Inferring Origin Flow Patterns in Wi-Fi with Deep Learning

Youngjune Gwon H. T. Kung

11th International Conference on Autonomic Computing (ICAC'14)

Philadelphia, PA

June 18, 2014

Outline

- Introduction
- Background
- Origin flow pattern inference in Wi-Fi
- Classical approaches
- Our approach
- Evaluation
- Conclusion

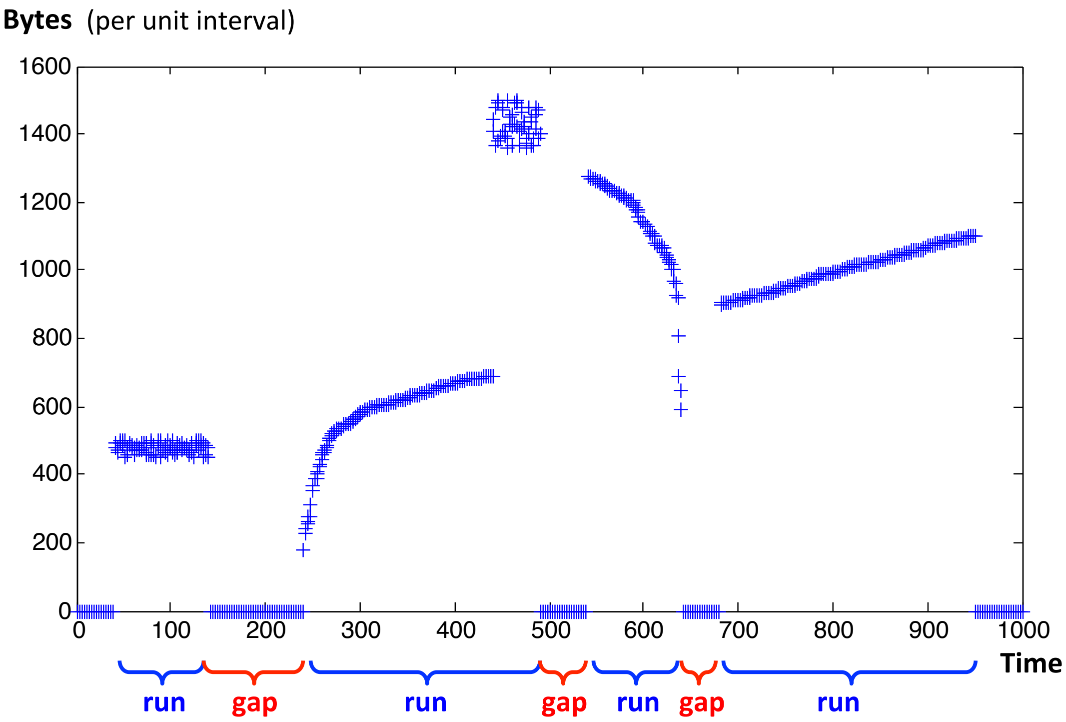
What Is Network Traffic Inference?

- Network traffic analysis is classical research topic
  - Study, measure, and estimate flow characteristics
    - *E.g.*, burst size and interarrival time distributions, mean values
  - Network nodes (routers) regularly **sample** packets
    - To provide data used for analysis

