

Experiments with Random Projections for Machine Learning

Dmitriy Fradkin and David Madigan
Rutgers University, Piscataway, NJ

1 Purpose

To evaluate the effectiveness of Random Projections (RPs) compared with PCA for machine learning.

2 Supervised Learning Problem

Inductive supervised learning infers a functional relation $y = f(\mathbf{x})$ from a set of training examples

$$T = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}.$$

In what follows the inputs are vectors $\mathbf{x}_i = [x_{i1}, \dots, x_{ip}]$ in \mathbb{R}^p , $y \in \{-1, 1\}$, and we refer to f as a classifier. The objective is to produce a classifier that makes accurate predictions for future input vectors.

3 The Need for Dimensionality Reduction

Data with large dimensionality presents problems for many machine learning algorithms, since their computational complexity can be superlinear in p and they may need complexity control to avoid overfitting.

Traditional methods (PCA/SVD) are computationally expensive:

- PCA is $O(p^2n) + O(p^3)$ [Golub and van Loan, 1983]
- SVD is somewhat more efficient: for sparse matrices with r non-zero entries there are $O(prn)$ algorithms [Papadimitriou et. al. 1998].

4 Random Projections

A theorem due to Johnson and Lindenstrauss (JL Theorem) states that for a set of points of size n in p -dimensional Euclidean space there exists a linear transformation of the data into a q -dimensional space, $q \geq O(\epsilon^{-2} \log(n))$ that preserves distances up to a factor $1 \pm \epsilon$ [Johnson and Lindenstrauss, 1984].

Theorem 1 [Achlioptas, 2001] Given n points in \mathbb{R}^p (in form of an $n \times p$ matrix X), choose $\epsilon, \beta > 0$ and $q \geq \frac{4+2\beta}{\epsilon^2} \ln(n)$, and let $E = \frac{1}{\sqrt{q}}XP$, for projection matrix P . Then, mapping from X to E preserves distances up to factor $1 \pm \epsilon$ for all rows in X with probability $(1 - n^{-\beta})$. Projection matrix P , $p \times q$, can be constructed in one of the following ways:

- $r_{ij} = \pm 1$ with probability 0.5 each
- $r_{ij} = \sqrt{3} * (\pm 1$ with probability 1/6 each, or 0 with probability 2/3)

The above projections are easy to implement and to compute.

Constructing a $p \times q$ random matrix is $O(pq)$. Performing the projection for n points is $O(npq)$.

We chose to implement the first of the methods suggested by Achlioptas: $r_{ij} = \pm 1$ with

probability 0.5 each.

Since we are not concerned with preserving distances per se, but only with preserving separation between points, we do not scale our projection: $E = XP$ instead of $E = \frac{1}{\sqrt{q}}XP$

5 Related Work

- Theoretical Approximate Nearest Neighbor algorithm with polynomial preprocessing and query time polynomial in p and $\log n$ [Indyk and Motwani, 1998]. Also, the first tight bounds on the quality of randomized dimensionality reduction.
- Learning mixtures of Gaussians in high dimensions [Dasgupta 1999], [Dasgupta, 2000]. Combination of RP with EM algorithm gives good classification results on a handwritten digit dataset.
- Preservation of volumes and affine distances [Magen 2002].
- Deterministic algorithm for constructing JL mappings [Engebretsen, Indyk and O'Donnell 2002], used to derandomize several randomized algorithms.
- Approximate kernel computations [Achlioptas, McSherry and Schölkopf, 2001], similarity computations for histogram models [Thaper et. al 2002].

[Bingham and Mannila, 2001] experimentally show that RP preserve similarity (inner products) well even when dimensionality of projection is moderate. (Also compared RP to PCA, SVD and DCT). Their data had $p = 5000$, $n = 2262$ for text data, and $p = 2500$, $n = 1000$ for image data. Projections were done to $q \in [1, 800]$.

6 Description of Data

Ionosphere, Spambase and Internet Ads were taken from UCI repository Colon and Leukemia were first used in [Alon et. al 1999] and [Golub et. al. 1999] respectively.

Table 1:

Name	# Instances	# Attributes
Ion	351	34
Spam	4601	57
Ads	3279	1554
Colon	62	2000
Leukemia	72	3571

- Colon and Leukemia datasets are of a high dimensionality but have few points. Thus we would expect RP to high dimensions to lead to good results, while PCA results should stop changing after some point. For these dataset we perform projections into spaces of dimensionality 5, 10, 25, 50, 100, 200 and 500.
- Ionosphere and Spam are relatively low-dimensional but have many more points

than Colon and Leukemia datasets. Such combination in theory leaves little space for RP to improve, while PCA should be able to do well. We project to dimensions 5, 10, 15, 20, 25 and 30.

- Ads dataset is both large and high-dimensional. We perform projections are done to 5, 10, 25, 50, 100, 200 and 500.

