The Random Projection Method; chosen chapters from DIMACS vol.65 by Santosh S. Vempala

Edo Liberty

October 13, 2006

Talk topics

Random Projections in $\mathbb{R}^d$

Fast low rank approximation

Random projection in $\mathbb{Z}_2^d$

Nearest neighbor on the hypercube

Summary

Moving to section $\longrightarrow$

Random Projections in $\mathbb{R}^d$

Fast low rank approximation

Random projection in $\mathbb{Z}_2^d$

Nearest neighbor on the hypercube

Summary

Random Projections in $\mathbb{R}^d$

We start by giving a short proof of the Johnson-Lindenstrauss lemma due to P.Indyk and R.Motowani.

Lemma

A set of n points $\boldsymbol{u}_1 \ldots \boldsymbol{u}_n$ *in* $\mathbb{R}^d$ *can be projected down to* $\boldsymbol{v}_1 \ldots \boldsymbol{v}_n$ *in* $\mathbb{R}^k$ *such that all pairwise distances are preserved:*

$$(1-\epsilon)\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 \leq \|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 \leq (1+\epsilon)\|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2$$

if

$$k > \frac{9 \ln n}{\epsilon^2 - \epsilon^3}$$

Projecting 1 vector

We start with a vector $\mathbf{u} \in \mathbb{R}^d$ and a random $d \times k$ matrix s.t $R(i,j)$ are drawn *i.i.d* from $N(0,1)$.
We set

$$v = \frac{1}{\sqrt{k}} R^T \mathbf{u}$$

Then:

$$E(\|\mathbf{v}\|^2) = \|\mathbf{u}\|^2$$

And

$$(1-\epsilon)\|\mathbf{u}\|^2 \leq \|\mathbf{v}\|^2 \leq (1+\epsilon)\|\mathbf{u}\|^2$$

with probability $P \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}$

Length is preserved in expectancy

Since $\mathbf{v} = \frac{1}{\sqrt{k}} R^T \mathbf{u}$ we can calculate the expectancy $E(\|v\|^2)$.

$$
\begin{aligned}
E(\|\mathbf{v}\|^2) &= E\left(\sum_{i=1}^{k}\left(\sum_{j=1}^{d}\frac{1}{\sqrt{k}}R(i,j)\mathbf{u}(j)\right)^2\right) \\
&= \sum_{i=1}^{k}\frac{1}{k}E\left(\left(\sum_{j=1}^{d}R(i,j)\mathbf{u}(j)\right)^2\right) \\
&= \sum_{i=1}^{k}\frac{1}{k}\sum_{j=1}^{d}E\left(R(i,j)^2\right)E\left(\mathbf{u}(j)^2\right) \\
&= \sum_{i=1}^{k}\frac{1}{k}\sum_{j=1}^{d}\frac{\|\mathbf{u}\|^2}{d} \\
&= \|\mathbf{u}\|^2.
\end{aligned}
$$

Bounding the error probability

We define $x_j = \frac{1}{\|\mathbf{u}\|}\langle R_j, \mathbf{u}\rangle$ and

$$x = \frac{k\|\mathbf{v}\|^2}{\|\mathbf{u}\|^2} = \sum_{j=1}^{k} \frac{(R_j^T \mathbf{u})^2}{\|\mathbf{u}\|^2} = \sum_{j=1}^{k} x_j^2$$

We now turn to calculate the probability

$$P\left[\|\mathbf{v}\|^2 \geq (1+\epsilon)\|\mathbf{u}\|^2\right] = P\left[x > (1+\epsilon)k\right]$$

Bounding the error probability

$$
\begin{aligned}
P\left[\|\mathbf{v}\| \geq (1+\epsilon)\|\mathbf{u}\|^2\right] &= P\left[x \geq (1+\epsilon)k\right] \\
&= P\left[e^{\lambda x} \geq e^{\lambda(1+\epsilon)k}\right] \\
&\leq \frac{E(e^{\lambda x})}{e^{\lambda(1+\epsilon)k}} \\
&= \frac{\Pi_{j=1}^{k} E(e^{\lambda x_j^2})}{e^{\lambda(1+\epsilon)k}} \\
&= \left(\frac{E(e^{\lambda x_1^2})}{e^{\lambda(1+\epsilon)}}\right)^k
\end{aligned}
$$

Here we are faced with calculating $E(e^{\lambda x_1^2})$.

