Today:

- Integral operators
- Demolish NLA using LRA

$$\kappa(A) = \|A\| \, \|A^{-1}\|$$

1 Dense Matrices and Computation

Matvec: A slow algorithm

Matrix-vector multiplication: our first 'slow' algorithm.

$O(N^2)$ complexity.

$$\beta_i = \sum_{j=1}^{N} \underbrace{A_{ij}}\alpha_i$$

Assume A dense.

targets

sources

$$u(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\,\alpha_j$$

$$u(x) = \sum_{j=1}^{N} G(x, y_j)\,\alpha_j$$

$$u(x) = \int_\Omega G(x, y)\,\alpha(y)\,dy$$

Kernel

density functions

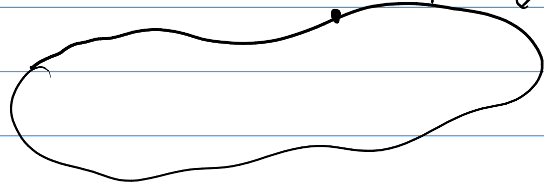
Kernel as in $N(A)$

not the same concept

$\frac{1}{|x-y|}$

$$u(x) = \int_\Gamma G(x,y)\,\underline{\alpha(y)}\,dy$$

Γ



for example:

$$V = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ \\ \\ 1 & x_n & x_n^2 & & x_n^m \end{pmatrix}$$

$$\begin{pmatrix} 1 & & & & x_n^m \\ \vdots & & & & \\ 1 & & & & \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \alpha_1 \cdot 1 + \alpha_2 \cdot x - \dots \\ \\ \\ \end{pmatrix}$$

Matrices and Point Interactions

$$A_{ij} = G(x_i, y_j)$$

Does that actually change anything?

Matrices and Point Interactions

$$A_{ij} = G(x_i, y_j)$$

Graphically, too:

Matrices and point interactions

$$\psi(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\varphi(y_j)$$

This *feels* different.

Point interaction matrices: Examples

What kind of matrices, then?

Integral 'Operators'

Why did we go through the trouble of rephrasing matvecs as

$$\psi(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\varphi(y_j)?$$

Cheaper Matvecs

$$\psi(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\varphi(y_j)$$

So what can we do to make evaluating this cheaper?

rank 1 $\quad G(x, y) = u(x) \cdot v(y)$

rank k $\quad G(x, y) = \sum_{i=1}^{k} u_i(x) \cdot v_i(y)$

$A = u v^T$

$A = \sum_{i=1}^{k} u_i v_i^T$

Fast Dense Matvecs

Consider

$$A_{ij} = u_i v_j,$$

let $\mathbf{u} = (u_i)$ and $\mathbf{v} = (v_j)$.

Can we compute $A\mathbf{x}$ quickly? (for a vector $\mathbf{x}$)

$$A x \leftarrow \text{rank } 1$$

$$= (u v^T) x$$

$$= u (v^T x) \leftarrow O(n)$$

rank k : $O(nk)$

Fast Dense Matvecs

$$A = \mathbf{u}_1 \mathbf{v}_1^T + \cdots + \mathbf{u}_k \mathbf{v}_k^T$$

Does this generalize?

Low-Rank Point Interaction Matrices

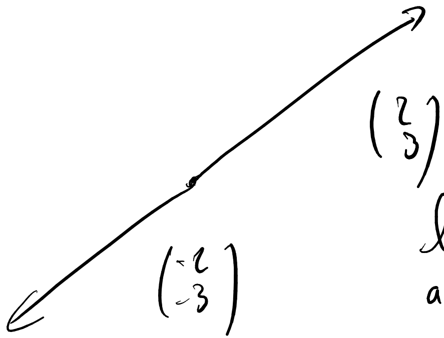
What would this:

$$\psi(x_i) = \sum_{j=1}^{N} G(x_i, y_j)\varphi(y_j)$$

look like for a low-rank matrix?

Numerical Rank

> What would a *numerical* generalization of 'rank' look like?



linear dependence and rank only make sense with a tolerance

Let $\varepsilon > 0$ be a tolerance

$$\| A - UV \| < \varepsilon$$

$A \in \mathbb{R}^{n \times n}$ $\quad U \in \mathbb{R}^{n \times k}$ $\quad V \in \mathbb{R}^{k \times n}$

$\Rightarrow$ A at most num. rank k

If $A \in \mathbb{R}^{m \times n}$

$$\operatorname{numrank}(A, \varepsilon) = \min\{k : \exists U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{k \times n}; \; \|A - UV\|_2 \le \varepsilon\}$$

Eckart-Young-Mirsky Theorem

Oddly enough, with the help of the SVD:

(see notes)

In the Frobenius norm:

$$\min_{\operatorname{rank}(B)=k} \| A - B \|_F = \| A - A_k \|_F = \sqrt{\sigma_{k+1}^2 + \dots + \sigma_n^2}$$

$$A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T$$

Constructing a tool

There is still a slight downside, though.

Representation

What does all this have to do with (right-)preconditioning?

2 Tools for Low-Rank Linear Algebra

Rephrasing Low-Rank Approximations

SVD answers low-rank-approximation ('LRA') question. But: too expensive.

First, rephrase the LRA problem:

$$A \approx UV$$ ← "factorization form" of LRA

$$A \approx QQ^T A$$ ← "projection form" of LRA

"ACA"

ONB

$$Q = \begin{pmatrix} | & | & | \\ q_1 & q_2 & q_3 \\ | & | & | \end{pmatrix}$$

$$Q Q^{\dagger} x$$

projection onto $q_1 \cdots q_3$

Using LRA bases

$A \approx QQ^TA$

> If we have an LRA basis Q, can we compute an SVD?

$$A = U \Sigma V^T$$

$$QQ^TA = QQ^TU\Sigma V^T$$

A: $n \times n$

Q: $n \times k$

B: $k \times n$

kN^2 1. $B = Q^TA$

kN^2 2. Compute SVD $B = \bar{U}\Sigma V^T$

kN^2 3. Set $U = Q\bar{U}$

$$U$$

$$Q\bar{U}\Sigma V^T$$

$Q^TA = \bar{U}\Sigma V^T$

$Q^TAV = \bar{U}\Sigma$

$Q^TAV\Sigma^{-1} = \bar{U}$

$$Q Q^T A V \Sigma^{-1} \Sigma V^T$$

$$= Q Q^T A \quad (+\text{ approx})$$

Finding an LRA basis

How would we *find* an LRA basis?

$$\|A - QQ^T A\|_2 \approx \min_{\text{rank}(X) \le k} \|A - X\|_2$$

Investigate the power method:

$A \in \mathbb{R}^{n \times n}$

Suppose $Ax_i = \lambda_i x_i$ $\quad i = 1 \dots n$

$x_i \neq 0$

$$x = \alpha_1 x_1 + \dots + \alpha_n x_n$$

$$A^n x = \lambda_1^n \alpha_1 x_1 + \lambda_2^n \alpha_2 x_2 + \dots + \lambda_n^n \alpha_n x_n$$

$$|\lambda_1| \geq |\lambda_2| \geq \dots |\lambda_n|$$

Giving up optimality

What problem should we actually solve then?

Recap: The Power Method

How did the power method work again?