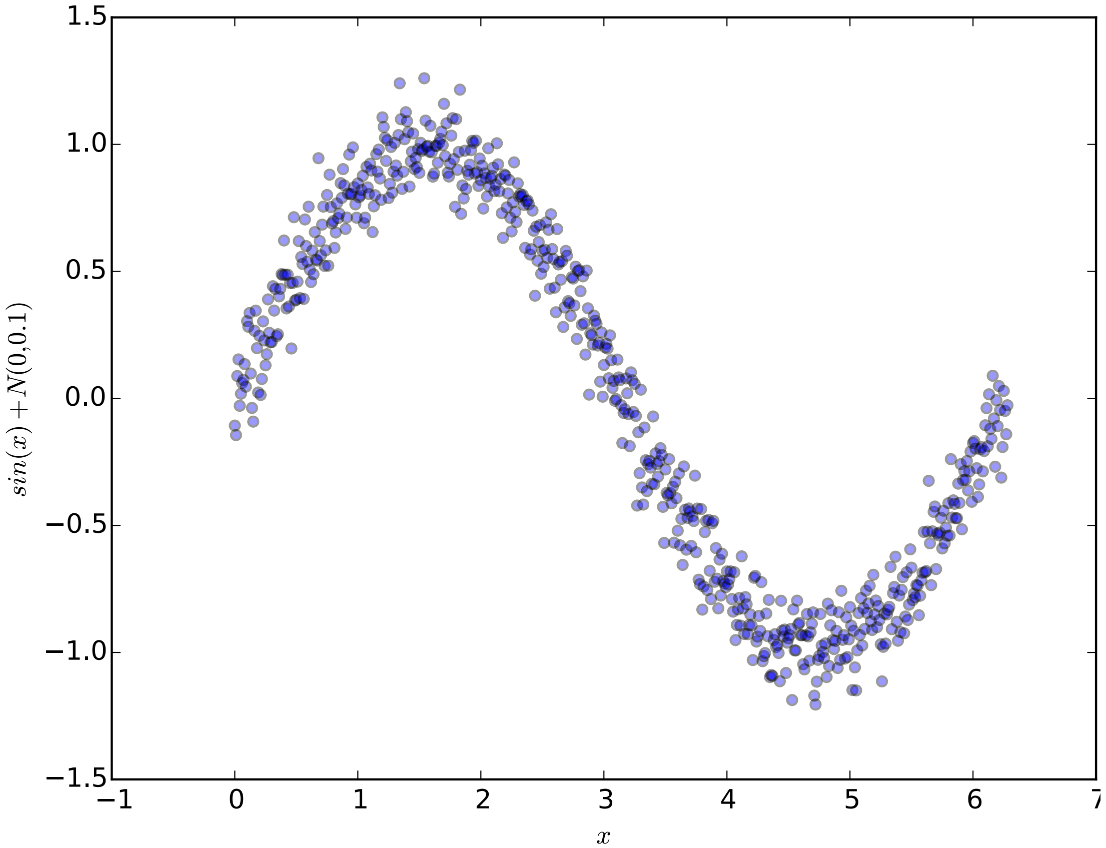
Andrew Nystrom, Research Scientist at Contata Solutions
John Hughes, Professor of Computer Science at Brown University