Expectation of $e^{\lambda x_1^2}$

Using the fact that x_1 itself is drawn from $N(0,1)$ We get that

$$
\begin{aligned}
E(e^{\lambda x_1^2}) &= \int_{-\infty}^{\infty} e^{\lambda t^2} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \\
&= \frac{1}{\sqrt{1-2\lambda}} \int_{-\infty}^{\infty} \frac{\sqrt{1-2\lambda}}{\sqrt{2\pi}} e^{-\frac{t^2}{2}(1-2\lambda)} dt \\
&= \frac{1}{\sqrt{1-2\lambda}} \quad \text{For } \lambda < 1/2.
\end{aligned}
$$

Bounding the error probability

Inserting $E(e^{\lambda X_1^2})$ into the probability equation we get:

$$P\left[\|\mathbf{v}\| \geq (1+\epsilon)\|\mathbf{u}\|^2\right] \leq \left(\frac{e^{-2(1+\epsilon)\lambda}}{1-2\lambda}\right)^{k/2}$$

Substituting for $\lambda = \frac{\epsilon}{2(1+\epsilon)}$ we get:

$$P\left[\|\mathbf{v}\| \geq (1+\epsilon)\|\mathbf{u}\|^2\right] \leq \left((1+\epsilon)e^{-\epsilon}\right)^{k/2}$$

Finally using that $1+\epsilon < e^{\epsilon-(\epsilon^2-\epsilon^3)/2}$ we obtain:

$$P\left[\|\mathbf{v}\| \geq (1+\epsilon)\|\mathbf{u}\|^2\right] \leq e^{-(\epsilon^2-\epsilon^3)k/4}$$

Bounding the error probability

$$
\begin{aligned}
P\left[\|\mathbf{v}\| \leq (1-\epsilon)\|\mathbf{u}\|^2\right] &= P\left[x \leq (1-\epsilon)k\right] \\
&= P\left[e^{-\lambda x} \geq e^{-\lambda(1-\epsilon)k}\right] \\
&= \left(\frac{E(e^{-\lambda x_1^2})}{e^{-\lambda(1-\epsilon)}}\right)^k \\
&\leq \left(\frac{e^{2(1-\epsilon)\lambda}}{1+2\lambda}\right)^{k/2} \quad , \lambda = \frac{\epsilon}{2(1-\epsilon)} \\
&\leq ((1-\epsilon)e^{\epsilon})^{k/2} \\
&\leq e^{-(\epsilon^2-\epsilon^3)k/4}
\end{aligned}
$$

Achieving for one point:

$$(1-\epsilon)\|\mathbf{u}\|^2 \leq \|\mathbf{v}\|^2 \leq (1+\epsilon)\|\mathbf{u}\|^2$$

with probability $P \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)\frac{k}{4}}$

Considering n points

In the case we have n points we have to preserve $O(n^2)$ distances. We can not fail with probability more then a constant, say, $P < 1/2$.

$$\begin{aligned} n^2 2e^{-(\epsilon^2-\epsilon^3)k/4} &< 1/2 \\ (\epsilon^2-\epsilon^3)k/4 &> 2\ln(2n) \\ k &> \frac{9\ln(n)}{\epsilon^2-\epsilon^3} \quad \text{For } n > 16. \end{aligned}$$

Finally, since the failure probability is smaller then $1/2$ we can repeat until success, in a constant number of times in expectancy.

References

The original proof of the possibility of embedding n points in dimension $O(\log(n))$ is due to W.B.Johnson and J.Lindenstrauss (1984) and used a random orthogonal matrix. P.Frankl and H.Meahara (1988) showed a simpler proof of this result.

The construction given here is due to P.Indyk and R.Motowani (STOC 1998). Although it is conceptually not different from previous results it is significantly easier to prove.

