# Random Projections for Dimensionality Reduction: Some Theory and Applications

Robert J. Durrant

University of Waikato

bobd@waikato.ac.nz

www.stats.waikato.ac.nz/~bobd

Télécom ParisTech, Tuesday 12th September 2017

# Outline

# Motivation - Dimensionality Curse

The 'curse of dimensionality': A collection of pervasive, and often counterintuitive, issues associated with working with high-dimensional data.
Two typical problems:

- Very high dimensional data (dimensionality $d \in \mathcal{O}(1000)$) and very many observations (sample size $N \in \mathcal{O}(1000)$): Computational (time and space complexity) issues.
- Very high dimensional data (dimensionality $d \in \mathcal{O}(1000)$) and hardly any observations (sample size $N \in \mathcal{O}(10)$): Inference a hard problem. Bogus interactions between features.

# Curse of Dimensionality

**Comment**: What constitutes high-dimensional depends on the problem setting, but data vectors with dimensionality in the thousands very common in practice (e.g. medical images, gene activation arrays, text, time series, ...).

Issues can start to show up when data dimensionality in the tens!

We will simply say that the observations, $\mathcal{T}$, are $d$-dimensional and there are $N$ of them: $\mathcal{T} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$ and we will assume that, for whatever reason, $d$ is too large.

# Mitigating the Curse of Dimensionality

An obvious solution: Dimensionality $d$ is too large, so reduce $d$ to $k \ll d$.

How?
Dozens of methods: PCA, Factor Analysis, Projection Pursuit, ICA, Random Projection ...

We will be focusing on Random Projection, motivated (at first) by the following important result:

# Johnson-Lindenstrauss Lemma

The JLL is the following rather surprising fact [DG02, Ach03]:

## Theorem (W.B.Johnson and J.Lindenstrauss, 1984)

*Let $\epsilon \in (0,1)$. Let $N, k \in \mathbb{N}$ such that $k \geqslant C\epsilon^{-2} \log N$, for a large enough absolute constant $C$. Let $V \subseteq \mathbb{R}^d$ be a set of $N$ points. Then there exists a linear mapping $R : \mathbb{R}^d \to \mathbb{R}^k$, such that for all $u, v \in V$:*

$$(1-\epsilon)\|u-v\|_2^2 \leqslant \|Ru-Rv\|_2^2 \leqslant (1+\epsilon)\|u-v\|_2^2$$

- Dot products are also approximately preserved by $R$ since if JLL holds then: $u^T v - \epsilon\|u\|\|v\| \leqslant (Ru)^T Rv \leqslant u^T v + \epsilon\|u\|\|v\|$. (Proof: parallelogram law).
- Scale of $k$ is sharp even for *adaptive* linear $R$ (e.g. 'thin' PCA): $\forall N, \exists V$ s.t. $k \in \Omega(\epsilon^{-2} \log N)$ is required [LN14, LN16].
- We shall prove shortly that with high probability *random projection* (that is left-multiplying data with a wide, shallow, random matrix) implements a suitable linear $R$.

# Jargon

‘With high probability’ (w.h.p) means with a probability as close to 1 as we choose to make it.

‘Almost surely’ (a.s.) or ‘with probability 1’ (w.p. 1) means so likely we can pretend it always happens.

‘With probability 0’ (w.p. 0) means so unlikely we can pretend it never happens.

# Intuition

Geometry of data gets perturbed by random projection, but not too much:

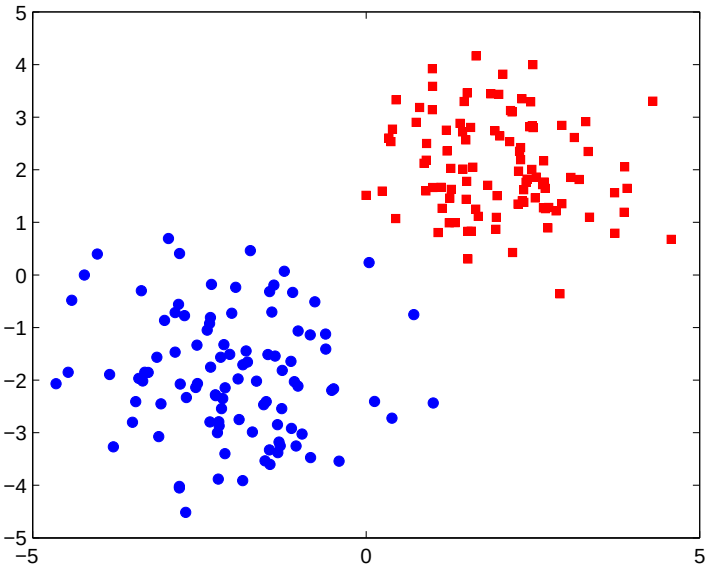



Figure: Original data

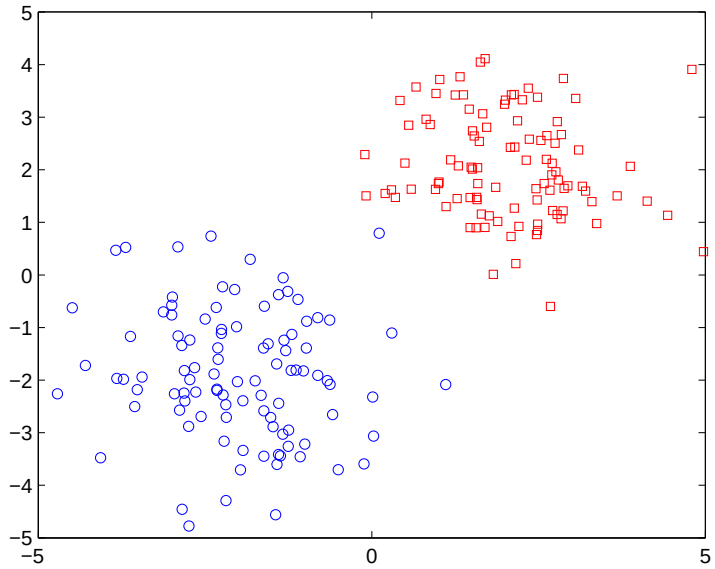



Figure: RP data (schematic)

# Intuition

Geometry of data gets perturbed by random projection, but not too much:

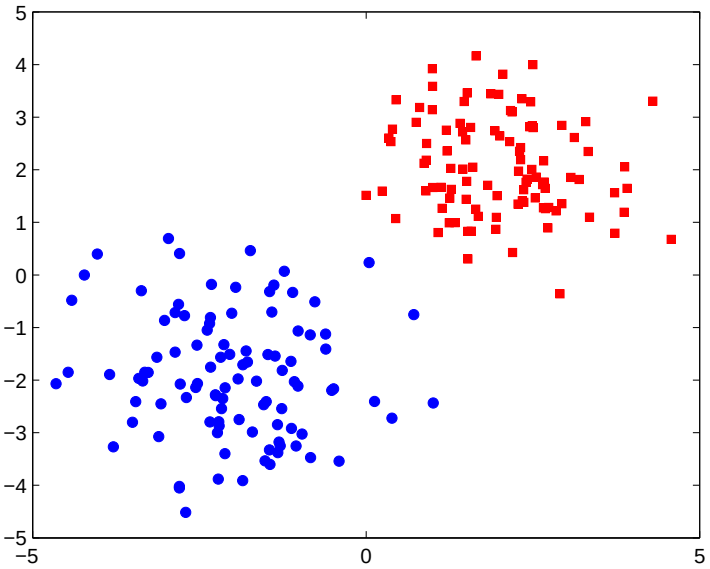



Figure: Original data

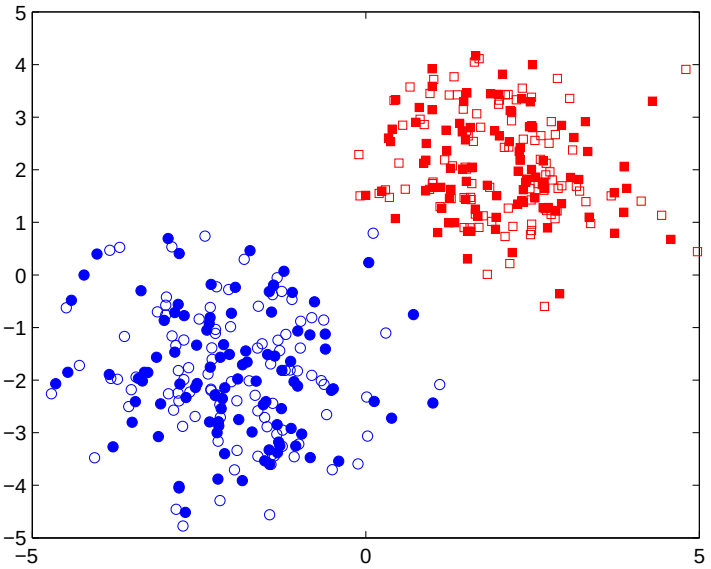



Figure: RP data & Original data

# Applications

Random projections have been used for:

- Classification. e.g. [BM01, FM03, GBN05, SR09, CJS09, RR08, DK15, CS15, HWB07, BD09]
- Clustering and Density estimation. e.g. [IM98, AC06, FB03, Das99, KMV12, AV09]
- Other related applications: structure-adaptive kd-trees [DF08], low-rank matrix approximation [Rec11, Sar06], sparse signal reconstruction (compressed sensing) [Don06, CT06], matrix completion [CT10], data stream computations [AMS96], heuristic optimization [KBD16].

