TTIC/CMSC 31150 Mathematical Toolkit **Spring 2013**

Lecture 1: April 24, 2013

Madhur Tulsiani Scribe: Nikita Mishra

1 Applications of dimensionality reduction

In this lecture we talk about various useful applications of dimensionality reduction, specifically using JohnsonLindenstrauss lemma. JL lemma states that a small set of points in a high-dimensional space can be embedded into a space of much lower dimension in such a way that distances between the points are nearly preserved. In the previous lecture, we looked at one such linear embedding whose entries were sampled from iid Gaussian distribution. We look at the applications of dimensionality reduction in Singular value decomposition. And we will use the techniques of random projections in finding approximate solution to the graph-coloring problem.

2 Singular value decomposition

Definition 2.1 *The singular value decomposition (SVD) of an mn real or complex matrix A is a factorization of the form:*

$$A = \sum_{i=1}^{r} \sigma_i U_i V_i^T$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, $U_1 \cdots U_r \in R^m$, $V_1 \cdots V_r \in R^n$, $\forall i, AV_i = \sigma_i U_i$ & $U_i^T A = \sigma_i V_i^T$

2.1 Rank-k approximation

We note that the best $rank-k$ approximation to A which minimizes $\|A-M\|_F^2 = \sum_{i,j}(A_{ij} - M_{ij})^2$ is given by,

$$M = A_k = \sum_{i=1}^{k} \sigma_i U_i V_i^T = A(sum_{i=1}^{k} V_i V_i^T)$$

The time complexity of calculating k-best approximation is is $O(kmn^2)$. We now try to see where we can actually apply our dimensionality reduction technique and gain in terms of time complexity for this algorithm. Now, in order to compute A_k, we need to compute vectors $V_1, \cdots, V_k$. So, the idea is,

- Compute a matrix B, which is a dimension reduced version of A [V04]
- Let $Y_i, \cdots, Y_k$ be top k-right singular vectors of B
- Use $\tilde{p_k} = \sum_{i=1}^{k} Y_i Y_i^T$

- Then, $\tilde{A}_k = A \cdot \tilde{P}_k$

Lemma 2.2 *Let $A_k \in R^{mxn}$ be a matrix generated by the previous algorithm, then with high probability over choice of B,*

$$\|A - \tilde{A}_k\|_F^2 \leq \|A - A_k\|_F^2 + \varepsilon \|A_k\|_F^2$$

where B is given by $B_{lxn} = \frac{G^T A}{\sqrt(l)}$ and $G_{ij} \sim N(0,1)$, $G \in R^{mxl}$

Proof: Let us denote, $\varphi(u) = \frac{GU}{sqrt(l)}$. Then $B = [|\varphi(A^1) \dots \varphi(A^n)]$ can be thought of as a matrix in which every column is a projected column of A. Now, for any matrix M, let $e_1, \dots, e_n$ be some orthonormal vectors. Then,

$$\begin{aligned}
\|M\|_F^2 &= \sum_{i=1}^{m} \|M_i\|_2^2 \\
&= \sum_{i=1}^{m} \sum_{i=1}^{n} < M_i, e_j > \\
&= \sum_{j=1}^{n} \|Me_j\|^2
\end{aligned}$$

Let $V_1 \dots V_r, V_{r+1}, \dots V_n$ be orthonormal vectors.

$$\|A\|_F^2 = \textstyle\sum_{j=1}^{n} \|AV_j\|^2 = \sum_{j=1}^{r} \sigma_i^2$$

Clearly,

$$\|A_k\|_F^2 \leq \|A\|_F^2$$

$$\begin{aligned}
\|A - \tilde{A}_k\|_F^2 &= \sum_{j=1}^{n} \|(A - \tilde{A}_k)Y_j\|^2 \\
&= \sum_{j=1}^{k} \|AY_j\|^2 \\
&= \|A\|_F^2 - \sum_{j=1}^{n} \|AY_j\|^2
\end{aligned}$$

Also,

$$\|A - \tilde{A}_k\|_F^2 = \|A\|_F^2 - \|A_k\|_F^2$$

Hence,

$$\begin{aligned}\|A - \tilde{A}_k\|_F^2 - \|A - \tilde{A}_k\|_F^2 &= \|A_k\|_F^2 - \sum_{j=1}^{n} \|AY_j\|^2 \\ &= \sum_{j=1}^{k} \|AV_j\|^2 - \sum_{j=1}^{k} \|AY_j\|^2\end{aligned}$$

Here, the first term contains the singular values of A, whereas the second terms contains the singular values of B. We compare the terms with $\sum_{j=1}^{k} \|BY_j\|^2$, and show that both of them are close to it by JL lemma. Hence, they must be close to each other.

$$\begin{aligned}\sum_{j=1}^{k} \|BY_j\|^2 &= \sum_{j=1}^{k} \|\frac{G^T AY_j}{\sqrt(l)}\| \\ &\leq \sum_{j=1}^{k} \|AY_j\|^2 \cdot (1+\varepsilon)\end{aligned}$$

