1})$. Here $\mathbf{1}$ is the identity matrix.

3.2 Sequential Recommendation

Now assume that for any given user the ratings are observed in sequence, that is, we have sets

$$\{u; (m_1, y_1), \ldots, (m_{n_u}, y_{n_u})\} \tag{4}$$

where obviously the (m_i, y_i) pairs are different for different users. As before assume that $v_u, v_m \sim \mathcal{N}(0, \lambda^2 \mathbf{1})$. However, now rather than using an additive Gaussian model as in (3) we assume that y is based on $\langle v_u, v_m \rangle$ but with correlated noise. That is

$$(y_1, \ldots, y_{n_u}) \sim \mathcal{N}((\langle v_u, v_{m_1} \rangle, \ldots \langle v_u, v_{m_{n_u}} \rangle), \Sigma). \tag{5}$$

To simplify things we assume that Σ^{-1} is band diagonal and sparse with only one off-diagonal band on each side.

1. Write out the joint log-likelihood in terms of v_u, v_m and Σ and λ^2.
2. Assume that we fix v_u, v_m, λ^2. Assuming a conjugate prior on Σ how can you compute a maximum a posteriori estimate for Σ? Hint — you're basically estimating an anticorrelation correction between adjacent ratings (y_i, y_{i+1}).

Problem Set 4 — Dynamic Programming

4 Collaborative Filtering (20)

Your task is to implement a simple collaborative filtering algorithm and test it at scale. A simple dataset can be downloaded as part of the MovieLens collection http://grouplens.org/node/12. You will work with the MovieLens 10M collection, i.e. a collection of 10 million ratings. It can be downloaded from
http://www.grouplens.org/sites/www.grouplens.org/external_files/data/ml-10m.zip

4.1 Eachmovie Dataset

1. Download the file. Read `README.html` for a description of the content. The files of interest for this assignment are `movies.dat` which contains Movie IDs, titles, and a set of categories.
2. Discuss why a random 80/20 split as suggested by default in the preprocessing scripts is a bad idea. Explain why the validation errors might be systematically lower than the actual errors.
3. Generate a proper 80/20 split by using the timestamps of the ratings (look at the description of the file to find out the time format). That is, use the last 20% of all ratings as the test set and the first 80% as the training set.
4. Explain why the validation error on this split is likely to be systematically higher than the actual errors of a deployed system.
5. Discard all movies / ratings which only appear in the test set.
6. Randomly permute the order of the instances in the training set.

4.2 Stochastic Gradient Descent for Factorization

Next you are going to implement a very simple collaborative filtering algorithm. That is, it uses the following model for ratings:

$$f(u,m) = b_u + b_m + \langle v_u, v_m \rangle \tag{6}$$

and the loss function

$$\sum_{(u,m)\in \text{Ratings}} \frac{1}{2}\left(b_u + b_m + \langle v_u, v_m\rangle - y_{um}\right)^2 + \frac{\lambda}{2}\left[\sum_u \|v_u\|^2 + b_u^2 + \sum_m \|v_m\|^2 + b_m^2\right] \tag{7}$$

1. Implement a stochastic gradient descent algorithm on (7) using an $O(t^{-\frac{1}{2}})$ learning rate. Hint — you need to perform several iterations through the training set.
2. Adjust λ such as to obtain good generalization performance on the test set. Report the performance.

Hint — for the sake of simplicity you can just pick 50 dimensions. This should ensure that you don't run out of memory (only 400 bytes per user and movie respectively). Also, you only need to update one (movie, user) pair at a time.

4.3 Movie Attributes

As an extension we make use of the movie category attributes. In its simplest form this is achieved by estimating category-specific attribute vectors v_c and to replace the movie attribute vector v_m by one that has categories added.

$$v_m \leftarrow v_m + \sum_{c\in \text{categories}(m)} v_c \tag{8}$$

Here $\text{categories}(m)$ represents the set of categories specific to movie m.

Problem Set 4 — Dynamic Programming

---

1. Specify the objective function which includes v_c. You can use a least mean squares quadratic penalty on v_c, too.
2. Extend the stochastic gradient descent algorithm.
3. Evaluate the performance of the new algorithm relative to the one without categories. For which movies do you see the biggest changes?