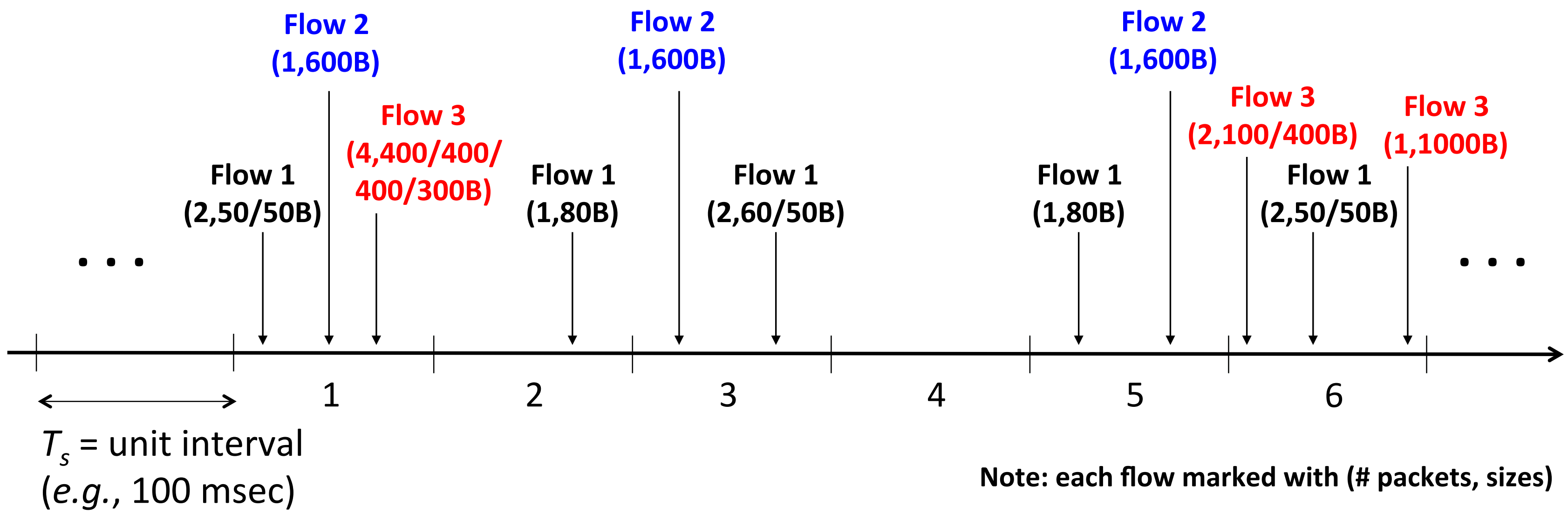
- Why?
  - Traffic monitoring
    - Spot anomalies, (D)DoS attacks, heavy hitters
  - Help manage networking resources
    - Wireless spectrum among most precious networking resources
  - Program network nodes (SDN)
    - Improve Tx-Rx scheduling, interference mitigation

Flow Pattern

- Sequence of data bytes (*run*) with waiting times (*gap*)
- **Runs-and-gaps model**
  - Flow pattern ⟹ ***time series data***
    - Simple, but powerful abstraction
  - Applicable at any node (src, dst, intermediate)

