Data-dependent Performance Modeling of Linear Solvers for Sparse Matrices

Jae-Seung Yeom† Jayaraman Thiagarajan† Abhinav Bhatele†
Greg Bronevetsky‡ Tzanio Kolev†

†{yeom2,jayaramanthi1,bhatele1,kolev1}@llnl.gov
‡bronevet@google.com

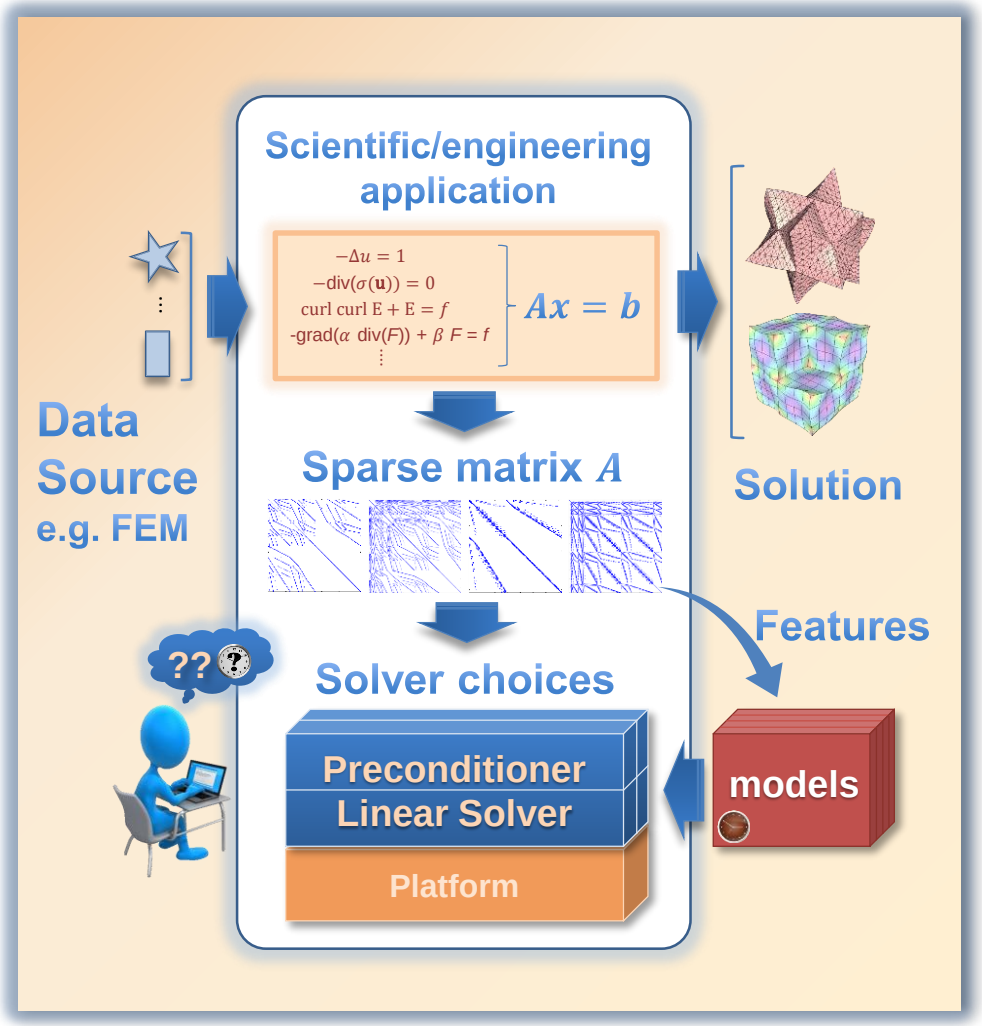
LLNL-PRES-689420

DOE Centers of Excellence Performance Portability Meeting, April 19-21, 2016

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

Overview of the problem



- Many computational science and engineering (CSE) code rely on solving sparse linear systems
- Performance of a linear solver depends on
  - input problem
  - data representation
  - algorithmic
  - Implementation
  - platform
- Need to choose an optimal solver for a given problem

Problem setup

- Given an input matrix and a set of PC-LS choices,
  - predict the fastest choice to solve the linear problem for novice users
- Two sources of matrix data
  - MFEM data: representative of PDE-based simulations
    - E.g., heat transfer, structural mechanics, accelerator design, radiation diffusion flow,
  - University of Florida (UFL) data : collected from various domains
- PC-LS choices from *Trilinos* (ver. 11.12.1): Aztec + Ifpack

Matrix data N_M

MFEM	UFL
879 matrices generated for PDE-based problems	361 chosen matrices

PC-LS choices $N_C = N_P * N_S$

Preconditioners	Linear solvers
NONE, JACOBI, NEUMANN, LS, DOM_DECOMP, ILU, ILUT, IC, ICT, CHEBYSHEV, POINT_RELAXATION, BLOCK_RELAXATION, AMESOS, SORA, IHSS	GMRES, GMRES_ CONDNUM, GMRESR, CG, CG_CONDNUM, CGS, BICGSTAB, TFQMR, FIXED_PT