# What is Random Projection? (1)

Canonical RP:

- Construct a (wide, flat) matrix $R \in \mathcal{M}_{k \times d}$ by picking the entries from a univariate Gaussian $\mathcal{N}(0, \sigma^2)$.
- Orthonormalize the rows of $R$, e.g. set $R' = (RR^T)^{-1/2}R$.
- To project a point $v \in \mathbb{R}^d$, pre-multiply the vector $v$ with RP matrix $R'$. Then $v \mapsto R'v \in R'(\mathbb{R}^d) \equiv \mathbb{R}^k$ is the projection of the $d$-dimensional data into a random $k$-dimensional projection space.

# Comment (1)

If $d$ is very large we can drop the orthonormalization in practice - the rows of $R$ will be nearly orthogonal to each other and all nearly the same length.

For example, for Gaussian ($\mathcal{N}(0, \sigma^2)$) $R$ we have [DK12]:

$$\Pr\left\{(1-\epsilon)d\sigma^2 \leqslant \|R_i\|_2^2 \leqslant (1+\epsilon)d\sigma^2\right\} \geqslant 1-\delta,\ \forall \epsilon \in (0,1]$$

where $R_i$ denotes the $i$-th row of $R$ and
$\delta = \exp(-(\sqrt{1+\epsilon}-1)^2 d/2) + \exp(-(\sqrt{1-\epsilon}-1)^2 d/2)$.

Similarly [Led01]:

$$\Pr\{|R_i^T R_j|/d\sigma^2 \leqslant \epsilon\} \geqslant 1 - 2\exp(-\epsilon^2 d/2),\ \forall i \neq j.$$

# Concentration in norms of rows of $R$

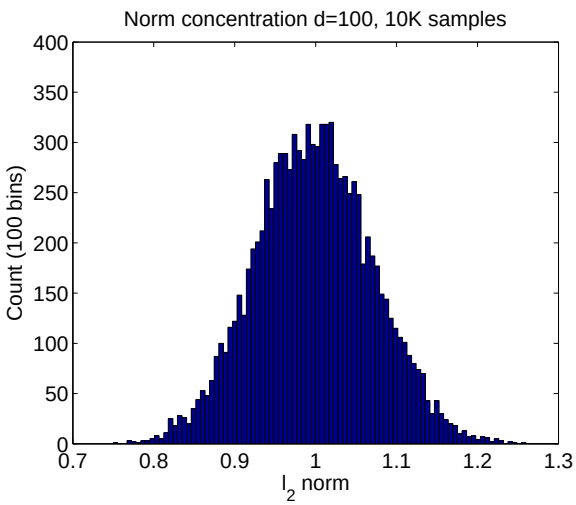



Figure: $d = 100$ norm concentration

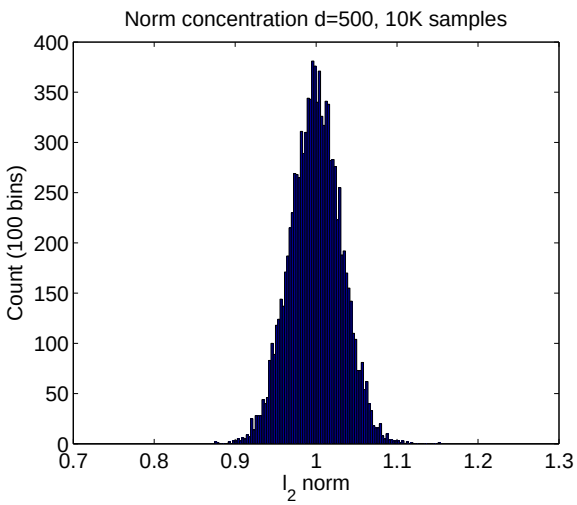



Figure: $d = 500$ norm concentration

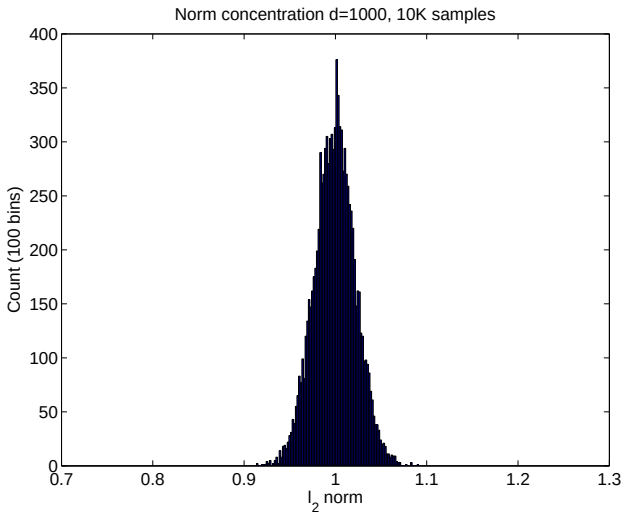



Figure: $d = 1000$ norm concentration

# Near-orthogonality of rows of $R$

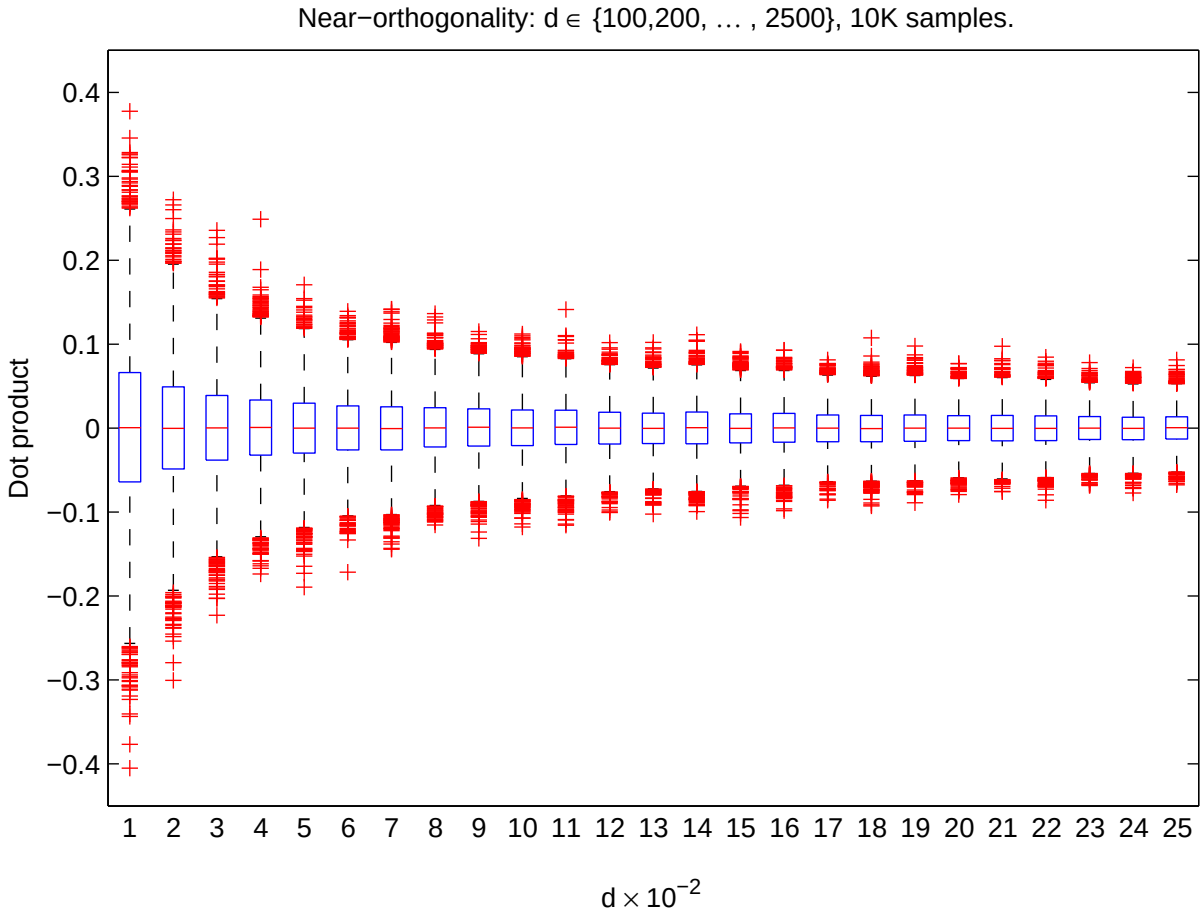



Figure: Normalized dot product is concentrated about zero, $d \in \{100, 200, \ldots, 2500\}$

# Why Random Projection?

- Linear.
- Cheap.
- Universal – JLL holds w.h.p for any fixed finite point set.
- Oblivious to data distribution.
- Target dimension doesn't depend on data dimensionality (for JLL).
- Interpretable - approximates an isometry (when $d$ is large).
- Tractable to analysis.