The same property was also proven independently by S.Dasgupta and A.Gupta (1999) and R.I.Arriaga and S.Vempala (FOCS 1999).

Moving to section $\longrightarrow$

Fast low rank approximation

We set to show how the results of the last section can be applied to accelerating low rank approximation of matrices.

An optimal low rank approximations can be easily computed using the SVD of A in $O(mn^2)$. Using random projections we show how to achive an ”almost optimal” low rank approximation in $O(mn\log(n))$.

We present a two step algorithm, suggested by Papadimitriou, Raghavan, Tamaki, and Vempala. First, we use k random projections to find a matrix B which is ”much smaller” then A, but still shares most of its (right) eigenspace. Then, we SVD B and project A on B’s k top eigenvectors.

Low rank approximation and SVD

A low rank approximation of an $m \times n$, $(m \geq n)$, matrix A is another matrix A_k such that:

1. The rank of A_k is at most k.
2. $\|A - A_k\|_{norm}$ is minimized.

It is well known that for both l_2, and the Frobenius norms

$$A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

Where the singular value decomposition (SVD) of A is:

$$A = USV^T = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

The accelerated two step algorithm

Let R be an $m \times \ell$ matrix such that $R(i,j)$ are drawn i.i.d from $N(0,1)$. Also we have that $\ell \geq \frac{c \log(n)}{\epsilon^2}$.

1. Compute $B = \frac{1}{\sqrt{\ell}} R^T A$.
2. Compute the SVD of B, $B = \sum_{i=1}^{\ell} \lambda_i \mathbf{a}_i \mathbf{b}_i^T$.

Return: $\widetilde{A}_k \leftarrow A \left(\sum_{i=1}^{k} \mathbf{b}_i \mathbf{b}_i^T \right)$.

B's singular values are not much smaller then A's

We show that

$$\sum_{i=1}^{k} \lambda_i^2 \geq (1-\epsilon) \sum_{i=1}^{k} \sigma_i^2$$

Where $A = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$, and $B = \frac{1}{\sqrt{\ell}} R^T A = \sum_{i=1}^{n} \lambda_i \mathbf{a}_i \mathbf{b}_i^T$.

$$\begin{aligned}
\sum_{i=1}^{k} \lambda_i^2 &\geq \sum_{i=1}^{k} \mathbf{v}_i^T B^T B \mathbf{v}_i \\
&= \sum_{i=1}^{k} \frac{1}{\ell} \mathbf{v}_i^T A^T R R^T A \mathbf{v}_i \\
&= \sum_{i=1}^{k} \sigma_i^2 \| \frac{1}{\sqrt{\ell}} R^T \mathbf{u}_i \|^2 \\
&\geq \sum_{i=1}^{k} (1-\epsilon) \sigma_i^2, \quad \text{With probability.}
\end{aligned}$$

Fast low rank approximation

Since A_k is the optimal solution we cannot hope to do better then $\|A - A_k\|_F^2$, Yet we show that we do not do much worse.

$$\|A - \widetilde{A}_k\|_F^2 \leq \|A - A_k\|_F^2 + 2\epsilon\|A_k\|_F^2$$

Reminder:

$$A = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad , \quad A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

$$B = \sum_{i=1}^{\ell} \lambda_i \mathbf{a}_i \mathbf{b}_i^T \quad , \quad \widetilde{A}_k = A \left(\sum_{i=1}^{k} \mathbf{b}_i \mathbf{b}_i^T \right) .$$

Fast low rank approximation

Since the b_is are orthogonal

$$\begin{aligned}
\|A - \widetilde{A}_k\|_F^2 &= \sum_{i=1}^{n} \|(A - \widetilde{A}_k)\mathbf{b}_i\|^2 \\
&= \sum_{i=1}^{n} \|A\mathbf{b}_i - A(\sum_{j=1}^{k} \mathbf{b}_j \mathbf{b}_j^T)\mathbf{b}_i\|^2 \\
&= \sum_{i=k+1}^{n} \|A\mathbf{b}_i\|^2 \\
&= \|A\|_F^2 - \sum_{i=1}^{k} \|A\mathbf{b}_i\|^2
\end{aligned}$$

on the other hand:

$$\|A - A_k\|_F^2 = \|A\|_F^2 - \|A_k\|_F^2.$$

Fast low rank approximation

From the last slide we have that:

$$\|A - \widetilde{A}_k\|_F^2 = \|A - A_k\|_F^2 + (\|A_k\|_F^2 - \sum_{i=1}^{k} \|A\mathbf{b}_i\|^2).$$