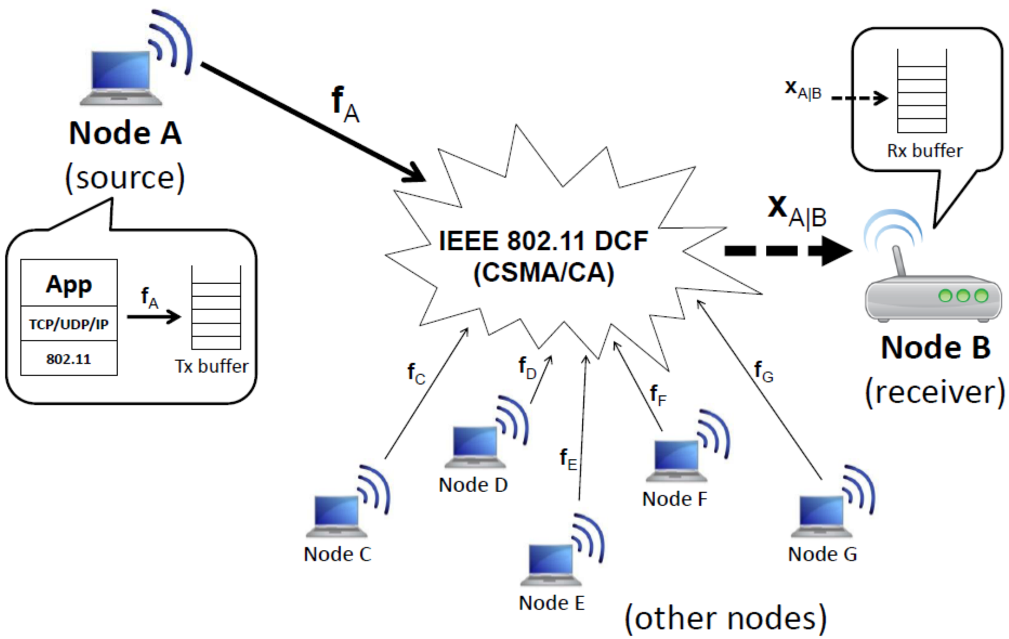


Runs-and-gaps Time Series Processing



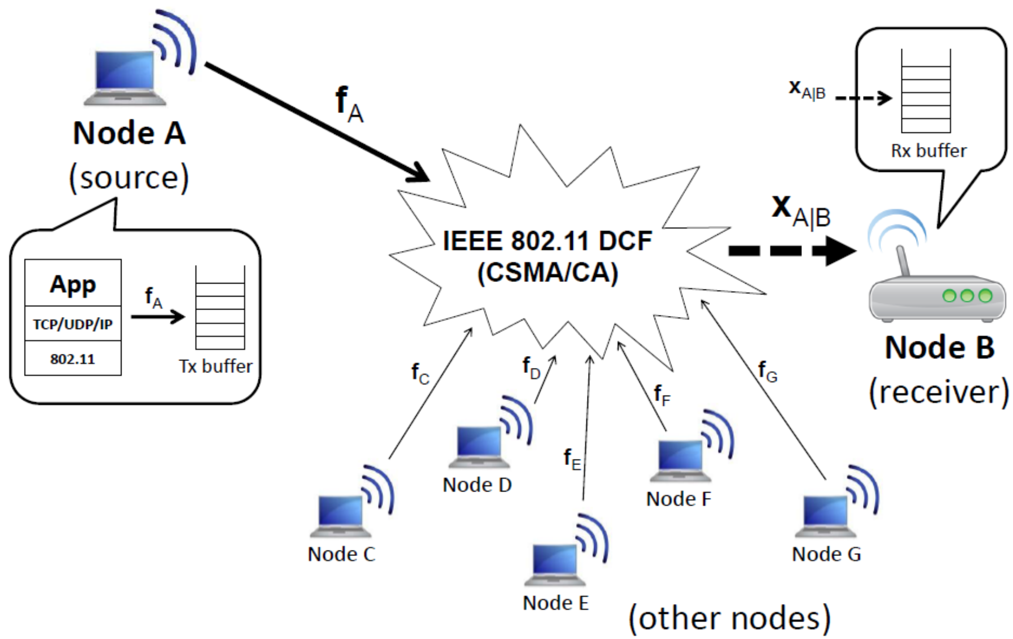
- Flow 1
  - $\mathbf{w}_1$ = [2 1 2 0 1 2], $\mathbf{x}_1$ = [100 80 110 0 80 100]
- Flow 2
  - $\mathbf{w}_2$ = [1 0 1 0 1 0], $\mathbf{x}_2$ = [600 0 600 0 600 0]
- Flow 3
  - $\mathbf{w}_3$ = [4 0 0 0 0 3], $\mathbf{x}_3$ = [1500 0 0 0 0 1500]

Origin Flow Pattern Inference in Wi-Fi (1)



- Origin flow pattern (**f**)
  - Conveys application-level data generation context
  - As entering source Tx buffer
- Measured flow pattern (**x**)
  - At best, **x** = *time-shifted* **f**
  - Reflects severity of congestion/mix with other flows
  - As timestamped at receiver Rx buffer

Origin Flow Pattern Inference in Wi-Fi (2)



- **Problem:** how to accurately infer origin flow pattern $\mathbf{f}_A$ from received pattern $\mathbf{x}_{A|B}$?
  - Key challenge: CSMA alters origin pattern by introducing complex, irregular mixture of competing flows
  - Bottomline: ***multiclass classification problem***

Approaches (Classical)

- Supervised learning
  - ARMAX
    - AR = delayed ground truth patterns ($\mathbf{f}$)
    - MA = model error ($\boldsymbol{\varepsilon}$)
    - X = delayed received patterns ($\mathbf{x}$)
    - Train $\underline{\mathbf{f}}_t = [\mathbf{f}_{t-1} \ldots \mathbf{f}_{t-n}\ \mathbf{x}_{t-1} \ldots \mathbf{x}_{t-m}\ \boldsymbol{\varepsilon}]\boldsymbol{\theta}$ with labeled dataset $\{\mathbf{x}^{(i)}, <\mathbf{f}^{(i)}, l^{(i)}>\}$
      - Estimate $\boldsymbol{\theta}$ via least squares (recursive LS by Kalman filtering)
  - Naïve Bayes classifier
    - Using feature $\mathbf{y} = [\mu_{\text{run}}\ \mu_{\text{gap}}]$ for given $\mathbf{x}$
    - Train $p(l|\mathbf{y}) \propto p(\mathbf{x}|l)$ from with $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, l^{(i)}\}$