# Proof of JLL (1)

We will prove the following randomized version of the JLL, and then show that this implies the original theorem:

## Theorem

*Let $\epsilon \in (0,1)$. Let $k \in \mathbb{N}$ such that $k \geqslant C\epsilon^{-2} \log \delta^{-1}$, for a large enough absolute constant $C$. Then there is a random linear mapping $P : \mathbb{R}^d \to \mathbb{R}^k$, such that for any unit vector $x \in \mathbb{R}^d$:*

$$Pr\left\{(1-\epsilon) \leqslant \|Px\|^2 \leqslant (1+\epsilon)\right\} \geqslant 1-\delta$$

- No loss to take $\|x\| = 1$, since $P$ is linear.
- Note that this mapping is universal and the projected dimension $k$ depends only on $\epsilon$ and $\delta$.
- Lower bound [LN14, LN16] $k \in \Omega(\epsilon^{-2} \log \delta^{-1})$.

# Proof of JLL (2)

Consider the following simple mapping:

$$Px := \frac{1}{\sqrt{k}} Rx$$

where $R \in \mathcal{M}_{k \times d}$ with entries $R_{ij} \overset{i.i.d}{\sim} \mathcal{N}(0, 1)$.

Let $x \in \mathbb{R}^d$ be an arbitrary unit vector.
We are interested in the quantity:

$$\|Px\|^2 = \left\| \frac{1}{\sqrt{k}} Rx \right\|^2 := \left\| \frac{1}{\sqrt{k}} (Y_1, Y_2, \ldots, Y_k) \right\|^2 = \frac{1}{k} \sum_{i=1}^{k} Y_i^2 =: Z$$

where $Y_i = \sum_{j=1}^{d} R_{ij} x_j$.

# Proof of JLL (3)

Recall that if $W_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ and the $W_i$ are independent, then $\sum_i W_i \sim \mathcal{N}\left(\sum_i \mu_i, \sum_i \sigma_i^2\right)$. Hence, in our setting, we have:

$$Y_i = \sum_{j=1}^{d} R_{ij} x_j \sim \mathcal{N}\left(\sum_{j=1}^{d} \mathrm{E}[R_{ij} x_j], \sum_{j=1}^{d} \mathrm{Var}(R_{ij} x_j)\right) \equiv \mathcal{N}\left(0, \sum_{j=1}^{d} x_j^2\right)$$

and since $\|x\|^2 = \sum_{j=1}^{d} x_j^2 = 1$ we therefore have:

$$Y_i \sim \mathcal{N}(0,1), \ \forall i \in \{1, 2, \ldots, k\}$$

it follows that each of the $Y_i$ are standard normal RVs and therefore $kZ = \sum_{i=1}^{k} Y_i^2$ is $\chi_k^2$ distributed.

Now we complete the proof using a standard Chernoff-bounding approach.

# Proof of JLL (4)

$$
\begin{aligned}
\Pr\{Z > 1+\epsilon\} &= \Pr\{\exp(tkZ) > \exp(tk(1+\epsilon))\},\ \forall t > 0 \\
\text{Markov ineq.} &\leqslant \mathrm{E}\left[\exp(tkZ)\right] / \exp(tk(1+\epsilon)), \\
Y_i \text{ indep.} &= \prod_{i=1}^{k} \mathrm{E}\left[\exp(tY_i^2)\right] / \exp(tk(1+\epsilon)), \\
\text{mgf of } \chi^2_k &= \left[\exp(t)\sqrt{1-2t}\right]^{-k} \exp(-kt\epsilon), \forall t < 1/2 \\
\text{next slide} &\leqslant \exp\left(kt^2/(1-2t) - kt\epsilon\right), \\
&\leqslant e^{-\epsilon^2 k/8}, \text{ taking } t = \epsilon/4 < 1/2.
\end{aligned}
$$

$\Pr\{Z < 1-\epsilon\} = \Pr\{-Z > \epsilon - 1\}$ is tackled in a similar way and we obtain same bound. Taking RHS as $\delta/2$ and applying union bound completes the proof (for single $x$).

# Estimating $(e^t\sqrt{1-2t})^{-1}$

$$\begin{aligned}
\left(e^t\sqrt{1-2t}\right)^{-1} &= \exp\left(-t-\frac{1}{2}\log(1-2t)\right), \\
\text{Maclaurin S. for } \log(1-x) &= \exp\left(-t-\frac{1}{2}\left(-2t-\frac{(2t)^2}{2}-\ldots\right)\right), \\
&= \exp\left(\frac{(2t)^2}{4}+\frac{(2t)^3}{6}+\ldots\right), \\
&\leqslant \exp\left(t^2\left(1+2t+(2t)^2\ldots\right)\right), \\
&= \exp\left(t^2/\left(1-2t\right)\right) \text{ since } 0<2t<1
\end{aligned}$$

# Randomized JLL implies Deterministic JLL

- Solving $\delta = 2\exp(-\epsilon^2 k/8)$ for $k$ we obtain $k = 8\epsilon^{-2}\log 2\delta^{-1}$. i.e. $k \in \mathcal{O}\left(\epsilon^{-2}\log\delta^{-1}\right)$.
- Let $V = \{x_1, x_2, \ldots, x_N\}$ an arbitrary set of $N$ points in $\mathbb{R}^d$ and set $\delta = 1/2N^2$, then $k \in \mathcal{O}\left(\epsilon^{-2}\log N\right)$.
- Applying union bound to the randomized JLL proof for all $\binom{N}{2}$ possible interpoint distances, for $N$ points we see a random JLL embedding of $V$ into $k$ dimensions succeeds with probability at least $1 - \binom{N}{2}\frac{1}{N^2} > \frac{1}{2}$.
- We succeed with positive probability for arbitrary $V$. Hence we conclude that, for any set of $N$ points, there exists linear $P : \mathbb{R}^d \to \mathbb{R}^k$ such that:

$$(1-\epsilon)\|x_i - x_j\|^2 \leqslant \|Px_i - Px_j\|^2 \leqslant (1+\epsilon)\|x_i - x_j\|^2$$

which is the (deterministic) JLL.

# From Point Sets to Manifolds

From JLL we obtain high-probability guarantees that for a suitably large $k$, independently of the data dimension, random projection approximately preserves Euclidean geometry of a finite point set. In particular Euclidean norms and dot products approximately preserved w.h.p.

JLL approach can be extended to (compact) Riemannian manifolds: 'Manifold JLL' [BW09].

**Key idea**: Preserve $\frac{\epsilon}{2}$-covering of smooth manifold instead of geometry of data points. Replace $N$ in JLL with corresponding covering number $M$ and take $k \in \mathcal{O}\left(\epsilon^{-2} \log M\right)$.

**Wrinkle**: Absent additional low-dimensional structure in data, $M$ is typically $\mathcal{O}\left(2^d\right)$ implying trivial guarantee $k = d$. In practice RP works better than this theory predicts.

# Applications of Random Projection

JLL implies that if $d$ is large, with a suitable choice of $k$, we can construct an '$\epsilon$-approximate' version of *any* algorithm which depends only on Euclidean norms and dot products of the data, but in a much lower-dimensional space. This includes:

- Nearest-neighbour algorithms.
- Clustering algorithms.
- Margin-based classifiers.
- Least-squares regressors.

That is, we trade off some accuracy (perhaps) for reduced algorithmic time and space complexity.

However the matrix-matrix multiplication is still costly when $d$ or $N$ very large – e.g. consider a dataset comprising many high-resolution images.
Thus much interest in speeding up this part of process.

# Comment (2)

In the proof of the randomized JLL the only properties we used which are specific to the Gaussian distribution were:

1. Closure under additivity.
2. Bounding squared Gaussian RV using mgf of $\chi^2$.

In particular, bounding via the mgf of $\chi^2$ gave us exponential concentration about mean norm.
Can do similar for matrices with zero-mean *sub-Gaussian* entries also, i.e. those distributions whose tails decay no slower than a Gaussian $\Longrightarrow$ similar theory for sub-Gaussian RP matrices too!
One method for getting around issue of dense matrix multiplication in dimensionality-reduction step (same time complexity, better constant).

# What is Random Projection? (2)

Different types of RP matrix easy to construct - take entries i.i.d from *nearly any* zero-mean subgaussian distribution. All behave in much the same way.

Popular variations [Ach03, AC06, Mat08]:

The entries $R_{ij}$ can be:

$$R_{ij} = \begin{cases} +1 & \text{w.p. } 1/2, \\ -1 & \text{w.p. } 1/2. \end{cases} \qquad R_{ij} = \begin{cases} \mathcal{N}(0, 1/q) & \text{w.p. } q, \\ 0 & \text{w.p. } 1-q. \end{cases}$$

$$R_{ij} = \begin{cases} +1 & \text{w.p. } 1/6, \\ -1 & \text{w.p. } 1/6, \\ 0 & \text{w.p. } 2/3. \end{cases} \qquad R_{ij} = \begin{cases} +1 & \text{w.p. } q, \\ -1 & \text{w.p. } q, \\ 0 & \text{w.p. } 1-2q. \end{cases}$$

For the RH examples, taking $q$ too small gives high distortion of sparse vectors [Mat08]. [AC06] get around this by using a random orthogonal matrix to ensure w.h.p all data vectors are dense.