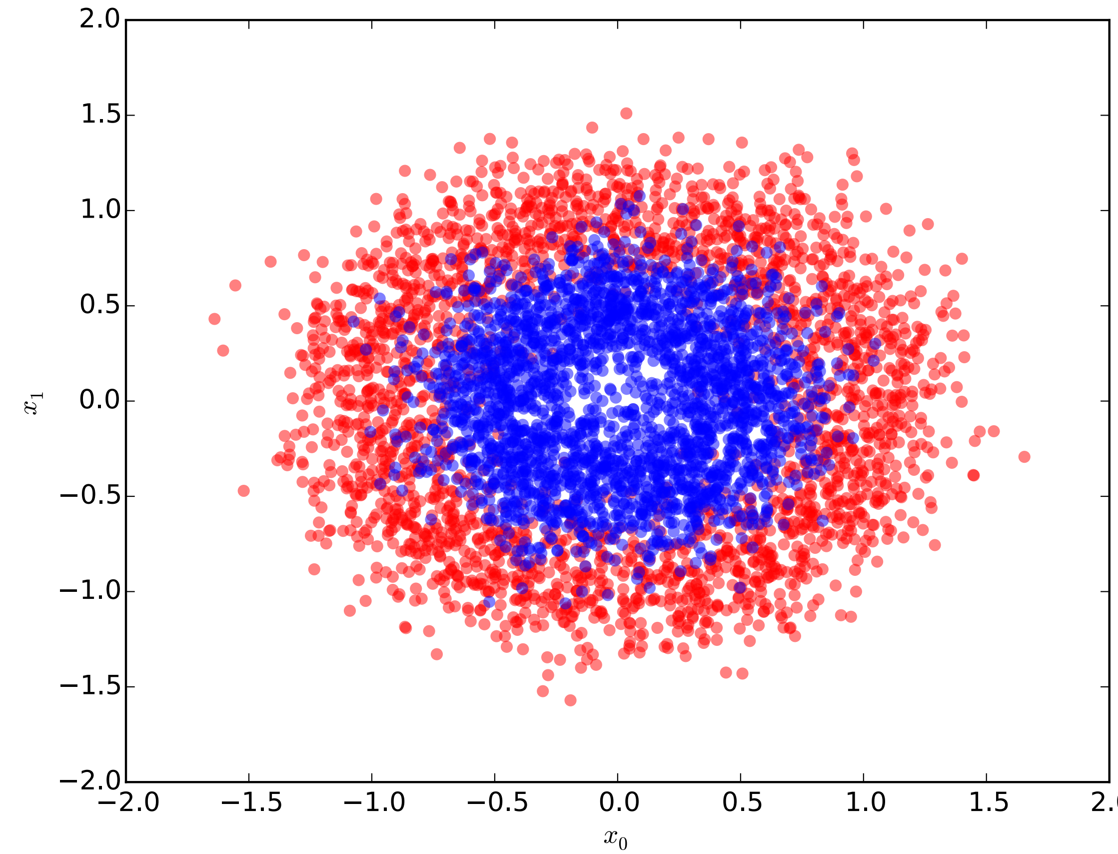
Efficient Calculation of Second Degree Polynomial Features on Sparse Matrices