Candidate model features
to understand prediction capability

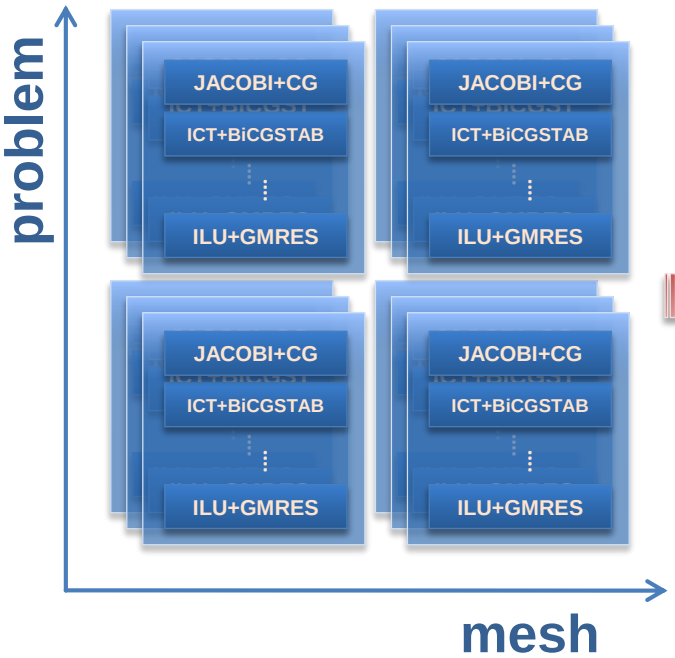
N_{row}	number of rows
NNZ	number of non-zeros, $nnz(A)$
NNZ_L	number of non-zeros in lower triangle
NNZ_U	number of non-zeros in upper triangle
NNZ_{max}	maximum number of non-zeros per row
NNZ_{avg}	average number of non-zeros per row
DE_{max}	largest element in magnitude along diagonal
DE_{min}	smallest non-zero element in magnitude along diagonal
bw_L	bandwidth of lower triangle
bw_U	bandwidth of upper triangle
NO	number of ones
NDD_{row}	number of rows that are strictly diagonally dominant, i.e., satisfying $2 * \|a_{j,j}\| - \sum_j \|a_{i,j}\| > 0$
$spread_L$	normalized sum of distances from diagonal of non-zero elements in lower triangle $\sum (i-j)/S$ where $i > j$, $a_{i,j} \neq 0$, and $S = \sum_{i=1}^{N_{row}-1} i * (N_{row} - i)$
$spread_U$	normalized sum of distances from diagonal of non-zero elements in upper triangle $\sum (j-i)/S$ where $j > i$, $a_{i,j} \neq 0$, and $S = \sum_{i=1}^{N_{row}-1} i * (N_{row} - i)$
$symm$	property showing how symmetric the matrix is as $1 - nnz(A - A.^T)/(nnz(A) - nnz_d(A))$ where $A.^T$ is a non-conjugate transpose of A and $nnz_d(A)$ is number of non-zeros on diagonal of A
$symm_s$	property showing how skew-symmetric the matrix is as $1 - (nnz(A + A.^T) - nnz_d(A))/(nnz(A) - nnz_d(A))$
$symm_p$	property showing how symmetric the non-zero pattern of matrix is as $1 - nnz(P - P.^T)/(nnz(P) - nnz_d(P))$ where $p_{i,j} = 1$ if $a_{i,j} \neq 0$, 0 otherwise
$\|\|A\|\|_1$	1-norm
$\|\|A\|\|_F$	Frobenius norm

$\rho(A)$	spectral radius
λ_2	second largest absolute eigenvalue
λ_{min}	non-zero smallest absolute eigenvalue
$\sqrt{\lambda_1/\lambda_2}$	where $\lambda_1 = \rho(A)$
$\sqrt{peak}$	$peak = \lambda_1 / ((\sum_i \lambda_i) - \lambda_1)$
$\sqrt{\kappa}$	where κ is the condition number computed by Matlab
s_{90}	portion of $\sqrt{\sigma} > 0.9 \times (\sqrt{\sigma_{max}} - \sqrt{\sigma_{min}}) + \sqrt{\sigma_{min}}$, where σ is a singular value
s_{mid}	$1 - (s_{90} + s_{10})$
s_{10}	portion of $\sqrt{\sigma} \leq 0.1 \times (\sqrt{\sigma_{max}} - \sqrt{\sigma_{min}}) + \sqrt{\sigma_{min}}$
N_{dim}	number of spatial dimensions (from the mesh)
N'_{row}	N_{row}/N_{dim} if P2, N_{row} otherwise
$Mesh_{do}$	finite element discretization order
$Mesh_{rl}$	mesh refinement level
$\sqrt{\kappa'}$	approximation to condition number term $\sqrt{\kappa}$, where $\kappa' = (N'_{row})^{2/N_{dim}}$
$\sqrt{\kappa'_{hmin}}$	approximation to condition number term $\sqrt{\kappa}$, where $\kappa'_{hmin} = (1/min(h))^2$
$\sqrt{\kappa'_{hmax}}$	approximation to condition number term $\sqrt{\kappa}$, where $\kappa'_{hmax} = (1/max(h))^2$