However even sparse$\times$dense matrix-matrix multiplication may be too slow. Can we do better?

# Faster Projections for Smooth Data

- Proof technique for JLL is essentially to show that (squared) norms of projected vectors are close to their expected value w.h.p., then recover correct scale using appropriate constant.
- Turning observation of [Mat08] around - plausible that for 'smooth enough' data even *very* sparse projection could still imply JLL-type guarantees.
- In particular can we obtain JLL for random subspace ('RS') [Ho98] - choosing $k$ features from $d$ uniformly at random without replacement?
- Comment: Clearly hopeless to attempt this for very sparse vectors e.g. consider the canonical basis vectors. On the other hand $k = 1$ will do if all features have identical absolute values.
- Q: Where is the breakdown point – i.e. given dataset $V$ of size $N$, at which value of $k$? How to characterise 'smoothness'? Can suitably 'smooth' data be found in the wild?

# Why is RS particularly interesting?

- Very widely-used randomized feature-selection scheme, e.g. basis for random forests, but theory for it is sparse.
- No matrix multiplication involved – time complexity linear in dimension $d \Longrightarrow$ faster approximation algorithms.
- Link to 'dropout' in deep neural networks – dropout essentially RS applied to internal nodes of network $\Longrightarrow$ potential speedup of training these huge models (e.g. conjecture back prop only on a very small random sample of nodes may work well).
- Potential for new theory:
  - Explaining effect of dropout.
  - For RS ensembles, e.g. explaining experimental findings in [DK15].
  - On learning from streaming data (streaming time series frequently subsampled in practice).
  - Compressive sensing, e.g. subsampling audio files in time domain.
  - Geometric interpretations for sampling theory.
- For many problems desirable (or essential) to work with original features.

# JLL for Random Subspace (1)

WLOG work in $\mathbb{R}^d$ and instantiate RS as a projection $P$ on to subspace spanned by $k$ coordinate directions.

## Theorem (Basic Hoeffding Bound [LD17])

*Let $\mathcal{T}_N := \{X_i \in \mathbb{R}^d\}_{i=1}^N$ be a set of $N$ points in $\mathbb{R}^d$ satisfying, $\forall i \in \{1, 2, \ldots, N\}$, $\|X_i^2\|_\infty \leqslant \frac{c}{d}\|X_i\|_2^2$ where $c \in \mathbb{R}_+$ is a constant $1 \leqslant c \leqslant d$. Let $\epsilon, \delta \in (0, 1]$, and let $k \geqslant \frac{c^2}{2\epsilon^2} \ln \frac{N^2}{\delta}$ be an integer. Let $P$ be a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$. Then with probability at least $1 - \delta$ over the random draws of $P$ we have, for every $i, j \in \{1, 2, \ldots, N\}$:*

$$(1 - \epsilon)\|X_i - X_j\|_2^2 \leqslant \frac{d}{k}\|P(X_i - X_j)\|_2^2 \leqslant (1 + \epsilon)\|X_i - X_j\|_2^2$$

# JLL for Random Subspace (2)

## Theorem (Serfling Bound [LD17])

*Let $\mathcal{T}_N, c, \epsilon, \delta$ as before. Define $f_k := (k-1)/d$ and let $k$ such that $k/(1-f_k) \geqslant \frac{c^2}{2\epsilon^2} \ln \frac{N^2}{\delta}$ be an integer. Let $P$ be a random subspace projection from $\mathbb{R}^d \mapsto \mathbb{R}^k$. Then with probability at least $1-\delta$ over the random draws of $P$ we have, for every $i, j \in \{1, 2, \ldots, N\}$:*

$$(1-\epsilon)\|X_i - X_j\|_2^2 \leqslant \frac{d}{k}\|P(X_i - X_j)\|_2^2 \leqslant (1+\epsilon)\|X_i - X_j\|_2^2$$

Comment: Always sharper than Theorem 3, but brings (typically unwanted, though benign) dependence on $d$ in choice of $k$.

# Proof Sketch

- View each vector as a finite population of size $d$. RS is then a simple random sample of size $k$ drawn without replacement from it.
- Sampling distribution of the mean from a finite population without replacement has smaller variance than sampling with replacement. . .
- . . .thus Hoeffding bound for independent sampling with replacement is also bound for sampling without replacement.
- Standard Hoeffding bound argument, except for data-dependent constant $c$ is additionally chosen to kill the dependency on $d$ (and implicitly enforces 'smoothness').
- Finer-grained approach uses Serfling bound, which exploits martingale structure in sampling scheme. Similar proof structure.

# JLL for Random Subspace (3)

## Corollary (to either bound)

*Under the conditions of Theorem 3 or 4 respectively, for any* $\epsilon, \delta \in (0,1)$*, with probability at least* $1 - 2\delta$ *over the random draws of* $P$ *we have:*

$$\left(X_i^T X_j - \epsilon \|X_i\| \|X_j\|\right) \leqslant \frac{d}{k}(PX_i)^T(PX_j) \leqslant \left(X_i^T X_j + \epsilon \|X_i\| \|X_j\|\right)$$

# Empirical Corroboration:

- We corroborate theory and compare RS projection with two RP variants as well as to principal components analysis (PCA) to see that in practice – given a suitable choice of $k$ – RS works as well as these alternatives.
- Data are 23 grayscale images from the USC-SIPI natural image dataset. From each image we sampled one hundred $50 \times 50$ squares by choosing their top left corner at random, and reshaped to give a vector in $\mathbb{R}^{2500}$.

| Name | Description | Image Size | $c$ |
|---|---|---|---|
| 5.1.09 | Moon Surface | 256x256 | 3.50 |
| 5.1.10 | Aerial | 256x256 | 2.44 |
| 5.1.11 | Airplane | 256x256 | 7.92 |
| 5.1.12 | Clock | 256x256 | 5.03 |
| 5.1.14 | Chemical plant | 256x256 | 2.92 |
| ⋮ | ⋮ | ⋮ | ⋮ |

# Representative Outcomes:

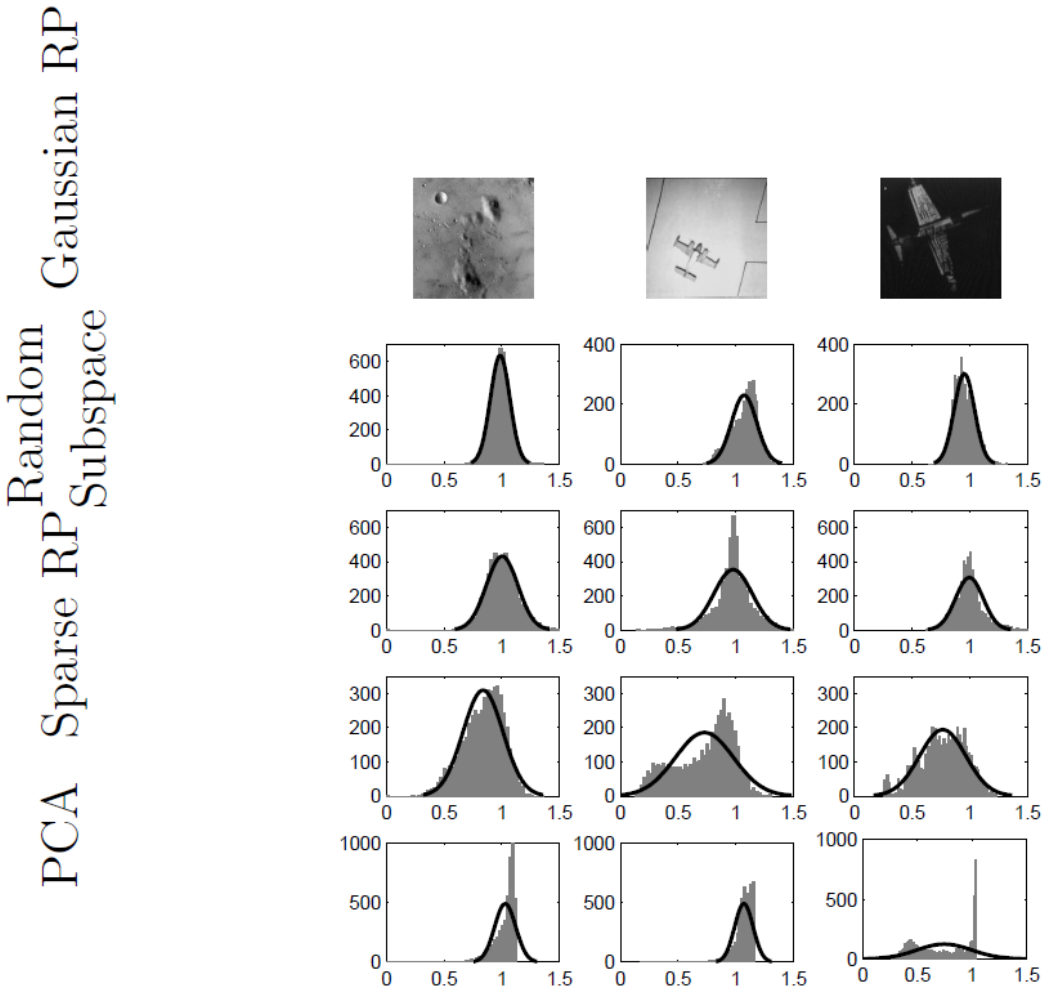



Figure: Fixed $k$, small $c$: Histograms of $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ for $k = 50$ dimensions on three representative images with overlaid normal density plots, $n = 4950$.