We now need to show that $\|A_k\|_F^2$ is not much larger then $\sum_{i=1}^{k} \|Ab_i\|^2$. We start by giving a lower bound on $\sum_{i=1}^{k} \|Ab_i\|^2$.

$$\begin{aligned}
\sum_{i=1}^{k} \lambda_i &= \sum_{i=1}^{k} \|B\mathbf{b}_i\|^2 \\
&= \sum_{i=1}^{k} \|\frac{1}{\sqrt{\ell}} R^T(A\mathbf{b}_i)\|^2 \\
&\leq (1+\epsilon) \sum_{i=1}^{k} \|Ab_i\|^2 \quad \text{With probability}
\end{aligned}$$

Fast low rank approximation

Reminder, we showed that:

$$\begin{aligned}(1+\epsilon)\sum_{i=1}^{k}\|Ab_i\|^2 &\geq \sum_{i=1}^{k}\lambda_i^2 \\ \sum_{i=1}^{k}\lambda_i^2 &\geq \sum_{i=1}^{k}(1-\epsilon)\sigma_i^2 = (1-\epsilon)\|A_k\|_F^2.\end{aligned}$$

Hence:

$$\begin{aligned}\sum_{i=1}^{k}\|Ab_i\|^2 &\geq \frac{1-\epsilon}{1+\epsilon}\|A_k\|_F^2 \\ &\geq (1-2\epsilon)\|A_k\|_F^2\end{aligned}$$

Finally:

$$\|A-\widetilde{A}_k\|_F^2 \leq \|A-A_k\|_F^2 + 2\epsilon\|A_k\|_F^2$$

With success probability:

$$P \geq 1 - 2ne^{-(\epsilon^2-\epsilon^3)k/4}.$$

Computational savings for full matrices

1. Computing the matrix B is $O(mn\ell)$.
2. Computing the SVD of B is $O(m\ell^2)$.
3. Where $\ell \geq \frac{c \log(n)}{\epsilon^2}$.

Hence the total running time is:

$$O\left(\frac{1}{\epsilon^2} mn \log(n)\right).$$

as compared to the straightforward SVD which is $O(mn^2)$.

More recent advances in low rank approximation

- A recent work by P.G.Martinsson, V.Rokhlin, and M.Tyghert (2006) give a tighter bound on the dimension of R, namely, that a rank k approximation can be achieved by projecting on $k + c$ vectors (c is a small constant). And a tighter bound on the error term. In case the matrix A exhibits some spectral decay.
- A.Frieze, R.Kannan, and, S. Vempala (1998) suggested a random sampling based algorithm that ϵ-approximates A_k with probability $1 - \delta$ and time complexity $Poly\left(k, \frac{1}{\epsilon}, \log(\frac{1}{\delta})\right)$.

Moving to section $\longrightarrow$

Random projection in $\mathbb{Z}_2^d$

In this section we extend the random projection idea to vectors is $\mathbb{Z}_2^d$ with distances measured in the ℓ_1 norm (The Hamming distance on the hypercube).

The method presented was suggested by Kushilevitz, Ostrovsky, and Rabini. Yet, the proof presented is a slightly simpler (S.Vempala).

We will show that multiplying (mod 2) a set of vectors V in $\mathbb{Z}_2^d$ by a random binary matrix R will project the points in V such that:

- ”Short” distance are potentially shrunk
- ”medium” distance are preserved within ϵ distortion
- ”long” distance are potentially stretched

Random Projection for the hypercube

Given a vector $\mathbf{v} \in \{0,1\}^d$ we project it down to $\mathbf{v}' = R^T\mathbf{v} \mod 2$. Where $R \in \{0,1\}^{d\times k}$ and $R(i,j) \in \{0,1\}$ with probabilities $(1-p, p)$ respectively.