7 Experimental Setup

We compare PCA and RP using a number of standard machine learning tools:

- decision trees (C4.5 - [Quinlan, 1993])
- linear SVM (SVMlight - [Joachims, 1999])
- nearest neighbor (NN)

Test set sizes were kept constant over different splits: Ionosphere - 51, Spambase - 1601, Colon - 12, Leukemia - 12, Ads - 1079.

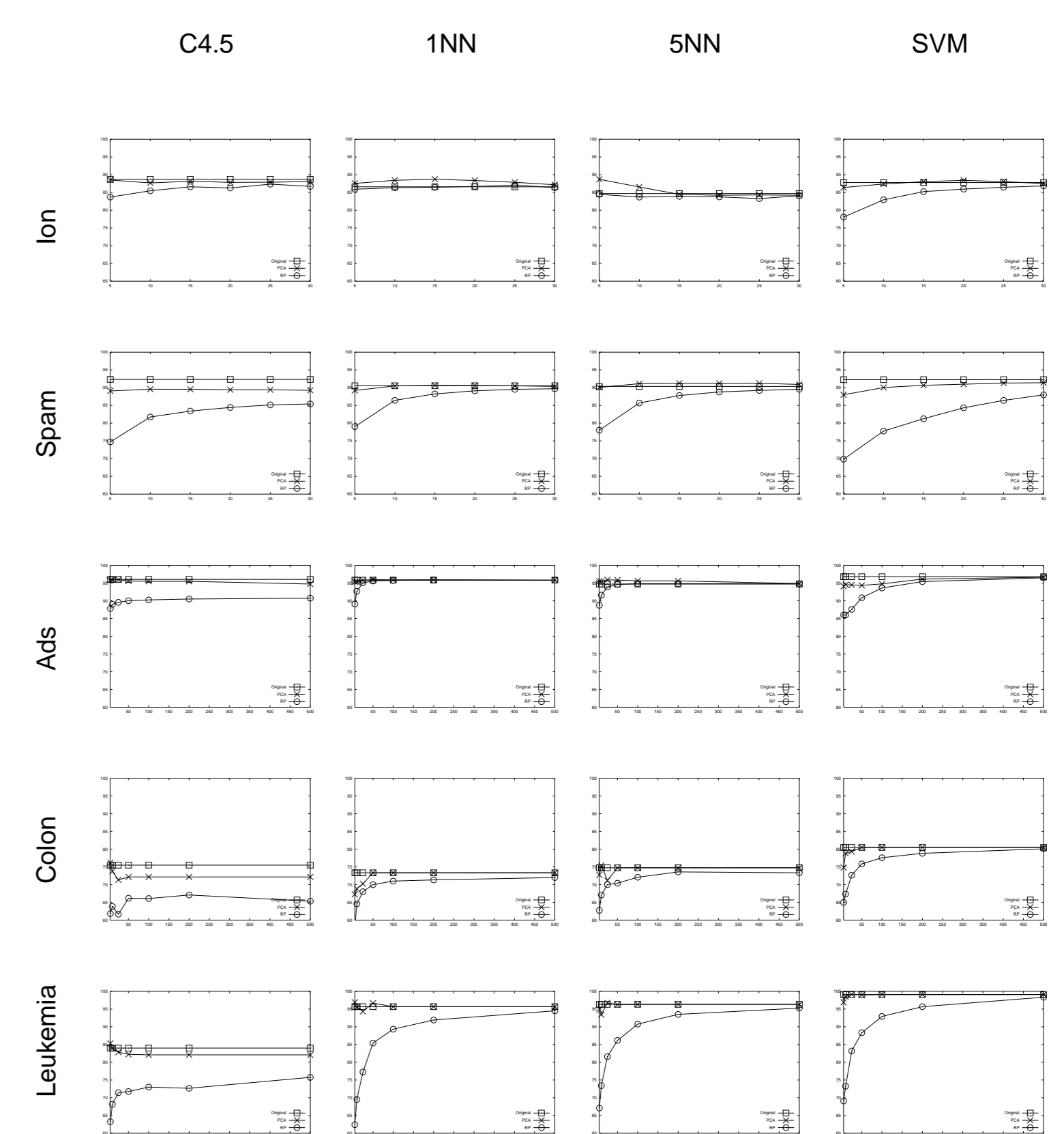


Table 2: Accuracy (Y-axis) using PCA and RP, compared to performance in the original dimension, plotted against the projection dimension (X-axis)

8 Conclusions

- RPs performance was (predictably) below the level of PCA
- But: RPs performance was improving noticeably with increasing dimensionality
- RPs seem well suited for use with Nearest Neighbor methods
- Decision tree did not combine with RP in a satisfactory way.

9 Directions for Further Study

- Train multiple classifier on several different projections and combine their decisions
 - different projections to the same dimension
 - projections to different dimensions
- Explore performance on significantly larger datasets

10 Acknowledgments

We would like to thank Andrei Angheliescu for providing the kNN code.