# Quantiles vs. $k$

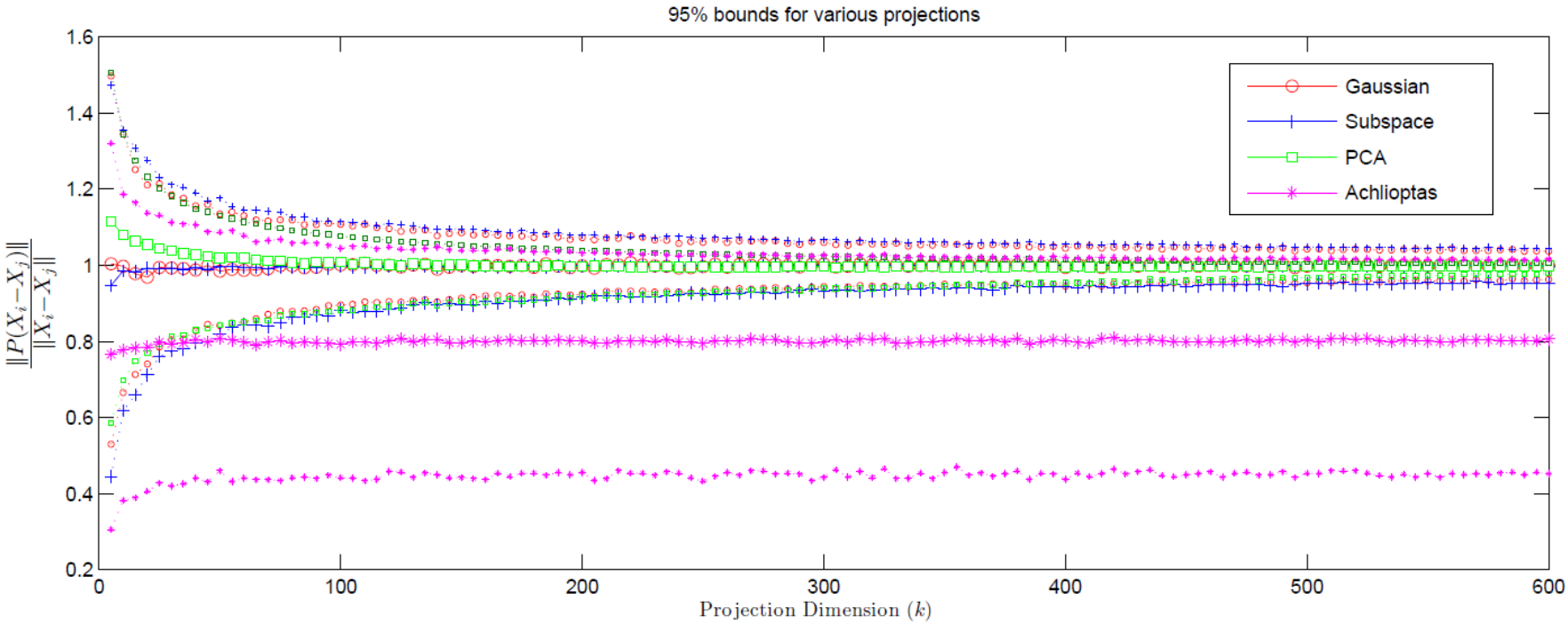



Figure: Mean and 5th and 95th percentiles of $\frac{\|P(X_i-X_j)\|}{\|X_i-X_j\|}$ for image data vs. $k$. We see that for $k \gtrsim 80$ Gaussian RP and RS are indistinguishable on these data. Note also the 5th percentile for Sparse RP cf. Figure 9: Sparse RP frequently seems to underestimate norms.

# Average Running Times

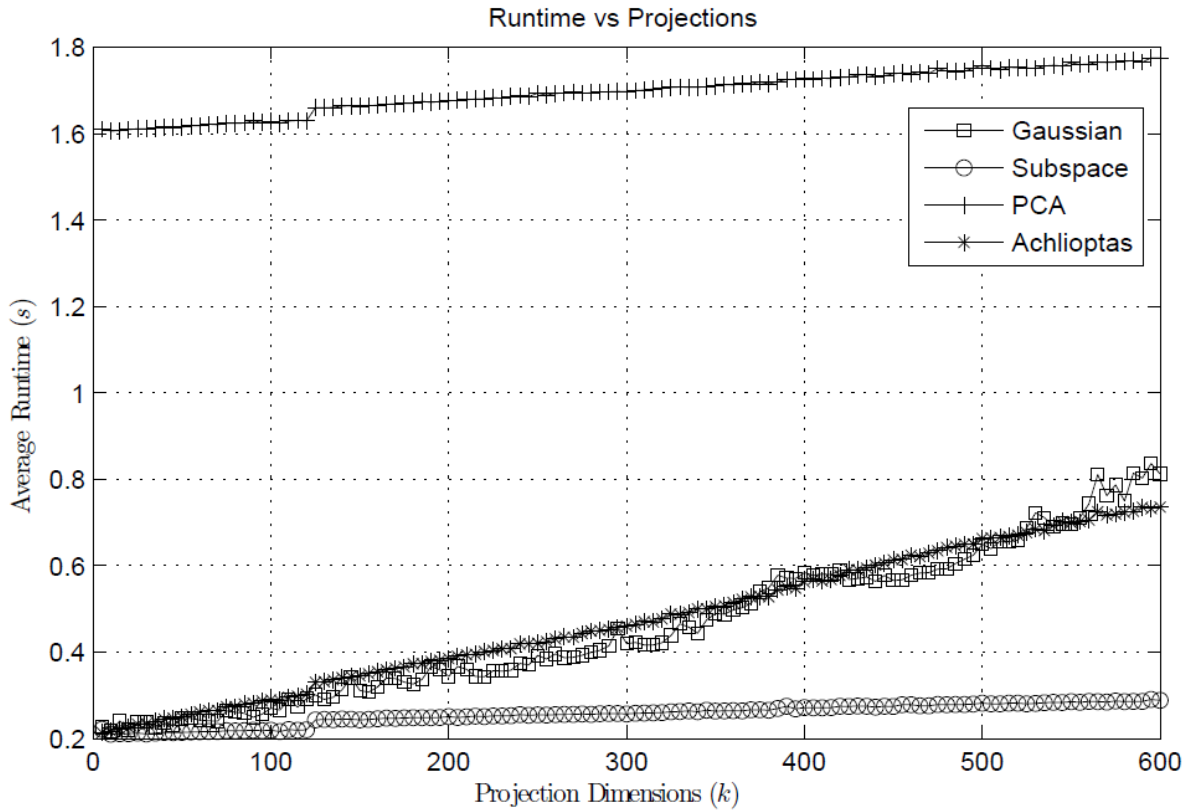
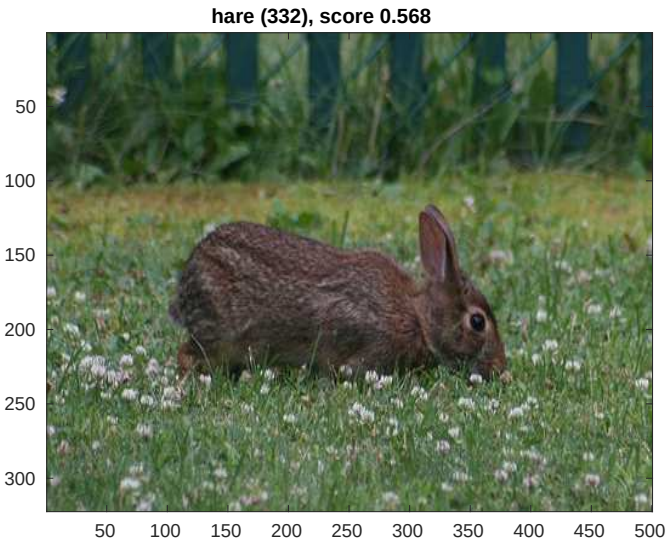


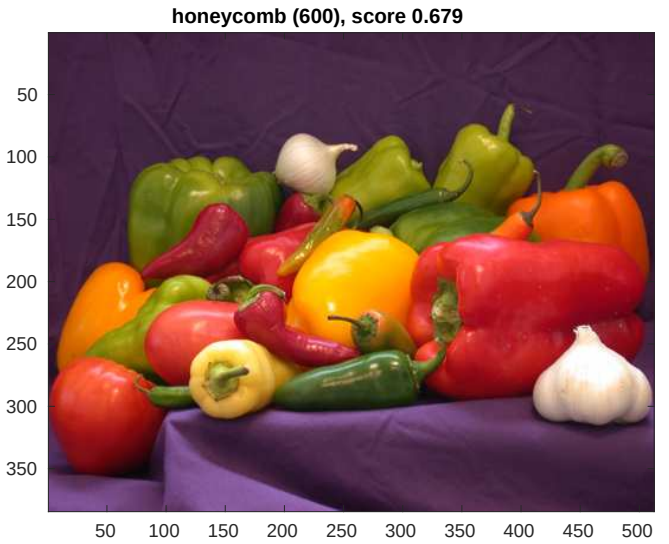

Figure: Comparison of the runtime on dense image data with dimensionality $d = 2500$.