Handling Nonlinearity

Regression



Classification

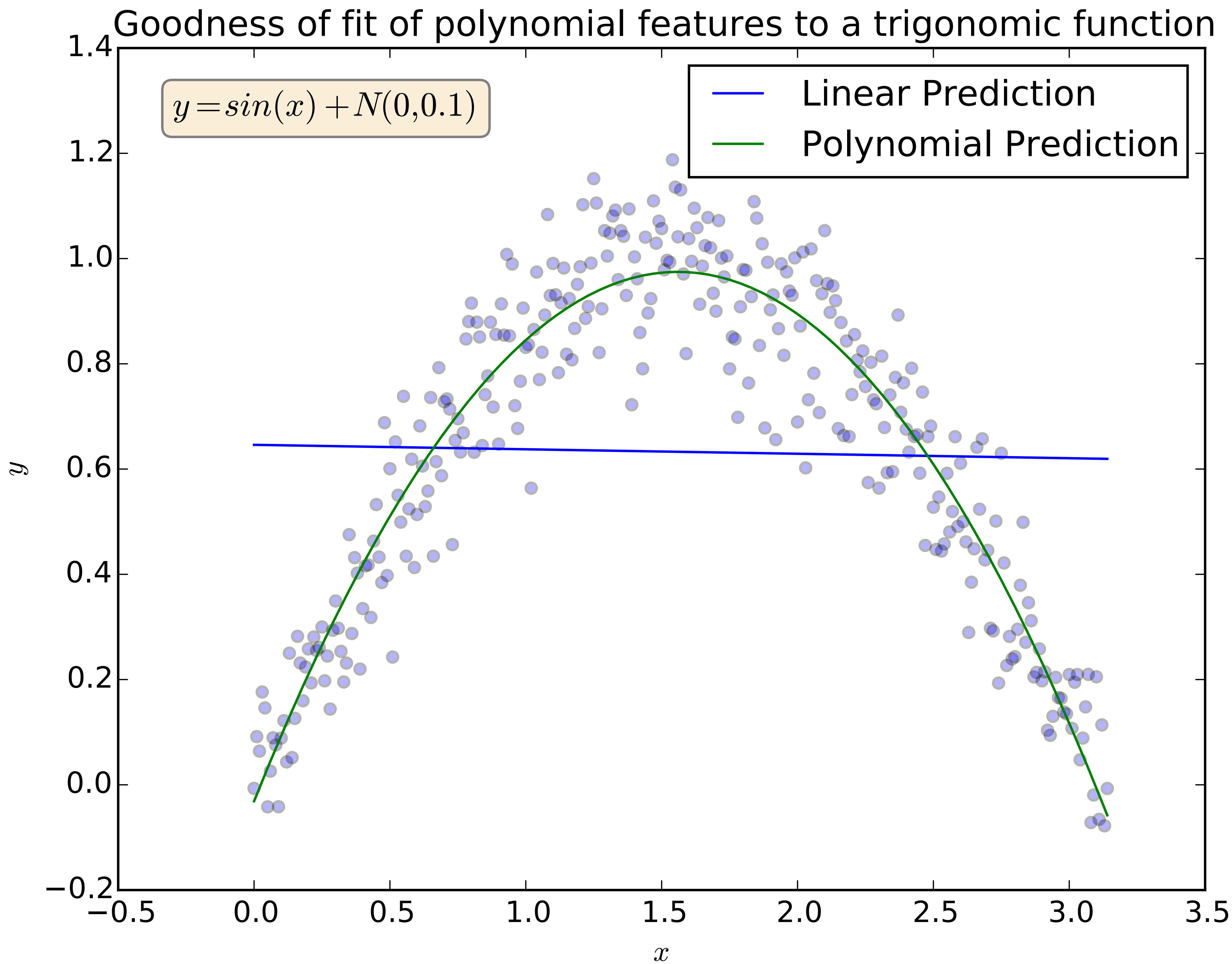


- Work in kernel space, e.g. rbf kernel
  - Time complexity $O(DN^2)$
- Nonlinear model, e.g. gradient boosting on stubs
  - Doesn't scale when over a couple hundred dimensions
- Add non-linear features