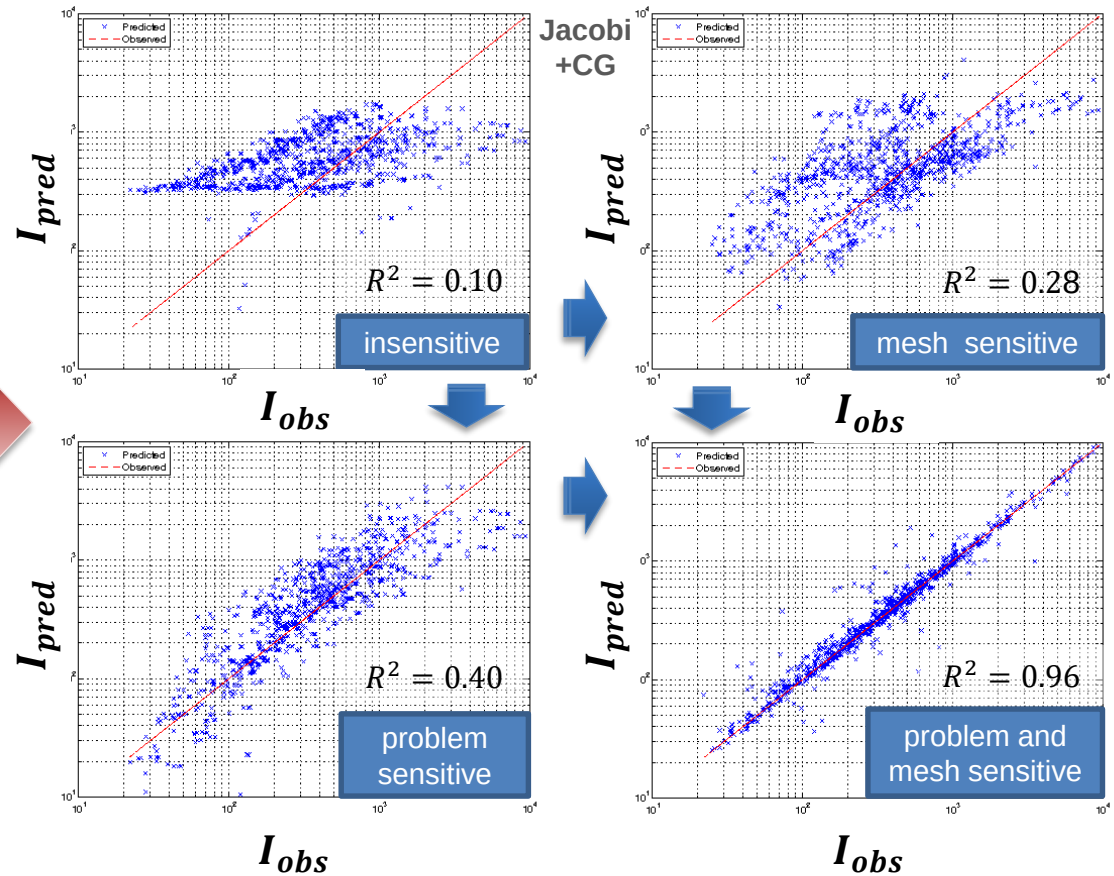
- **Basic**: readily computable
- **Advanced**: expensive to compute
- **Domain-specific:** expert knowledge

Challenges in data and initial attempt
with linear regression to predict execution time

- Identifying/designing important features
- Too many branching on categorical features → not scalable/not general
- Highly skewed distribution
- System noise



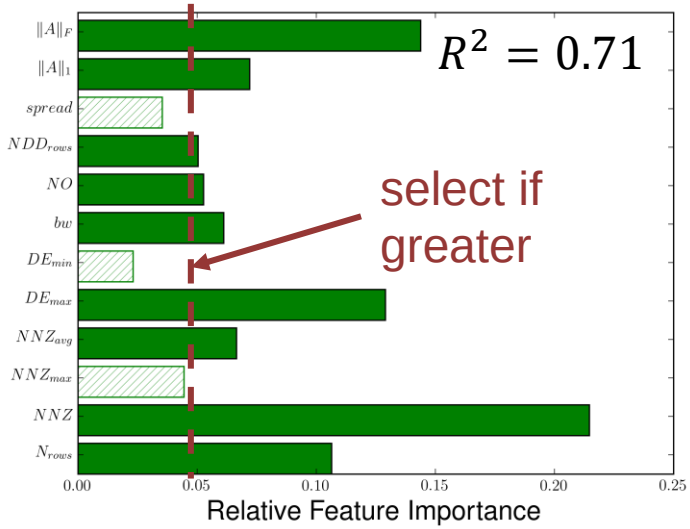
$$N_{runs} = N_M * N_C * 11 * 10$$



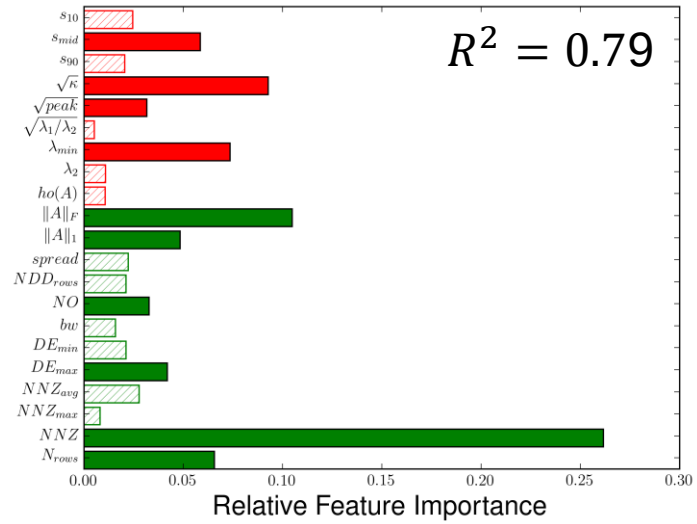
Selecting features based on relative importance
identified by Gradient Boosted Machine (GBM)