# Preliminary Experiments with NNs

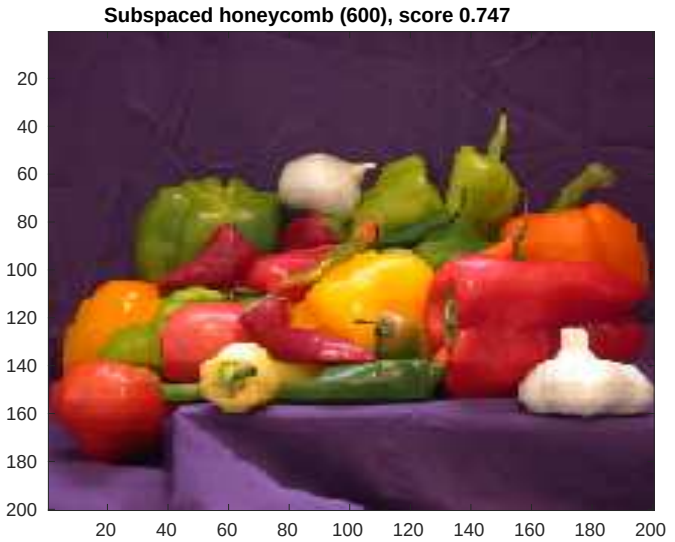
- Classification performance evaluation only (so far. . .)
- Used (challenge-winning) GoogLeNet with pretrained weights from Imagenet challenge.
- Original images replaced with versions compressed using RS.
- Evaluation on 100,000 full colour images of varying sizes and resolutions from ILSVRC 2012 Imagenet challenge - 1000 classes.
- Classification error using one RS example marginally worse than state-of-art, RS ‘voting’ ensemble approach (sum of scores) better than state-of-art.

# Example Image Inputs and Outcomes (1)

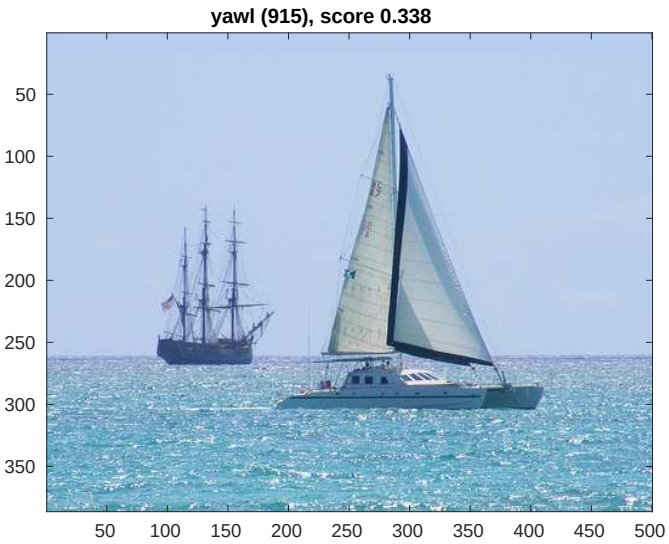


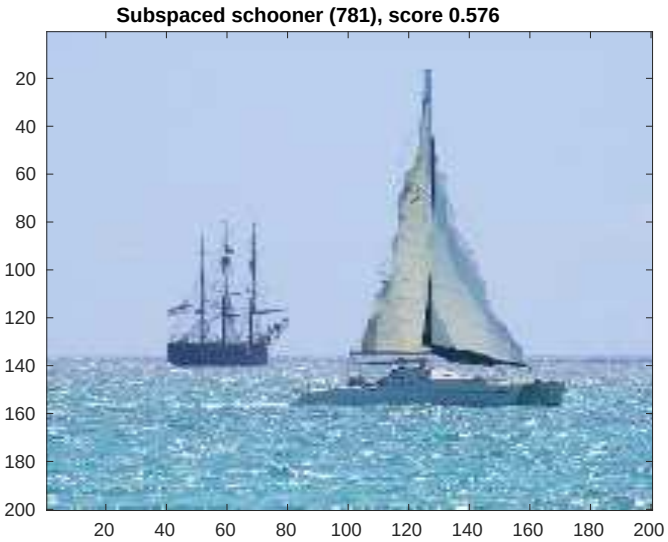




# Example Image Inputs and Outcomes (2)








# Experiments: Effect of Ensemble Size, $k$, Top 1 Error

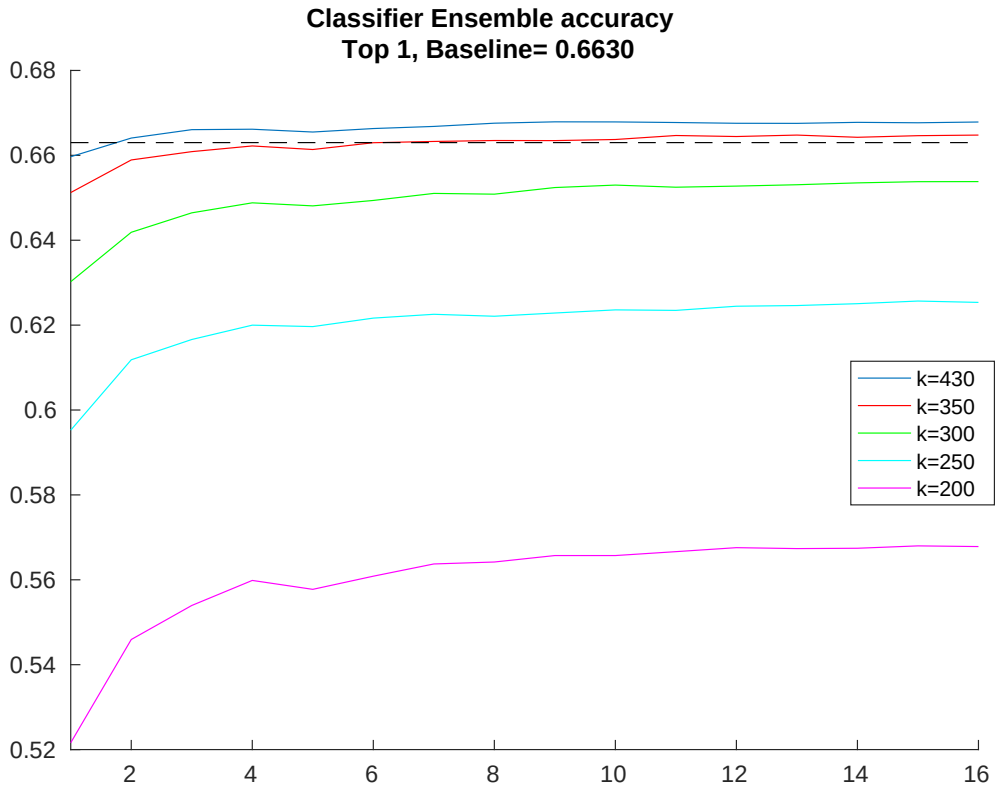



Figure: Top 1 test error rate vs. ensemble size estimated from 12 runs over 100,000 images. Error bars omitted: 1 s.e. is approximately width of plotted line.