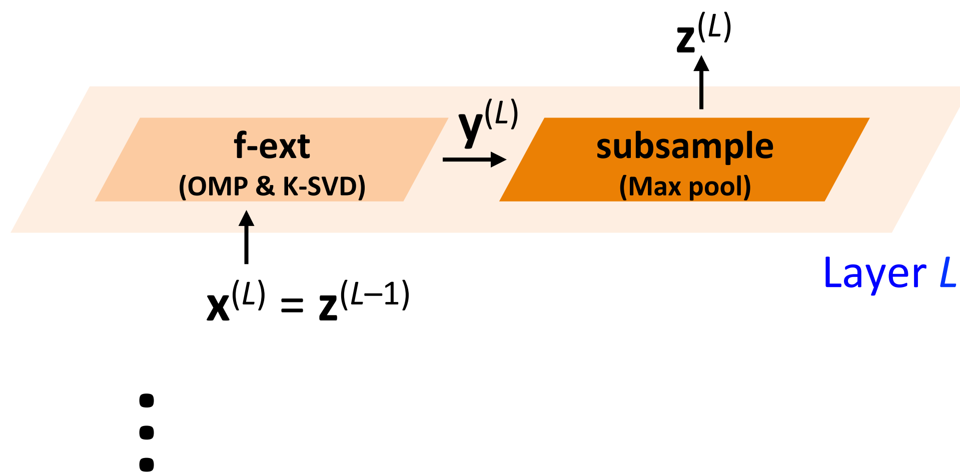
- Semi-supervised learning
  - Gaussian mixtures
    - Use same feature, bivariate $\mathbf{y} = [\mu_{\text{run}}\ \mu_{\text{gap}}]$ for given $\mathbf{x}$
    - Train K-Gaussian sum $\sim \{\mathbf{w}, (\boldsymbol{\mu}, \boldsymbol{\Sigma})\}$ via EM with $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}$ (*unsupervised*)
      - $\mathbf{w}$ = mixing weights, $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ = Gaussian parameters
    - Classification: use SVM (*supervised*)
      - Train with posterior (membership) probabilities with $\{\mathbf{x}^{(i)}, <\mathbf{f}^{(i)}, l^{(i)}>\}$

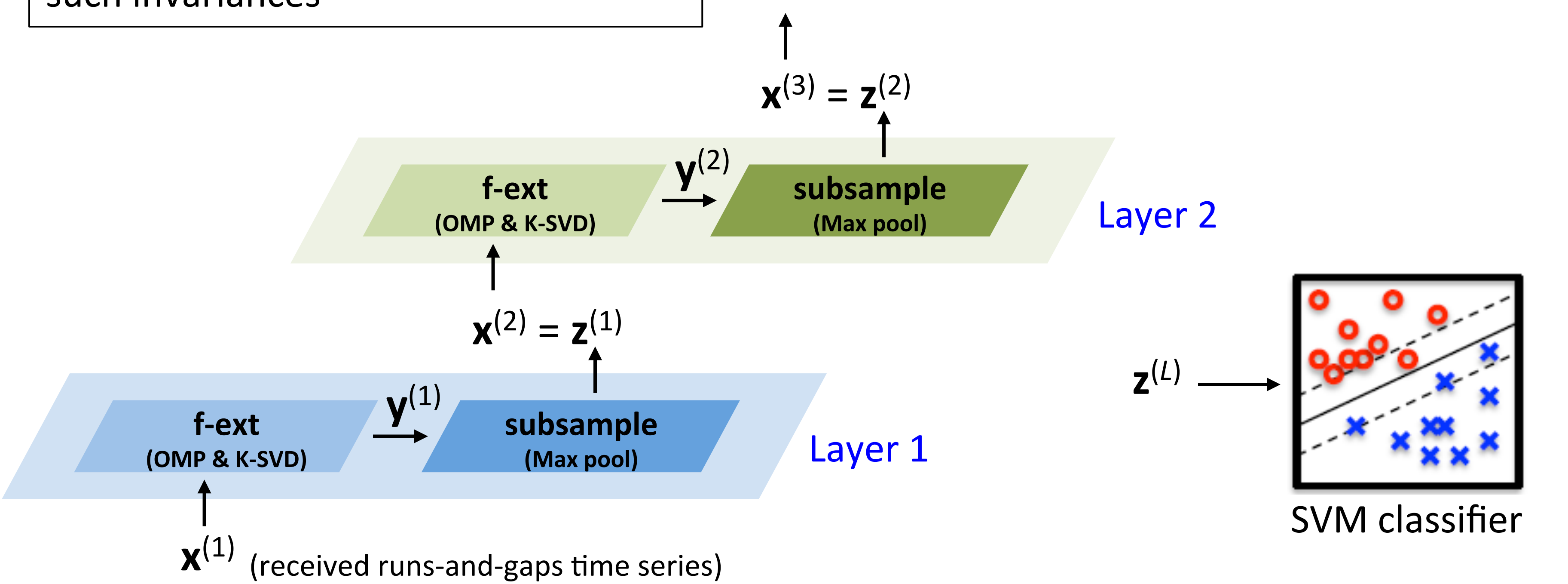
Our Approach

- Semi-supervised learning
  - Phase I: unsupervised feature learning
    1. Sparse coding & dictionary learning (*unlabeled* **x**'s)
    2. Subsample features via (max) pooling
    3. Repeat for multiple layers (feed current layer's result as next layer's input)
  - Phase II: supervised classifier training
    1. Do multi-layer sparse coding and pooling with *labeled* **x**'s
    2. Train SVM classifiers with final feature vector resulted at top