Lemma

If we set $p = \frac{\epsilon^2}{\ell}$, $\boldsymbol{u}' = R^T\boldsymbol{u}$, *and* $\boldsymbol{v}' = R^T\boldsymbol{v}$. *Then with probability at least* $1 - e^{-C\epsilon^4 k}$ *(for some constant C)*

$$|\boldsymbol{u}-\boldsymbol{v}|_H < \frac{\ell}{4} \quad \Rightarrow \quad |\boldsymbol{u}'-\boldsymbol{v}'|_H < (1+\epsilon)kp\frac{\ell}{4}$$

$$\frac{\ell}{4} \le |\boldsymbol{u}-\boldsymbol{v}|_H \le \frac{\ell}{2\epsilon} \quad \Rightarrow \quad (1-\epsilon)kp \le \frac{|\boldsymbol{u}'-\boldsymbol{v}'|_H}{|\boldsymbol{u}-\boldsymbol{v}|_H} \le (1+\epsilon)kp$$

$$|\boldsymbol{u}-\boldsymbol{v}|_H > \frac{\ell}{2\epsilon} \quad \Rightarrow \quad |\boldsymbol{u}'-\boldsymbol{v}'|_H > (1-\epsilon)kp\frac{\ell}{2\epsilon}.$$

Expectancy of $|\mathbf{u}' - \mathbf{v}'|_H$

We start as before by calculating the expectancy of $|\mathbf{u}' - \mathbf{v}'|_H$.

$$\begin{aligned} E\left(|\mathbf{u}' - \mathbf{v}'|_H\right) &= E\left(|R^T(\mathbf{u} - \mathbf{v}) \mod 2|_H\right) \\ &= E\left(\sum_{j=1}^{k} R_j^T(\mathbf{u} - \mathbf{v}) \mod 2\right) \\ &= kP\left[R_1^T(\mathbf{u} - \mathbf{v}) = 1 \mod 2\right] \end{aligned}$$

Let us calculate the probability of the event $\mathcal{E}$:

$$P[\mathcal{E}] = P\left[R_1^T(\mathbf{u} - \mathbf{v}) = 1 \mod 2\right]$$

We can view the generation of R_1 as choosing coordinates with probability $2p$ and then setting them to *zero* with probability $1/2$. $\mathcal{E}$ will happen *half* of the times we choose at least one coordinate of $(\mathbf{u} - \mathbf{v})$. Therefor:

$$P[\mathcal{E}] = \frac{1}{2}\left(1 - (1 - 2p)^{|\mathbf{u}-\mathbf{v}|_H}\right)$$

Expectancy of $|\mathbf{u}' - \mathbf{v}'|_H$

Using the calculation from the last slide and the fact that:

$$pt - p^2t^2 \leq \frac{1}{2}\left(1 - (1-2p)^t\right) \leq pt$$

we get:

$$kp|\mathbf{u} - \mathbf{v}|_H - kp^2|\mathbf{u} - \mathbf{v}|_H^2 \leq E\left(|\mathbf{u}' - \mathbf{v}'|_H\right) \leq kp|\mathbf{u} - \mathbf{v}|_H$$

We also see that $|\mathbf{u}' - \mathbf{v}'|_H$ is a binomial variable $Bi(k, P[\mathcal{E}])$ with expectancy $kP[\mathcal{E}]$ which will help with bounding the probability of deviation from the expectancy.

Bounding the failure probability of the first property

$$|\mathbf{u}-\mathbf{v}|_H < \frac{\ell}{4} \Rightarrow |\mathbf{u}'-\mathbf{v}'|_H < (1+\epsilon)kp\frac{\ell}{4}$$