# Experiments: Effect of Ensemble Size, $k$, Top 3 Error

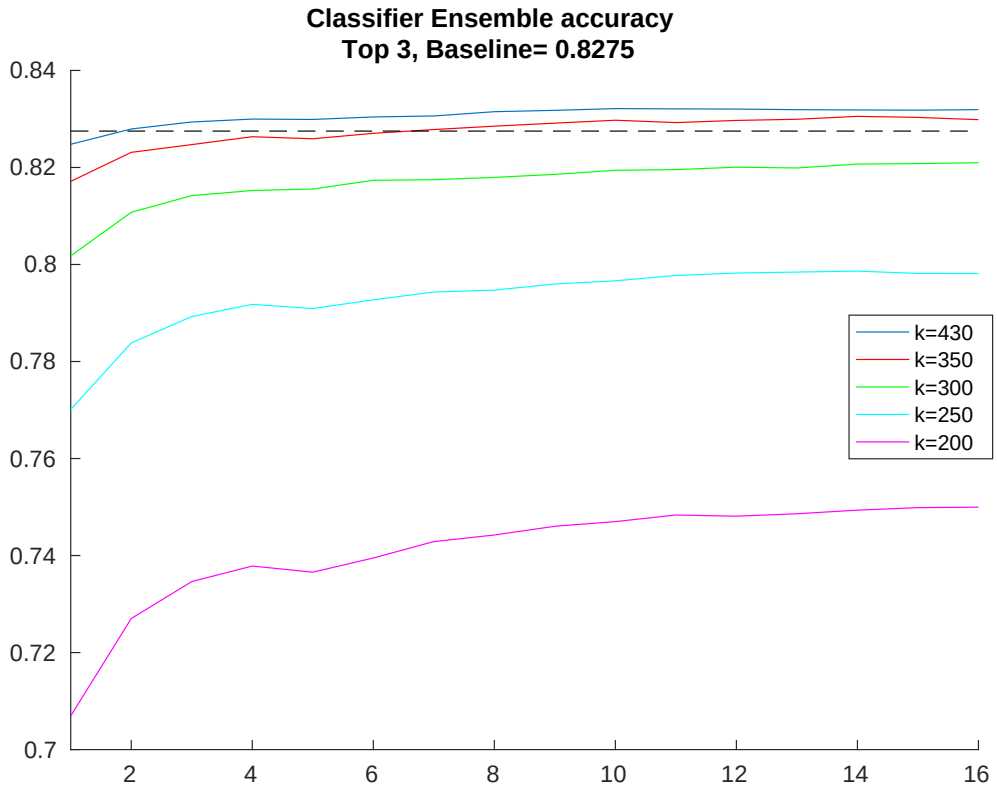



Figure: Top 3 test error rate vs. ensemble size estimated from 12 runs over 100,000 images. Error bars omitted: 1 s.e. is approximately width of plotted line.

# Experiments: Effect of Ensemble Size, $k$, Top 5 Error

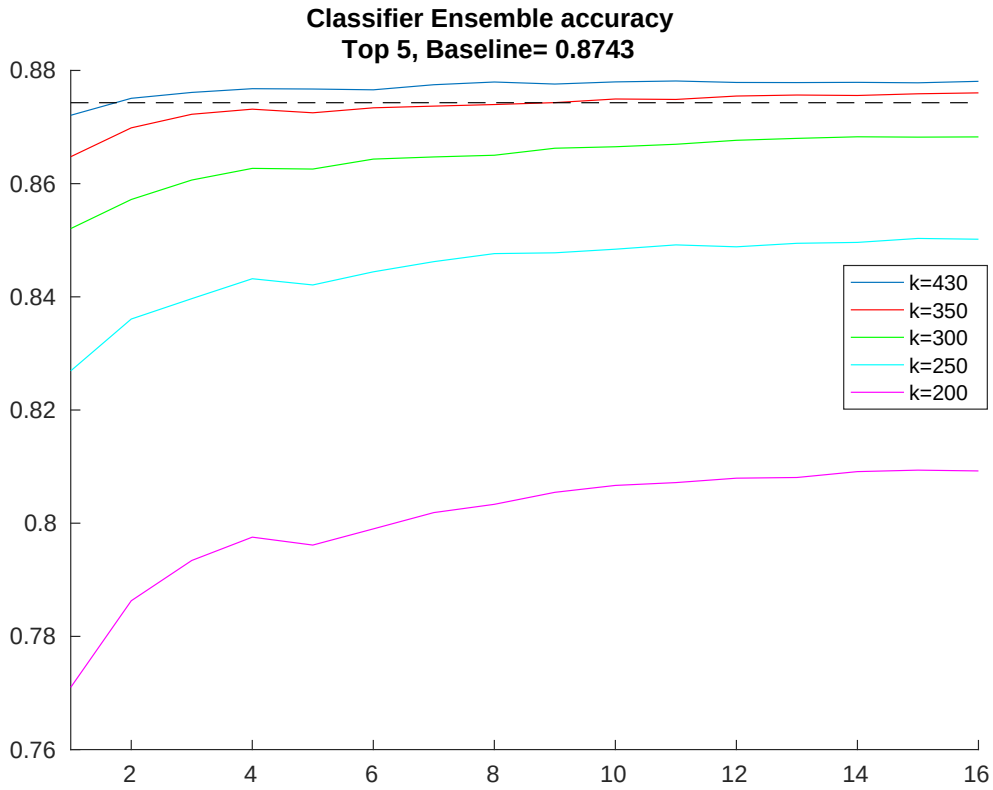


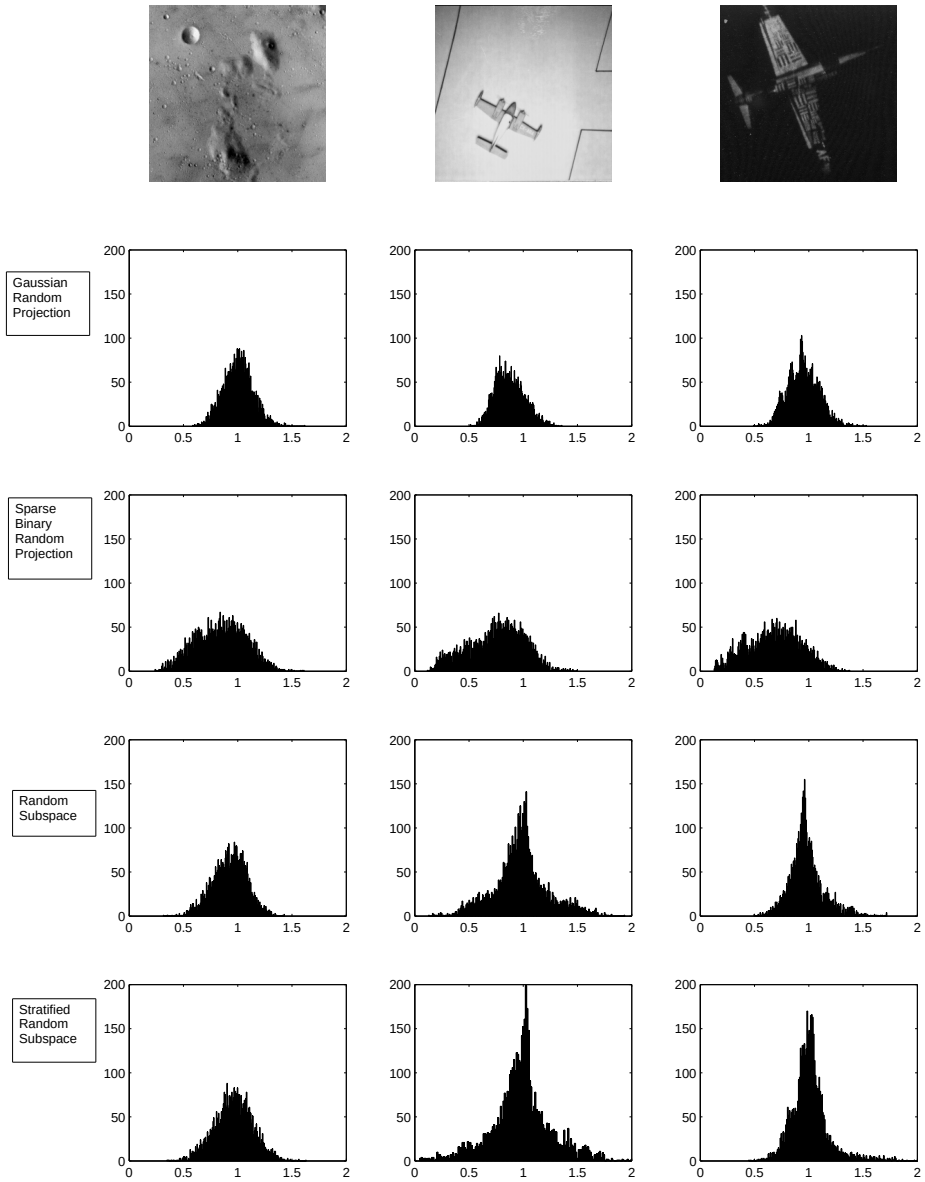

Figure: Top 5 test error rate vs. ensemble size estimated from 12 runs over 100,000 images. Error bars omitted: 1 s.e. is approximately width of plotted line.

# Preliminary Experiments with Stratification

- Statistical theory suggests if data can be split into approximately homoskedastic (uniform variance) strata with well-separated means, then variance of sampling distribution of population mean can be reduced by stratified sampling (here population mean $\equiv$ Euclidean norm).
- We transpose the data matrix and apply $k$-means clustering to the features (i.e. rather than the observations) to search for such strata.
- No obvious 'best' number of clusters for all images: Highly data-dependent. Sweet spot seems to be between 3 and 7 clusters for the image data we worked with.
- Two stratification schemes tried: Proportional Allocation (gives unbiased estimate of norms) and Neyman Allocation (gives biased estimate of norms, but with reduced standard error).
- Obtains improved stability in norm estimates, as theory would suggest, but improvement only marginal.
- Conclusion: $k$-means not a great way to find strata.

# Stratification Experiments:

Stratified sampling with 3 strata and proportional allocation.
Histograms of $\frac{\|P(X_i - X_j)\|}{\|X_i - X_j\|}$ for $k = 50$ dimensions on three representative images, $n = 4950$.