4 sets to understand the impact of feature choices

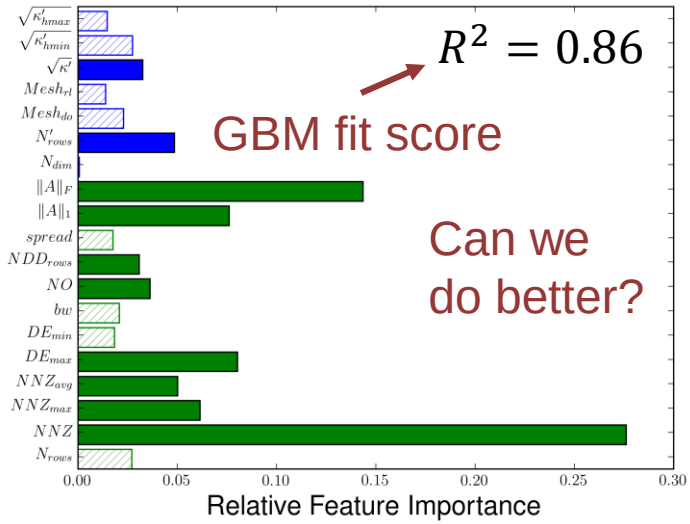
F1 Basic



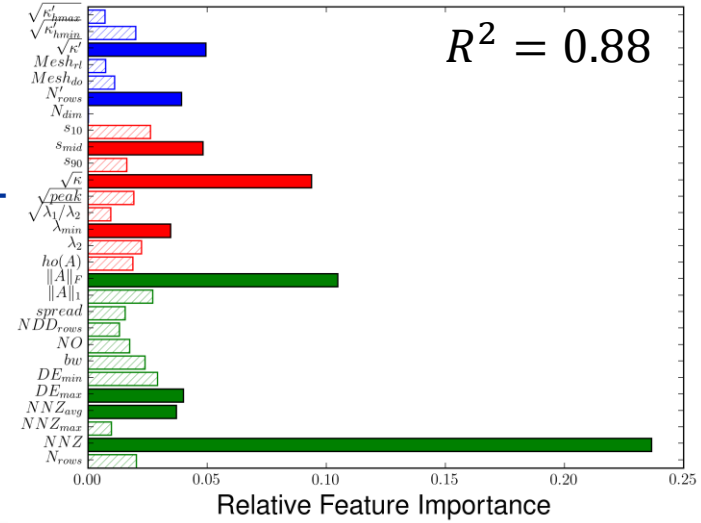
F2 Basic + Advanced



F3 Basic + Domain-specific

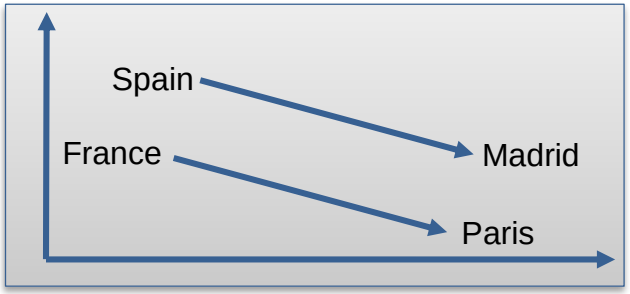


F4 Basic + Advanced +Domain-specific

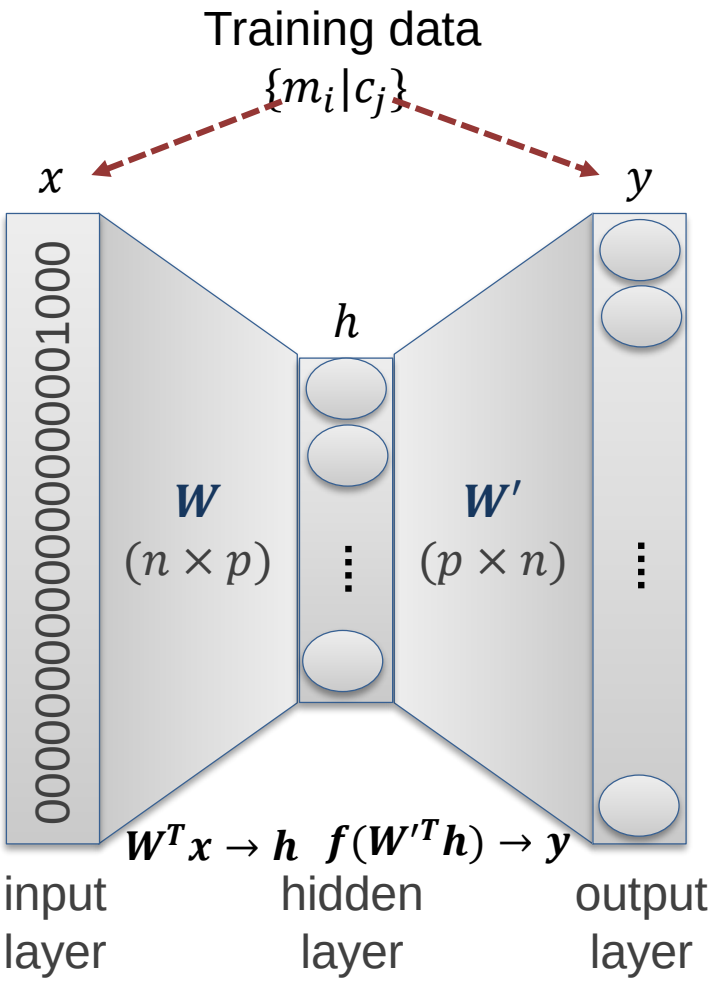


Neural word embedding

- Based on neural network
- Low-dimensional representation
  - Vector space W
- More similar words closer in space
- Preserves many linguistic regularities and patterns
- *Word2Vec* (Mikolov et al., '13)
  - Train 100 billion words in a day