Using the Chernoff bound, the facts that $|\mathbf{u}-\mathbf{v}|_H \leq l/4$, and the fact that $P[\mathcal{E}] \leq p|u-v|_H$ we get:

$$\begin{aligned}
P\left[|\mathbf{u}'-\mathbf{v}'|_H > (1+\epsilon)\frac{kp\ell}{4}\right] &\leq P\left[|\mathbf{u}'-\mathbf{v}'|_H > kP[\mathcal{E}] + \epsilon\frac{kp\ell}{4}\right] \\
&= P\left[|\mathbf{u}'-\mathbf{v}'|_H > kP[\mathcal{E}](1+\frac{\epsilon p\ell}{4P[\mathcal{E}]})\right] \\
&\leq P\left[|\mathbf{u}'-\mathbf{v}'|_H > kP[\mathcal{E}](1+\frac{\epsilon\ell}{4|\mathbf{u}-\mathbf{v}|_H})\right] \\
&\leq e^{-k(\frac{\epsilon\ell}{4|\mathbf{u}-\mathbf{v}|_H})^2 P[\mathcal{E}]/3} \\
&\leq e^{-C\epsilon^4 k}
\end{aligned}$$

Bounding the failure probability of the second property

$$\frac{\ell}{4} \leq |\mathbf{u}-\mathbf{v}|_H \leq \frac{\ell}{2\epsilon} \Rightarrow (1-\epsilon)kp \leq \frac{|\mathbf{u}'-\mathbf{v}'|_H}{|\mathbf{u}-\mathbf{v}|_H} \leq (1+\epsilon)kp$$

Using the facts that $p|\mathbf{u}-\mathbf{v}|_H - p^2|\mathbf{u}-\mathbf{v}|_H^2 \leq P[\mathcal{E}]$ and that $\frac{\ell}{4} \leq |\mathbf{u}-\mathbf{v}|_H \leq \frac{\ell}{2\epsilon}$ we have that

$$(1-\epsilon)p|\mathbf{u}-\mathbf{v}|_H \leq P[\mathcal{E}] \leq p|\mathbf{u}-\mathbf{v}|_H$$

Using the above fact, and the Chernoff bound we get

$$\begin{aligned} &P\left[\left||\mathbf{u}'-\mathbf{v}'|_H - kp|\mathbf{u}-\mathbf{v}|_H\right| > \epsilon kp|\mathbf{u}-\mathbf{v}|_H\right] \\ &\leq P\left[\left||\mathbf{u}'-\mathbf{v}'|_H - kP[\mathcal{E}]\right| > \tfrac{\epsilon}{2}kP[\mathcal{E}]\right] \\ &\leq 2e^{-C\epsilon^4 k}. \end{aligned}$$

Bounding the failure probability of the third property

$$|\mathbf{u}-\mathbf{v}|_H > \frac{\ell}{2\epsilon} \Rightarrow |\mathbf{u}'-\mathbf{v}'|_H > (1-\epsilon)kp\frac{\ell}{2\epsilon}.$$

Using the monotonicity of $P[\mathcal{E}]$ w.r.t $|\mathbf{u}-\mathbf{v}|_H$, and the lower bound on $P[\mathcal{E}]$ we gain:

$$P[\mathcal{E}] \geq \frac{\epsilon}{2} + \frac{\epsilon^2}{4} \geq p\frac{\ell}{2\epsilon}$$

And therefore

$$\begin{aligned} P\left[|\mathbf{u}'-\mathbf{v}'|_H < (1-\epsilon)kp\frac{\ell}{2\epsilon}\right] &\leq P\left[|\mathbf{u}'-\mathbf{v}'|_H < (1-\epsilon)kP[\mathcal{E}]\right] \\ &\leq e^{-C\epsilon^3 k} \end{aligned}$$

Discussion

The dimensionality reduction $\mathbb{Z}_2^d \to \mathbb{Z}_2^k$ is much weaker in preserving lengths then $\mathbb{R}^d \to \mathbb{R}^k$. Yet, we gained the ability to distinguish between pairs of vectors whose distances are more (or less) then ℓ.
According to the properties if $|\mathbf{u} - \mathbf{v}|_H \leq l$ then:

$$|\mathbf{u}' - \mathbf{v}'|_H \leq (1+\epsilon)\epsilon^2 k$$

And for every such pair we have with high probability that:

$$|\mathbf{u} - \mathbf{v}|_H \leq \frac{1+\epsilon}{1-\epsilon}\ell$$

assuming $k = O(\frac{\log(n)}{\epsilon^4})$.
We will now see how this property is applicable to the approximate nearest neighbor problem on the hypercube.