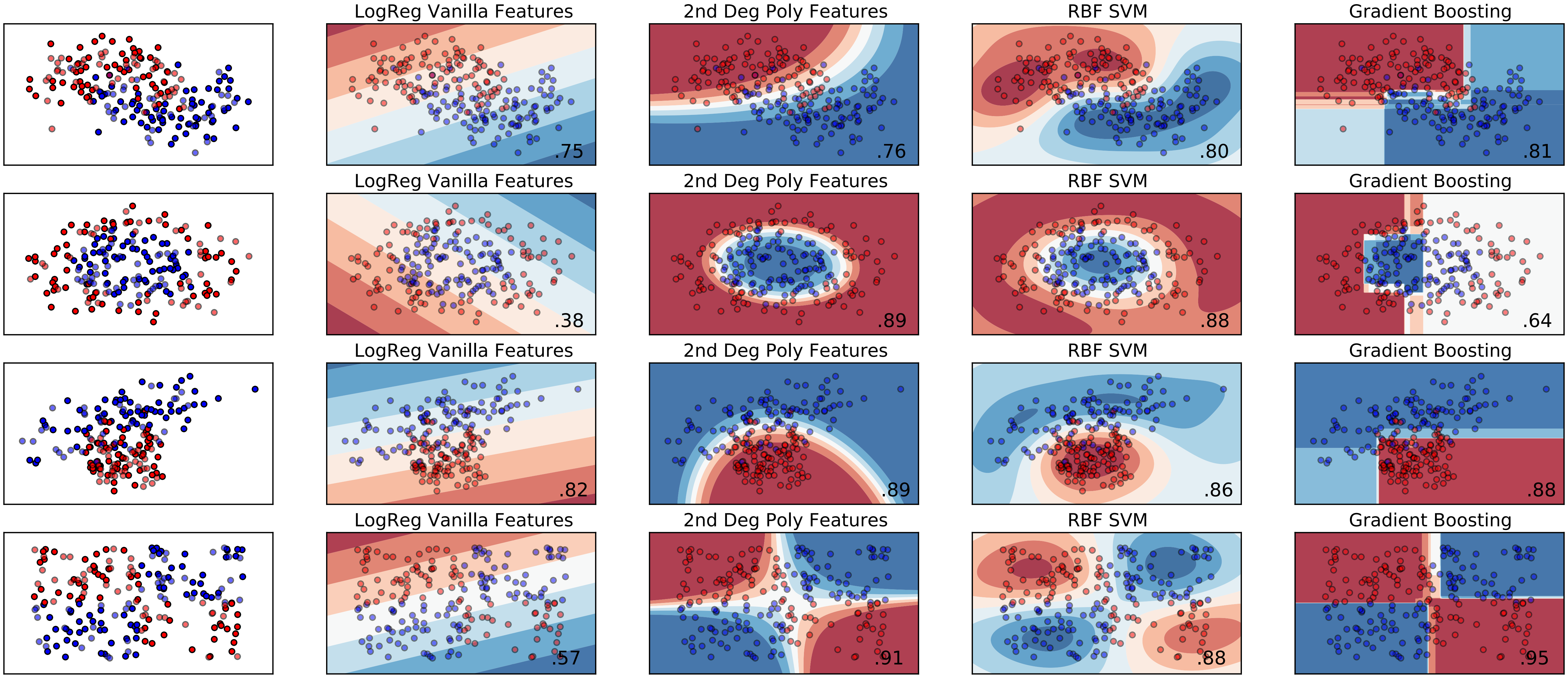
Polynomial Features

- Products of all combinations of features
  - $[a, b] \rightarrow [a, b, a^2, ab, b^2]$
- When input to linear model:
  - $y = w_0 \cdot a + w_1 \cdot b$
  - $y = w_0 \cdot a + w_1 \cdot b + w_2 \cdot a^2 + w_3 \cdot ab + w_4 \cdot b^2$
- All parabolas, hyperbolas, ellipses

Polynomial Features in Regression



Polynomial Features in Classification



Naive Calculation

$\textsc{Dense Polynomials}(A)$

$$
\begin{array}{l}
\quad N = \text{row count of } A \\
\quad D = \text{column count of } A \\
\quad B = \text{Matrix of size } N \times \frac{D^2+D}{2} \\
\quad \textbf{for } row \text{ in } A \\
\qquad k = 0 \\
\qquad \textbf{for } \imath = 0 \textbf{ to } D \\
\qquad\quad \textbf{for } \jmath = i \textbf{ to } D \\
\qquad\qquad B[i,k] = row[i] \cdot row[j] \\
\qquad\qquad k = k + 1
\end{array}
$$

B[*i*,*k*] is zero whenever *row*[*i*] or *row*[*j*] is zero
Many unnecessary products when *A* is sparse.

Main Idea: Exploiting Sparse Structure

- Sparse matrix formats track nonzero column indices
- Only iterate over these!
- Given two columns, where does the product belong?
- Assume we have such a mapping…

Exploiting Sparse Structure (cont.)

SPARSE POLYNOMIAL FEATURES(A)

$\text{PolyMap}(a, b) = ?$
$N = $ row count of A
$D = $ column count of A
$B = $ Compressed Sparse Row Matrix of size $N \times \frac{D^2+D}{2}$
for row in A
    $N_{zc} = $ nonzero columns of row
    for $i = 0$ **to** $|N_{zc}|$
        for $j = i$ **to** $|N_{zc}|$
            $k = \text{PolyMap}(N_{zc}[i], N_{zc}[j])$
            $r = $ index of row
            $B[r, k] = row[N_{zc}[i]] \cdot row[N_{zc}[j]]$

Find PolyMap function. Maps pairs of column indices onto $(D^2+D)/2$

Finding the Mapping Function

$\{x_i \cdot x_j : i, j \in \{0, 1, ..., D-1\} \wedge i \leq j\}$

$\{(i, j) : i, j \in \{0, 1, ..., D-1\} \wedge i \leq j\} \rightarrow \{0, 1, ..., \frac{D^2+D}{2} - 1\}$

$$
j
$$

$$
\begin{array}{cc}
 & \begin{array}{ccccc} x_0 & x_1 & x_2 & x_3 & x_4 \end{array} \\
\begin{array}{c} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} &
\begin{pmatrix}
0 & 1 & 2 & 3 & 4 \\
- & 5 & 6 & 7 & 8 \\
- & - & 9 & 10 & 11 \\
- & - & - & 12 & 13 \\
- & - & - & - & 14
\end{pmatrix}
\end{array}
$$

$\vec{x}$

i

$$
\text{polynomial-index}(i, j | D) = \frac{2Di - i^2 + 2j - 3i - 2}{2} + i + 1
$$

Algorithm Comparison

$\textsc{Dense Polynomials}(A)$

N = row count of A
D = column count of A
B = Matrix of size $N \times \frac{D^2+D}{2}$
for row in A
  $k = 0$
  for $i = 0$ **to** D
    for $j = i$ **to** D
      $B[i,k] = row[i] \cdot row[j]$
      $k = k + 1$

O(ND²), Θ(ND²)