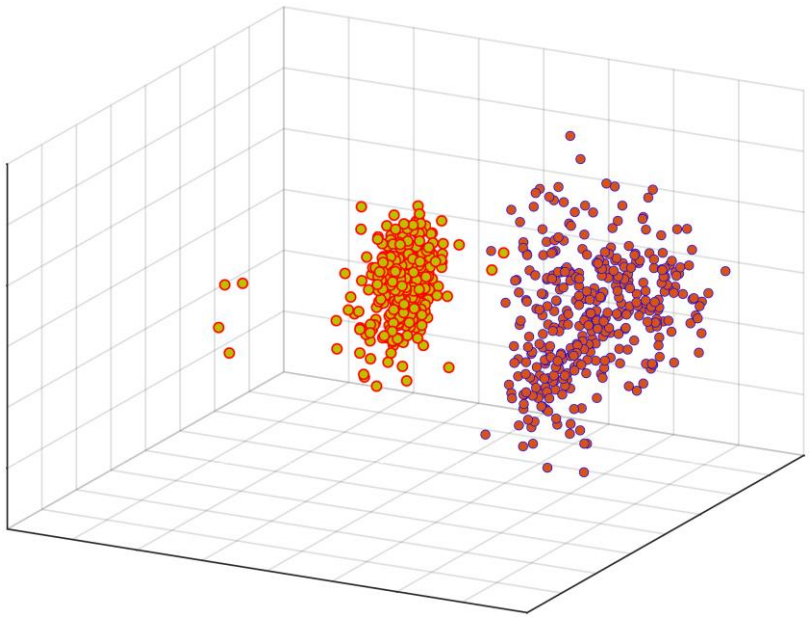


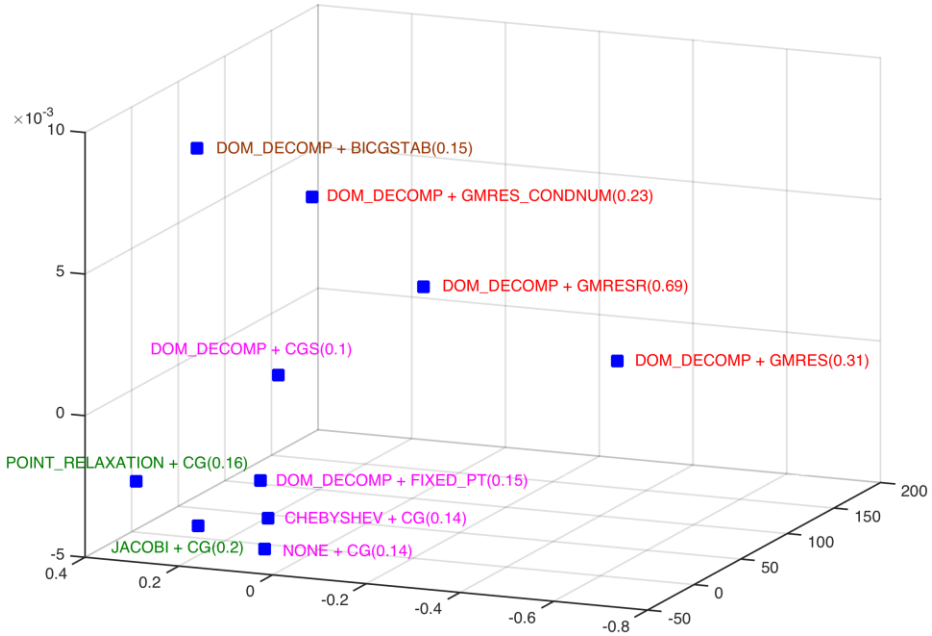
vec("Madrid") - vec("Spain") + vec("France") ≈ vec("Paris")



