EE 270: Large Scale Matrix Computation and Optimization **Winter 2020**

Lecture 7 — 28 January

Lecturer: Mert Pilanci *Scribe: Courtney Moran*

Least Squares Optimization and Random Projections

Least Squares Regression

- The goal is to predict the value of a continuous target variable, b
- For a set of data points, $(a_1, b_1), \ldots, (a_n, b_n)$ such that $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$, we want to find some x, x_0 such that $f(a) = x^T a + x_0$ best 'fits' the data
- To choose x, we define our loss function, $L(x, x_0)$, to be the sum of the squared error in each data point
- $L(x, x_0) = \frac{1}{n} \sum_{i=1}^{n} (b_i - a_i^T x - x_0)^2$
- This objective is to choose x, x_0 as to minimize the loss function
- We can add a regularization term to this loss function, which can prevent overfitting
- $L(x, x_0) = \frac{1}{n} \sum_{i=1}^{n} (b_i - a_i^T x - x_0)^2 + \lambda ||x||_2^2$
- We can simplify this problem by 'centering' the data and removing the x_0 variable from the problem. To do this we need to select an optimal x_0^*
- $x_0^* = \frac{1}{n} \sum_{i=1}^{n} (b_i - x^T a_i) = \tilde{b} - x^T \tilde{a}$ such that $\tilde{a} = \sum_i a_i$ and $\tilde{b} = \sum_i b_i$
- $L(x, x_0^*) = \frac{1}{n} \sum_{i=1}^{n} (b_i - \tilde{b} - x^T (a_i - \tilde{a}))^2 + \lambda ||x||_2^2$
- We have now simplified the problem to a minimization over x only
- $\min_x ||\tilde{A}x - \tilde{b}||_2^2 + n\lambda ||x||_2^2$
- We can find a closed from solution to this problem by simply taking the gradient of the loss function and setting it equal to 0
- $\frac{\partial}{\partial x} L(x, x_0^*) = 2\tilde{A}^T (\tilde{A}x^* - \tilde{b}) + 2n\lambda x^* = 0$
- $x^* = (\tilde{A}^T \tilde{A} + n\lambda I)^{-1} \tilde{A}^T \tilde{b}$
- Note, when $\lambda > 0$, this inverse term always exists. However, when $\lambda = 0$, $\tilde{A}$ must be full column rank for invertibility
- Without regularization, we can solve standard least squares problems ($\min_x ||Ax - b||_2^2$) with $x_{LS} = (A^T A)^{-1} A^T b$ (if A is full column rank)

Autoregressive Models

- $b[n] = a[n+1] \approx \sum_k x_k a[n-k]$
- For the AR(2) model, we have 2 non-zero filter coefficients:
- $a[n+1] = -x_0 a[n] - x_1 a[n-1]$
- ex. a sine wave can be exactly represented with the AR(2) model, where $a[n] = \sin(\alpha n)$
- These can be used to predict future values

Singular Value Decomposition (SVD)

- Every matrix has a singular value decomposition: $A = U\Sigma V^T$, such that U, V^T are orthonormal matrices, Σ is a diagonal matrix with non-increasing nonnegative entries
- Our standard least square solution is $x_{LS} = (A^T A)^{-1} A^T b$. We can substitute A with its SVD representation to find a solution that works for any A
- $x_{LS} = (V\Sigma U^T U \Sigma V^T)^{-1} V \Sigma U^T b = (V\Sigma^2 V^T)^{-1} V \Sigma U^T b = V\Sigma^{-2} V^T V \Sigma U^T b = V\Sigma^{-1} U^T b$
- We define the 'pseudo-inverse', $A^\dagger = V\Sigma^{-1} U^T$
- *all matrices have a pseudo-inverse of this form

Classical Methods for Least Squares

Direct Methods

- These tend to be slower ($O(nd^2)$) but more accurate ($err \approx 10^{-10}$)
- Cholesky Decomposition: form $A^T A$, decompose to $A^T A = R^T R$ such that R is upper triangular, solve $A^T A)^{-1} Ab = (R^T R)^{-1} A^T b$
- QR Decomposition: decompose to $A = QR$ and solve $Rx = Q^T b$
- SVD: $x_{LS} = V\Sigma^{-1} U^T b$ (this is the most stable method)
- one problem with using direct methods, is that they must store an entire copy of A in memory, so they cannot be used on matrices that are too large to be loaded into memory

Indirect Methods

- Faster ($O(\sqrt{\kappa} nd)$ such that $\kappa = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$ is the condition number of A), but less accurate ($err \approx 10^{-3}$)
- Gradient Descent with momentum (Chebyshev iteration)
- Conjugate Gradient *preferred method

- other iterative methods
- only the gradient $\nabla f(x) = A^T(Ax - b)$ needs to be loaded into memory, so these methods work for very large matrices

Faster Least Squares Optimization: Random Projection

- **Left Sketching:** form SA and SB such that $S \in \mathbb{R}^{m \times n}$ is a random projection matrix, and $m < n$
- we can now solve a smaller problem $\min_x ||SAx - Sb||_2^2$ using any classical method. For direct methods the complexity becomes $O(md^2)$
- To confirm that the expectation is equal to our original objective:
- $\mathbb{E}||SAx - Sb||_2^2 = \mathbb{E}(Ax - b)^T S^T S(Ax - b) = (Ax - b)^T(Ax - b) = ||Ax - b||_2^2$
- For indirect methods, the gradient is $(SA)^T(SAx - b)$. So if A is sparse, but S is dense, this could actually take longer since sparse matrices can be evaluated very quickly

Approximation Result

- Let $S \in \mathbb{R}^{n \times d}$ be a JL-embedding. Then our least squares problem becomes:
- $x_{LS} = \arg\min_x ||Ax - b||_2^2 \longrightarrow \tilde{x} = \arg\min_x ||SAx - Sb||_2^2$
- let $f(x) = ||Ax - b||_2^2$
- if $m \geq \text{constant} \times \frac{\text{rank}(A)}{\epsilon^2}$, then $f(x_{LS}) \leq f(\tilde{x}) \leq (1 + \epsilon^2) f(x_{LS})$
- We can also bound the prediction difference with $||A(x_{LS} - \tilde{x})||_2^2 \leq \epsilon^2$ with high probability

Gaussian Sketch

- let S be $\frac{1}{\sqrt{m}} \times$ i.i.d Gaussian with $\mathbb{E}[S^T S] = I$
- Given our solution $\tilde{x} = \arg\min_x ||SAx - Sb||_2^2$ can we say that it is unbiased? In other words, is $\mathbb{E}[\tilde{x}] = x_{LS}$?
- We can rewrite b as $b = Ax_{LS} + b^\perp$, then substituting this into our expression for $\tilde{x}$ we get:
- $\tilde{x} = (A^T S^T SA)^{-1} A^T S^T S(Ax_{LS} + b^\perp) = x_{LS} + (A^T S^T SA)^{-1} A^T S^T Sb^\perp$
- $\mathbb{E}[(A^T S^T SA)^{-1} A^T S^T Sb^\perp] = \mathbb{E}[(A^T S^T SA)^{-1} A^T S^T]\mathbb{E}[Sb^\perp] = 0$
- The previous equality holds since $\mathbb{E}[Sb^\perp] = 0$ because $b^\perp \perp \text{range}(A)$
- Thus, we see that $\mathbb{E}[\tilde{x}] = x_{LS}$ so our approximation is unbiased