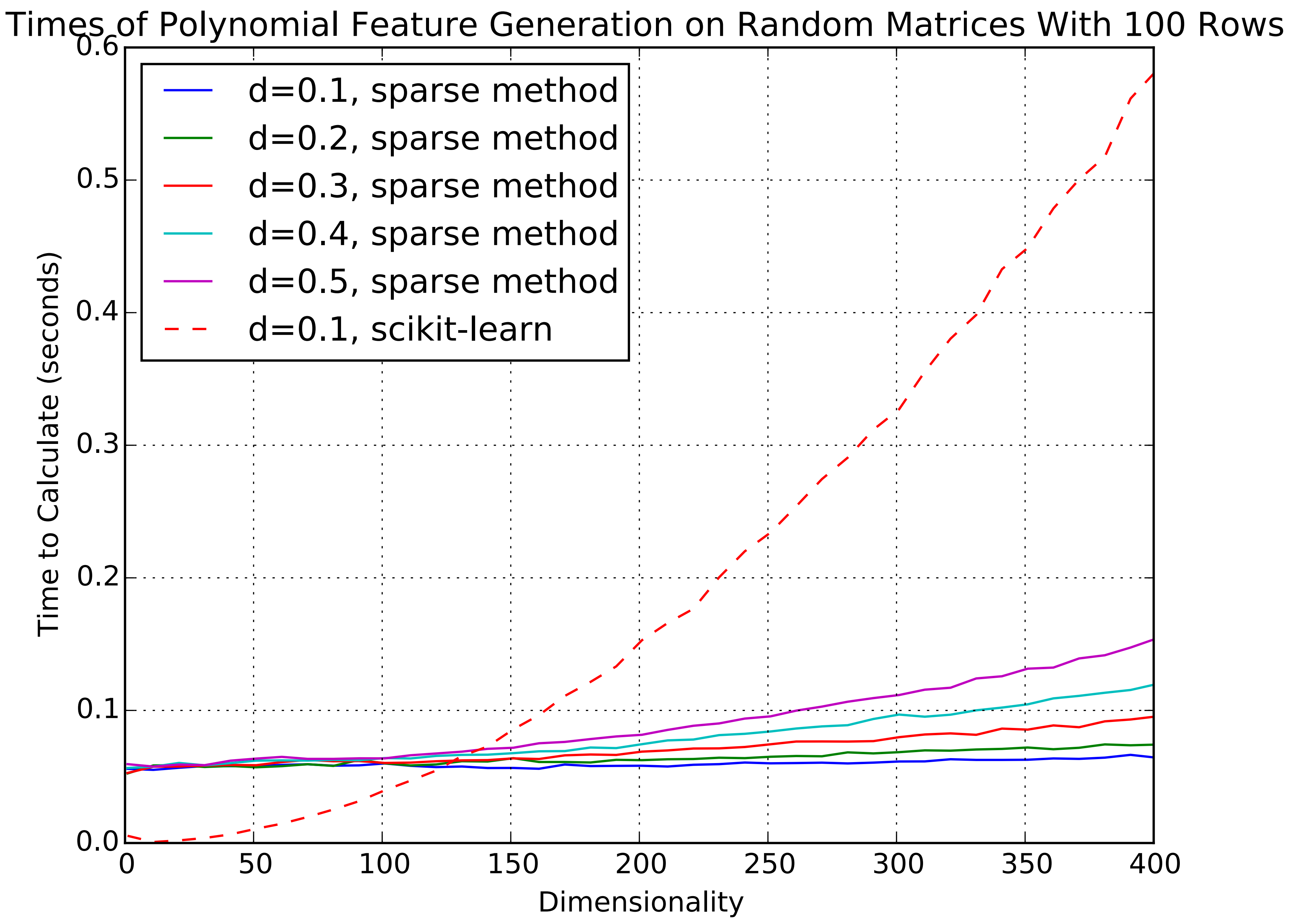
$\textsc{Sparse Polynomial Features}(A)$

$\text{PolyMap}(a,b) = \frac{2Da - a^2 + 2b - 3a - 2}{2} + a + 1$
N = row count of A
D = column count of A
B = Compressed Sparse Row Matrix of size $N \times \frac{D^2+D}{2}$
for row in A
  N_{zc} = nonzero columns of row
  for $i = 0$ **to** $|N_{zc}|$
    for $j = i$ **to** $|N_{zc}|$
      $k = \text{PolyMap}(N_{zc}[i], N_{zc}[j])$
      r = index of row
      $B[r,k] = row[N_{zc}[i]] \cdot row[N_{zc}[j]]$

$O(ND^2)$, $\Theta(Nd^2D^2)$
where d is density

Average complexity decreases quadratically w.r.t. density!

How fast is it?



Real Usage

dataset	instances	features	polynomial features	density	dense space	sparse space	dense time	sparse time
20 Newsgroups	11,314	130,107	8,463,980,778	0.12	scikit-learn: MemoryError	5333 MB	scikit-learn: MemoryError	109 s
Connect Four	67,557	126	8,127	0.11	4191 MB	735 MB	26 s	44 s

Code available on github.com/AWNystrom