$$n = N_M + N_C$$

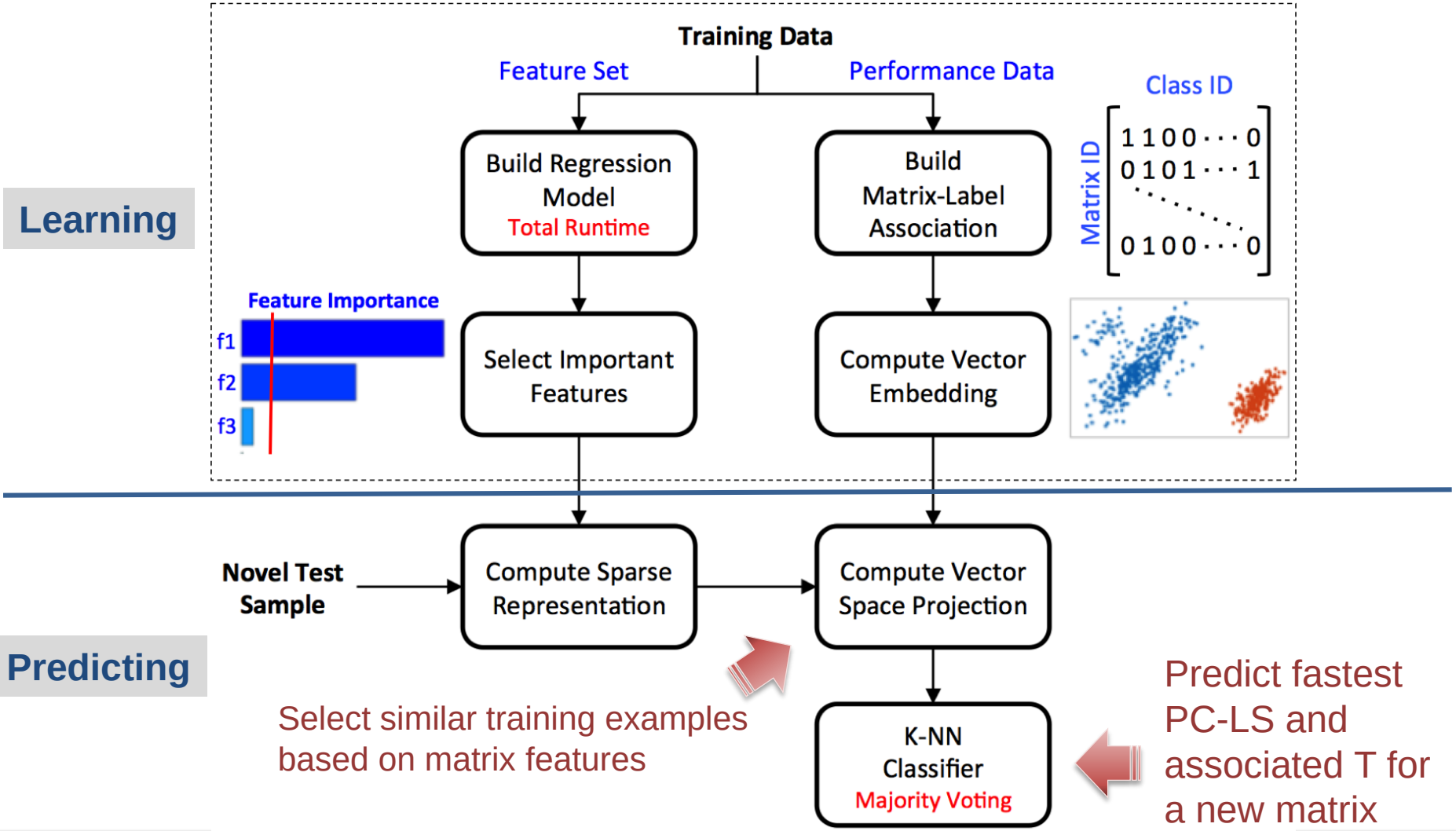
Modeling approach
performance vector space by *word2vec*





- Performance vector space (PVS)
  - Weight matrix (W) learned between input layer and hidden layer
- To represent matrix relation in a lower dimension space
- Map from PVS to label
  - Weight matrix (W') learned between hidden layer and output layer
- To determine a set of PC-LSs that are effective for the given matrix

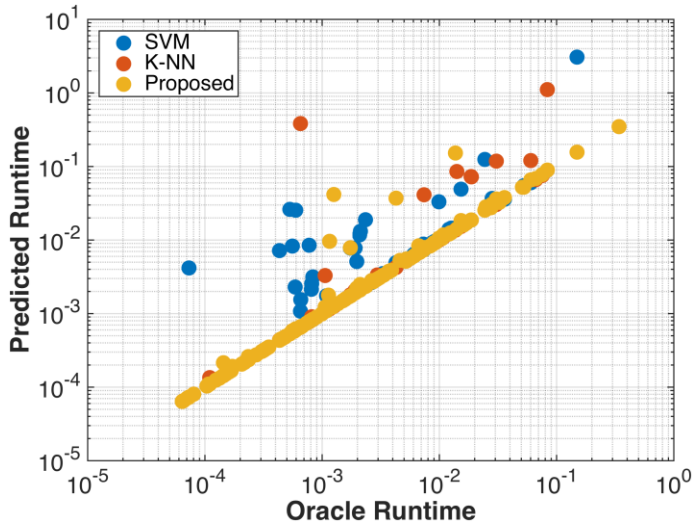
Overall workflow



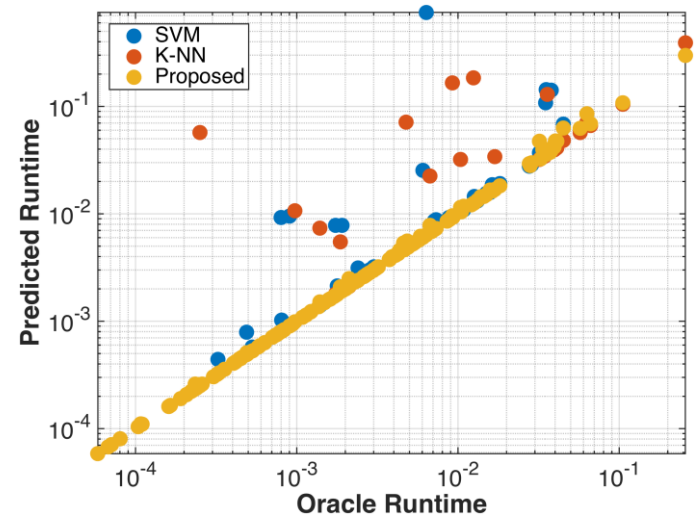
Predicting execution time for MFEM matrices