# Conclusions and Future Work

- Random projections have a wide range of *theoretically well-motivated* and *effective* applications in machine learning and data mining.
- Overhead of matrix-matrix multiplication can be removed for 'smooth' datasets using RS, with no obvious disadvantages.
- Variance in projected norms can be further reduced by using RS with stratified sampling. How to better identify strata automatically and cheaply an interesting (and probably hard) problem.
- RS provides one potential route to meaningful theory, with typical-case guarantees, for dropout regularization of NNs – this would be interesting in its own right.
- Potential of RS to both speed up back-propagation and reduce model size of deep NNs intriguing - we have just started work in this direction, watch this space!
- Further experiments and extension of RS ensemble idea – some potential applications in sight e.g. edge computing.

# References I


[AC06] N. Ailon and B. Chazelle, *Approximate nearest neighbors and the fast johnson-lindenstrauss transform*, Proceedings of the thirty-eighth annual ACM symposium on Theory of computing, ACM, 2006, pp. 557–563.

[Ach03] D. Achlioptas, *Database-friendly random projections: Johnson-Lindenstrauss with binary coins*, Journal of Computer and System Sciences **66** (2003), no. 4, 671–687.

[AMS96] N. Alon, Y. Matias, and M. Szegedy, *The space complexity of approximating the frequency moments*, Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, ACM, 1996, pp. 20–29.

[AV09] R. Avogadri and G. Valentini, *Fuzzy ensemble clustering based on random projections for dna microarray data analysis*, Artificial Intelligence in Medicine **45** (2009), no. 2, 173–183.

[BD09] C. Boutsidis and P. Drineas, *Random projections for the nonnegative least-squares problem*, Linear Algebra and its Applications **431** (2009), no. 5-7, 760–771.

[BM01] E. Bingham and H. Mannila, *Random projection in dimensionality reduction: applications to image and text data.*, Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2001) (F. Provost and R. Srikant, ed.), 2001, pp. 245–250.

[BW09] R.G. Baraniuk and M.B. Wakin, *Random projections of smooth manifolds*, Foundations of Computational Mathematics **9** (2009), no. 1, 51–77.

# References II


[CJS09] R. Calderbank, S. Jafarpour, and R. Schapire, *Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain*, Tech. report, Rice University, 2009.

[CS15] Timothy I Cannings and Richard J Samworth, *Random projection ensemble classification*, arXiv preprint arXiv:1504.04595 (2015).

[CT06] E.J. Candes and T. Tao, *Near-optimal signal recovery from random projections: Universal encoding strategies?*, Information Theory, IEEE Transactions on **52** (2006), no. 12, 5406–5425.

[CT10] Emmanuel J Candès and Terence Tao, *The power of convex relaxation: Near-optimal matrix completion*, IEEE Transactions on Information Theory **56** (2010), no. 5, 2053–2080.

[Das99] S. Dasgupta, *Learning Mixtures of Gaussians*, Annual Symposium on Foundations of Computer Science, vol. 40, 1999, pp. 634–644.

[DF08] S. Dasgupta and Y. Freund, *Random projection trees and low dimensional manifolds*, Proceedings of the 40th annual ACM symposium on Theory of computing, ACM, 2008, pp. 537–546.

[DG02] S. Dasgupta and A. Gupta, *An Elementary Proof of the Johnson-Lindenstrauss Lemma*, Random Struct. Alg. **22** (2002), 60–65.

# References III


[DK12] R.J. Durrant and A. Kabán, *Error bounds for Kernel Fisher Linear Discriminant in Gaussian Hilbert space*, Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AIStats 2012), 2012.

[DK15] Robert J Durrant and Ata Kabán, *Random projections as regularizers: learning a linear discriminant from fewer observations than dimensions*, Machine Learning **99** (2015), no. 2, 257–286.

[Don06] D.L. Donoho, *Compressed Sensing*, IEEE Trans. Information Theory **52** (2006), no. 4, 1289–1306.

[FB03] X.Z. Fern and C.E. Brodley, *Random projection for high dimensional data clustering: A cluster ensemble approach*, International Conference on Machine Learning, vol. 20, 2003, p. 186.

[FM03] D. Fradkin and D. Madigan, *Experiments with random projections for machine learning*, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 522–529.

[GBN05] N. Goel, G. Bebis, and A. Nefian, *Face recognition experiments with random projection*, Proceedings of SPIE, vol. 5779, 2005, p. 426.

[Ho98] T.K. Ho, *The random subspace method for constructing decision forests*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **20** (1998), no. 8, 832–844.

# References IV


[HWB07] C. Hegde, M.B. Wakin, and R.G. Baraniuk, *Random projections for manifold learningproofs and analysis*, Neural Information Processing Systems, 2007.

[IM98] P. Indyk and R. Motwani, *Approximate nearest neighbors: towards removing the curse of dimensionality*, Proceedings of the thirtieth annual ACM symposium on Theory of computing, ACM New York, NY, USA, 1998, pp. 604–613.

[KBD16] Ata Kabán, Jakramate Bootkrajang, and Robert John Durrant, *Toward large-scale continuous eda: A random matrix theory perspective*, Evolutionary computation **24** (2016), no. 2, 255–291.

[KMV12] A.T. Kalai, A. Moitra, and G. Valiant, *Disentangling gaussians*, Communications of the ACM **55** (2012), no. 2, 113–120.

[LD17] Nick Lim and Robert J. Durrant, *Linear dimensionality reduction in linear time: Johnson-lindenstrauss-type guarantees for random subspace*, http://arxiv.org/abs/1705.06408 (2017), no. 1705.06408.

[Led01] M. Ledoux, *The concentration of measure phenomenon*, vol. 89, American Mathematical Society, 2001.

[LN14] Kasper Green Larsen and Jelani Nelson, *The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction*, arXiv preprint arXiv:1411.2404 (2014).

[LN16] _____, *Optimality of the johnson-lindenstrauss lemma*, arXiv preprint arXiv:1609.02094 (2016).

# References V


[Mat08] J. Matoušek, *On variants of the johnson–lindenstrauss lemma*, Random Structures & Algorithms **33** (2008), no. 2, 142–156.

[Rec11] B. Recht, *A simpler approach to matrix completion*, Journal of Machine Learning Research **12** (2011), 3413–3430.

[RR08] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, Advances in neural information processing systems **20** (2008), 1177–1184.

[Sar06] T. Sarlos, *Improved approximation algorithms for large matrices via random projections*, Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on, IEEE, 2006, pp. 143–152.

[SR09] A. Schclar and L. Rokach, *Random projection ensemble classifiers*, Enterprise Information Systems (Joaquim Filipe, Jos Cordeiro, Wil Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, and Clemens Szyperski, eds.), Lecture Notes in Business Information Processing, vol. 24, Springer, 2009, pp. 309–316.

# Appendix

**Proposition** JLL for dot products.
Let $x_n, n = \{1 \dots N\}$ and $u$ be vectors in $\mathbb{R}^d$ s.t. $\|x_n\|, \|u\| \leqslant 1$.
Let $R$ be a $k \times d$ RP matrix with i.i.d. entries $R_{ij} \sim \mathcal{N}(0, 1/\sqrt{k})$ (or with zero-mean sub-Gaussian entries).
Then for any $\epsilon, \delta > 0$, if $k \in \mathcal{O}\left(\frac{8}{\epsilon^2} \log(4N/\delta)\right)$ w.p. at least $1 - \delta$ we have:

$$|x_n^T u - (Rx_n)^T Ru| < \epsilon \tag{1}$$

simultaneously for all $n = \{1 \dots N\}$.

# Proof of JLL for dot products

Outline: Fix one $n$, use parallelogram law and JLL twice, then use union bound.

$$
\begin{aligned}
4(Rx_n)^T(Ru) &= \|Rx_n + Ru\|^2 - \|Rx_n - Ru\|^2 && (2)\\
&\geqslant (1-\epsilon)\|x_n + u\|^2 - (1+\epsilon)\|x_n - u\|^2 && (3)\\
&= 4x_n^T u - 2\epsilon(\|x_n\|^2 + \|u\|^2) && (4)\\
&\geqslant 4x_n^T u - 4\epsilon && (5)
\end{aligned}
$$

Hence, $(Rx_n)^T(Ru) \geqslant x_n^T u - \epsilon$, and because we used two sides of JLL, this holds except w.p. no more than $2\exp(-k\epsilon^2/8)$.
The other side is similar and gives $(Rx_n)^T(Ru) \leqslant x_n^T u + \epsilon$ except w.p. $2\exp(-k\epsilon^2/8)$.
Put together, $|(Rx_n)^T(Ru) - x_n^T u| \leqslant \epsilon \cdot \frac{\|x\|^2+\|u\|^2}{2} \leqslant \epsilon$ holds except w.p. $4\exp(-k\epsilon^2/8)$.
This holds for a fixed $x_n$. To ensure that it holds for all $x_n$ together, we take union bound and obtain eq.(1) must hold except w.p. $4N\exp(-k\epsilon^2/8)$. Finally, solving for $\delta$ we obtain that $k \geqslant \frac{8}{\epsilon^2}\log(4N/\delta)$.