Moving to section $\longrightarrow$

Exact nearest neighbor search

Exact nearest neighbor problem: Given a *database* V of n points in dimension d and a point $\mathbf{q}$; return the point $\mathbf{v}$ in V that is the closest to $\mathbf{q}$. (One is allowed preprocessing of V).
Trivial solution: Calculate distances from $\mathbf{q}$ to every point in V and choose the minimal one. This is not good enough!

- space use is $O(nd)$.
- query time is $O(nd)$.

Objective:

- Preprocessing space usage of $Poly(n, d)$
- Query time $O(Poly(\log(n), d))$.

Known algorithms for the exact nearest neighbor problem

For algorithms using space $O(nd)$, the best known query time is exponential either in d or in $\log(n)$ (e.g. the trivial solution)

S.Meiser (1993) gave the first algorithm with query time $Poly(d, log(n))$. But his algorithm uses storage of $O(n^d)$.

By relaxing the problem to the *Approximate* nearest neighbor problem we are able to considerably improve on this.

Approximate Nearest Neighbor

Approximate Nearest Neighbor problem: Given a *database* V of n points in dimension d and a point q, **and** $\epsilon > 0$.
If: $min_{\mathbf{v} \in V} \|\mathbf{q} - \mathbf{v}\| = r$
Return: a (possibly different) point $\mathbf{u} \in V$ such that $\|\mathbf{q} - \mathbf{u}\| \leq (1 + \epsilon)r$.

Approximate Nearest Neighbor

We reduce the problem to this one:
problem: Given a *database* V of n points in dimension d and a point $\mathbf{q}$, $\epsilon > 0$ **and** a distance r.
If: there is a point in $\mathbf{v} \in V$ such that $\|\mathbf{q} - \mathbf{v}\| \leq r$
Return: a (possibly different) point $\mathbf{u} \in V$ such that $\|\mathbf{q} - \mathbf{u}\| \leq (1 + \epsilon)r$.
Else: return failure.
We can binary search to find smallest r such that a query point has a neighbor and thus solving the nearest neighbor problem up to ϵ. The binary search will multiply the running time by a constant that does not depend on d or n.

Approximate Nearest Neighbor on the hypercube

We have seen how to reduce the dimension of points on the hypercube. We now want to try to apply it to our problem.
Idea 1:

1. Project the points in V into dimension $k = O(log(n)/\epsilon^2)$.
2. Assign to each of the 2^k points the closest $\mathbf{v}'$ to it.
3. Project a query point $\mathbf{q}$ into the same space and return the stored entry.

This would not work since long and short distances are potentially distorted.

Approximate Nearest Neighbor on the hypercube

We can distinguish between distances that are shorter or longer then ℓ if $p = \epsilon^2/\ell$. Moreover on the hypercube there are at most d different distances.

Idea 2:

1. Project the points in V into d different dimension k cubes, one for each distance.
2. Each point $\mathbf{x}'$ is assigned a point in V such that $|\mathbf{x}' - \mathbf{v}'| \leq (1+\epsilon)\epsilon^2 k$.
3. Otherwise no point is assigned.
4. Project a query point $\mathbf{q}$ to $\mathbf{q}'$. If there is a data point $\mathbf{v}$ assigned to $\mathbf{q}'$ conclude

$$|\mathbf{q} - \mathbf{v}| \leq \frac{1+\epsilon}{1-\epsilon}\ell$$

Otherwise conclude:

$$|\mathbf{q} - \mathbf{v}| \geq I \text{ for all points } \mathbf{v} \text{ in } V.$$

Approximate Nearest Neighbor on the hypercube

There is a problem with the last idea. Namely, there are 2^d possible query points. And with only one projection per distance

$$P_{failure} \leq \frac{1}{2^d} \Rightarrow k = \Omega(d)$$

Idea 3: Create M different independent random data structures $D_1 \ldots D_M$ for each distance. When a query is handled for a given distance, one of the data structures is used at random.