predicted vs. observed

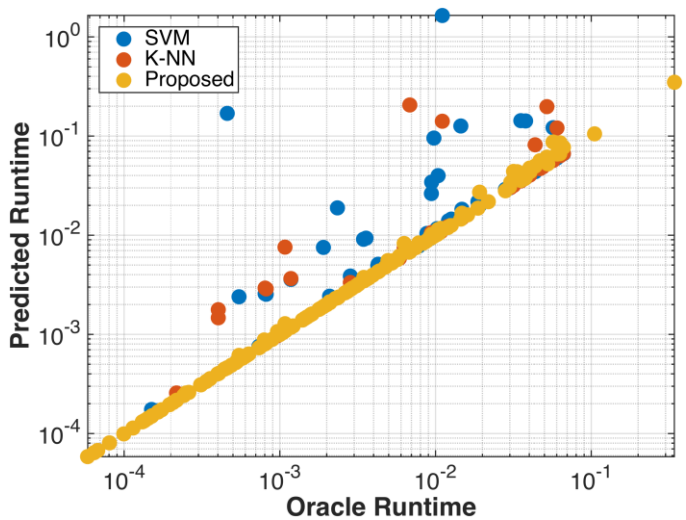
F1 Basic



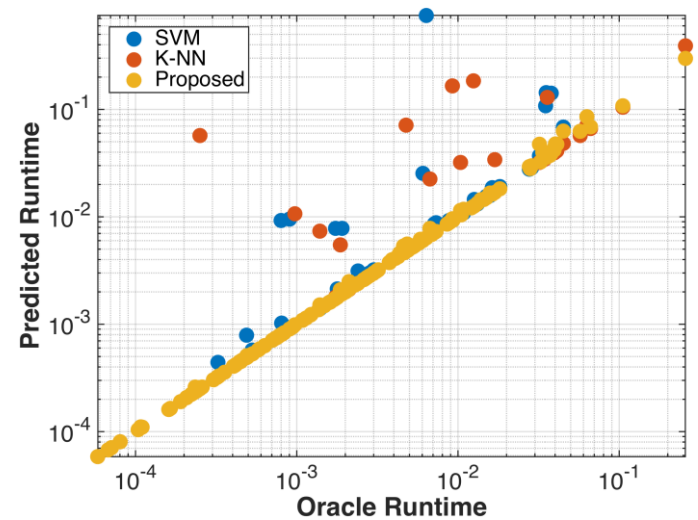
F2 Basic + Advanced



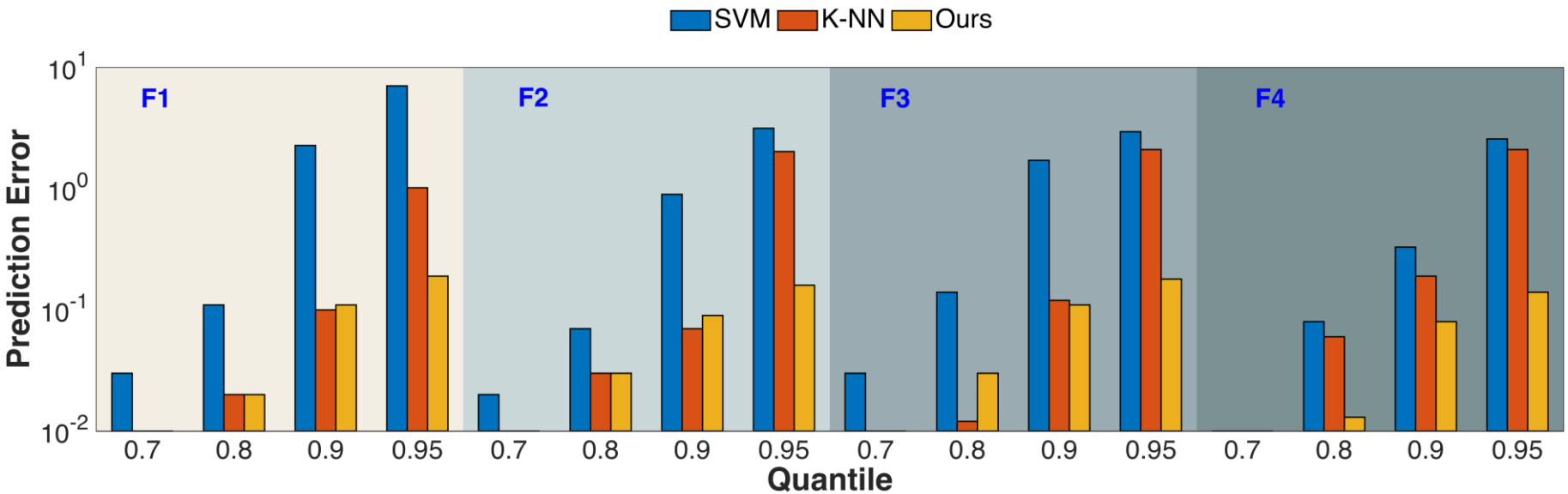
F3 Basic + Domain-specific



F4 Basic + Advanced +Domain-specific



Prediction error (ARE)