Similarly,

$$\begin{aligned}\sum_{j=1}^{k} \|BY_j\|^2 &\geq \sum_{j=1}^{k} \|BV_j\|^2 \\ &\geq \sum_{j=1}^{k} \|\frac{G^T AV_j}{\sqrt(l)}\| \\ &\geq \sum_{j=1}^{k} \|AV_j\|^2 \cdot (1-\varepsilon) \\ &\geq \sum_{j=1}^{k} \|A_k\|^2 \cdot (1-\varepsilon)\end{aligned}$$

Hence,

$$\begin{aligned}\|A - \tilde{A}_k\|_F^2 - \|A - \tilde{A}_k\|_F^2 &\leq \|A_k\|_F^2 - \frac{1-\varepsilon}{1+\varepsilon}\|A_k\|_F^2 \\ &\leq 2\varepsilon\|A_k\|_F^2\end{aligned}$$

■

3 Approximate coloring

Graph coloring is an assignment of labels (called "colors") to the vertices of a graph such that no two adjacent vertices share the same color; this is called a vertex coloring. Checking if a graph is k-colorable is NP-hard, for $k \geq 2$. Graph coloring is closely related to solving a k-cut problem, which in turn can have approximate solutions using Semi-definite programming relaxations. In this section, we look at one such example of SDP relaxation and finally how random projections can be applied on this relaxation to yield lower computational cost.

3.1 SDP relaxation

We consider a special case of 3-coloring. Assume that graph $G = (V, E)$ is a connected graph. For each $i \in V$, find a unit vector $v_i \in R^2$, such that the unit vectors are extreme points of triangle given in R^2, centered at origin.

$$\forall (i,j), < V_i, V_j >= -\frac{1}{2}$$
$$\forall i \in V, \|X_i\| = 1$$

The graph is 3-colorable iff this integer program is feasible. Note that, we have a constraint on the dimensions, hence it is not a semi-definite program. Therefore, the idea is to relax this semi-definite program. Instead of finding the minimum number of colors, it is sufficient to find large independent set. Since, all the elements in the independent set can have the same color.

Using a very simple technique, we can obtain a loose bound on the minimum number of required colors to color a graph. It is sufficient to find a large independent set of size $\frac{n}{k}$. Suppose, we initally have n uncolored nodes which are recursively colored by taking out $\frac{n}{k}$ independet sets. And we repeat the process t times until we color all the nodes. Thus, the minimum coloring is atmost $t - colors$.

$$n(1 - \tfrac{1}{k})^t \leq 1$$
$$\text{or } t \leq \frac{-\ln(n)}{\ln(1-\frac{1}{k})} \leq k\ln(n)$$

So, if $k = n^{\frac{1}{3}}$, we are done. In the next section, we try to improve this bound.

- Let us pick a random Gaussian variable $g_i \sim N(0, 1)$.
- $S = \{i \in V :< V_i, g >\geq l\}$
- Remove one end-point of every edge in S to obtain S', which is an independent set.

We try to analyze this algorithm. Intuitively, there is a very small chance that the projection of g will large on both the vectors whose inner product is -frac12. Hence, the number of edges in S must be small.

$$E(|S'|) = E(|S| - (\#\text{ of edges in S}))$$

Let X_i denote a indicator random variable such that,

$$X_i = \begin{cases} 1 & \text{if i} \in \text{S} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} E(|S|) &= \sum_{i \in V} E(|X_i|) \\ &\geq \sum_{i \in V} P(< V_i, g > \geq l) \end{aligned}$$

$$< V_i, g >= \textstyle\sum_{j=1}^{n} (V_i)_j g_j \sim N(0,1) \forall i$$

Let X_e denote a indicator random variable such that,

$$X_e = \begin{cases} 1 & \text{if e} \in \text{S} \\ 0 & \text{otherwise.} \end{cases}$$

$$E(\# \text{ of edges in S}) = E(\textstyle\sum_e X_e)$$

$$\begin{aligned} E(\# \text{ of edges in S}) &= E(\sum_e X_e) \\ &= \sum_{e \in (i,j)} E(X_e) \\ &= \sum_{e \in (i,j)} P(< U_i, g > \geq l \&\& < V_j, g > \geq l) \\ &\leq \sum_{e \in (i,j)} P(< U_i + V_j, g > \geq 2l) \end{aligned}$$

We note that,

$$< (u_i + V_j), g > \sim N(0, \|V_i + V_j\|^2)$$

$$\begin{aligned} \|V_i + V_j\|^2 &= \|V_i\|^2 + \|V_j\|^2 + 2 < V_i, V_j > \\ &= 2 + 2(-1/2) \\ &= 1 \text{Still a unit vector!!} \\ &= \sum_{e=(i,j)} F(2l) \end{aligned}$$

$$E(|s'|) = nF(l) - mF(2l)$$

where,

$$F(l) = \textstyle\int_l^\infty \frac{e^{-\frac{y^2}{2}}}{sqrt(2\pi)} dy$$

References

[V04] S. Vempala , "The random projection method", *Vol. 65. American Mathematical Soc.*, 2004