Approximate Nearest Neighbor on the hypercube

Claim: using the described method, for a set of n points in $\mathbf{Z}_2^d$ approximate nearest neighbor queries can be answered correctly, with probability $1 - \delta$ in time:

$$O\left(\frac{d}{\epsilon^4}\log(\frac{n}{\delta})\log(d)\right)$$

using a data structure of size:

$$O\left(d^2(Cn\log(d))^{c/\epsilon^4}\right).$$

Where C and c are constants.

Probability that the data structures are ”incorrect”

For a given distance the probability that D_i is built incorrectly is:

$$P_{incorrect}(D_i) \leq 2ne^{-C\epsilon^4 k}$$

thus the expected number $N_{D_{fail}}$ of D_is that are built incorrectly for a given distance is:

$$E(N_{D_{fail}}) \leq 2ne^{-C\epsilon^4 k}M$$

We can bound the probability of having ”too many” failures:

$$P_{fail} = P\left[N_{D_{fail}} > 2e \cdot E(N_{D_{fail}})\right] \leq 2^{4ene^{-C\epsilon^4 k}M}$$

Probability that the data structures are ”incorrect”

By setting:

$$k = \frac{\ln(8en\log(d)/\delta)}{C\epsilon^4}$$
$$M = (d + \log(\frac{2d}{\delta}))\frac{2\log(d)}{\delta}$$

we get:

$$P_{fail} \leq \frac{\delta}{2d2^d}$$

summing over all d distances and 2^d queries we get that with probability $1 - \delta/2$:

$$N_{D_{fail}} = (2e)(2ne^{-C\epsilon^4 k}) \leq \frac{\delta}{2\log(d)}M$$

Probability that a query search fails

Since at most

$$\frac{\delta}{2\log(d)}M$$

data structures are built incorrectly out of M, for each distance, the probability of using one of the incorrect data structures is smaller then:

$$\frac{\delta}{2\log(d)}$$

Since there are only $log(d)$ search steps for the distance r, the failure probability of all the steps is smaller then $\frac{\delta}{2}$.

Conclusion

- By adding the two failure probabilities we get with probability at least $1 - \delta$ that all data structures are built correctly and that all points q can be searched correctly.
- Query time: it takes time dk to project a point to the lower dimension and this is done at most $log(d)$ times.

$$\text{Query time} = d \log(d) k = O(\frac{d \log(d)}{\epsilon^4} \log(\frac{n}{\delta}))$$

- Space requirement: we have M data structures for each of the d lengths and each data structure is of size 2^k

$$\text{Space requirement} = Md2^k = O(d^2(Cn \log(d))^{c/\epsilon^4})$$

Discussion

The algorithm presented here was presented by E.Kushilevitz, R.Ostrovsky, and Y.Rabani (STOC 98). The same paper contains a probabilistic mapping of n points from $(\mathbb{R}^n, \ell_2)$ to $(\mathbb{Z}^{O(n)}, \ell_1)$, which automatically extend these results to the $(\mathbb{R}^n, \ell_2)$ case.

It improves on previous results by J.Klainberg (STOC 1997) and K.Clarkson (SIAM 1998). Clarkson's algorithm has exponential query time (in d). Klainberg suggested two algorithms. One with query time $Poly(d, \log(n))$ but with space requirement $O((n \log(d))^{2d})$. The other with space requirement of $O(dn)$ but with query time of $O(n + d \log^3 n)$.

Summary: What did we cover

- We showed that a random matrix can be used to project points in $\mathbb{R}^d$ to $\mathbb{R}^k$ in a length preserving way (with high probability). Such that $k \ll d$.
- We used this fact to accelerate Low rank approximations of $m \times n$ matrices from $O(mn^2)$ to $O(mn\log(n))$.
- We saw that a random binary matrix can reduce dimension from $\mathbb{Z}_2^d$ to $\mathbb{Z}_2^k$. Such that $k \ll d$ and still preserve some properties about pairwise distances.
- Finally, we used this fact to solve the approximate nearest neighbor problem. Achieving query time of $Poly(d, log(n))$. And storage space of $Poly(d, n)$.