Impact of ineffective outlier handling in classification

y_p: predicted time

y_o: observed time

$$E = \frac{|y_p - y_o|}{y_o}$$

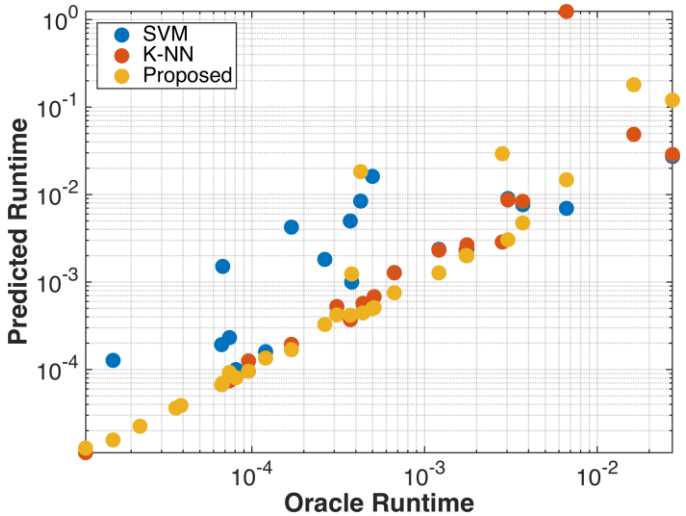
Absolute Relative Error (ARE)

Mean errors

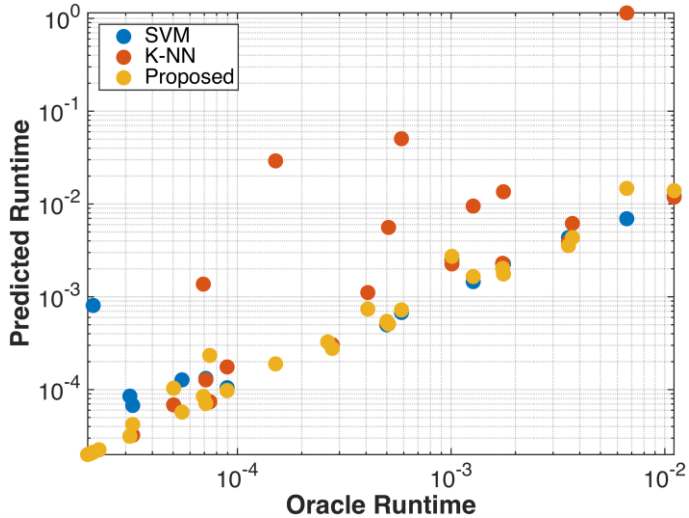
Feature set	F1	F2	F3	F4	Blind
SVM	1.58	2.08	3.42	2.06	
K-NN	3.66	1.68	0.39	1.76	0.14
Ours	0.18	0.025	0.03	0.025	

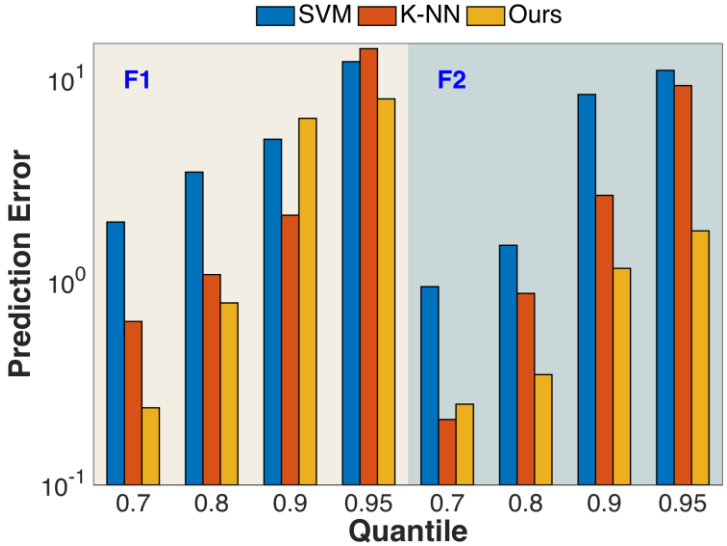
Predicting execution time for UFL matrices
predicted vs. observed

F1 Basic



F2 Basic + Advanced





Mean errors

Feature set	F1	F2	Blind
SVM	3.52	2.93	
K-NN	3.41	2.69	2.19
Ours	2.13	0.8	

User-centered metric

ARE vs. Classification accuracy (%)

- Our method outperforms in terms of absolute relative error (ARE)
  - in the presence of outlying data
  - Even when classification accuracy is lower than SVM and k-NN
- This helps users to identify optimal or close to optimal PC-LS choices

MFEM

ARE

Feature set	F1	F2	F3	F4	Blind
SVM	1.58	2.08	3.42	2.06	0.14
K-NN	3.66	1.68	0.39	1.76	
Ours	0.18	0.025	0.03	0.025	

Classification accuracy (%)

Feature set	F1	F2	F3	F4	Blind
SVM	73.4	75.8	76.0	80.1	69
K-NN	72.8	75.1	74.6	78.6	
Ours	70.1	74.2	75.2	77.9	

Conclusion

- Data-dependent performance modeling
- Highly challenging dataset
  - Large number of classes, categorical features, limited amount, skewed label distribution, outlying data
- Formulated the problem as a classification problem of predicting the fastest PC-LS choice for a given matrix
  - Neural word embedding to take advantage of patterns in relationship
    - Between matrices, and between matrices and PC-LS choices
  - Consistent feature space and performance vector space (PVS)
- User-centered metric
  - Outperforms in the presence of outlying data in terms of absolute relative error (ARE)

Lawrence Livermore
National Laboratory