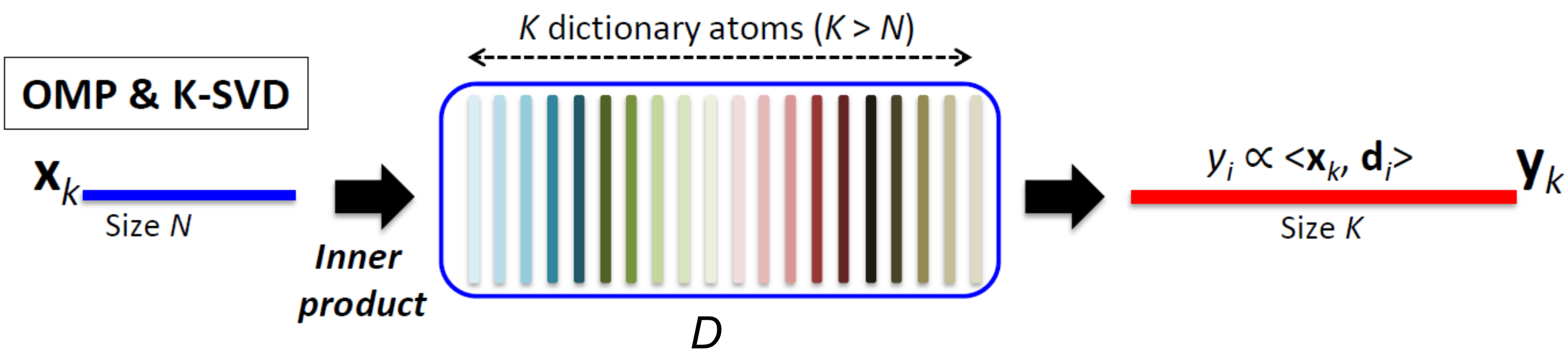
Multi-layer Feature Learning and SVM Classification

CSMA spreads flow invariances (some preserved original run lengths) over long period $\Longrightarrow$ feature learning & pooling over multiple layers identify such invariances



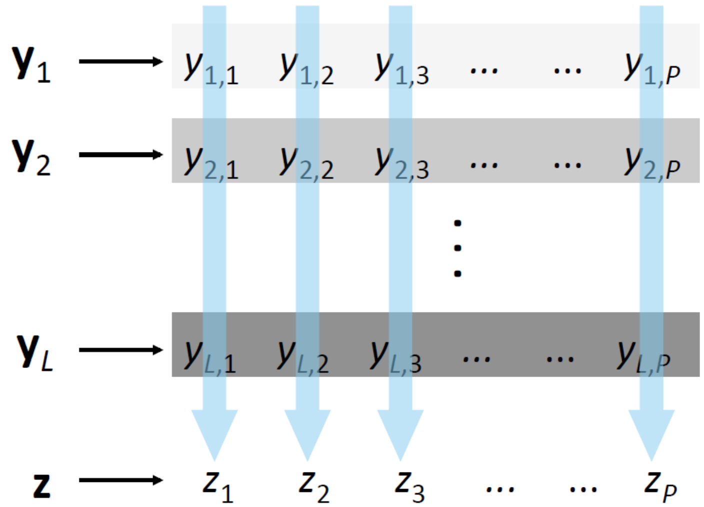


What Is Sparse Coding?



- Describe input **x** as *M* linear combination of *D*'s columns
- $\mathbf{x} = D\mathbf{y}$
  - $\mathbf{x}$ = measured flow pattern
  - $\mathbf{y}$ = extracted feature from $\mathbf{x}$
  - OMP computes $\mathbf{y}$ & K-SVD trains D
    - $\min \|\mathbf{X} - D\mathbf{Y}\|_F^2$ s.t. $\|\mathbf{y}_k\|_0 \leq M \ \forall k$
  - Sparsity: $M \ll N < K$
- Sparse coding, clustering, and mixtures are fundamentally same idea

What Is Max Pooling?



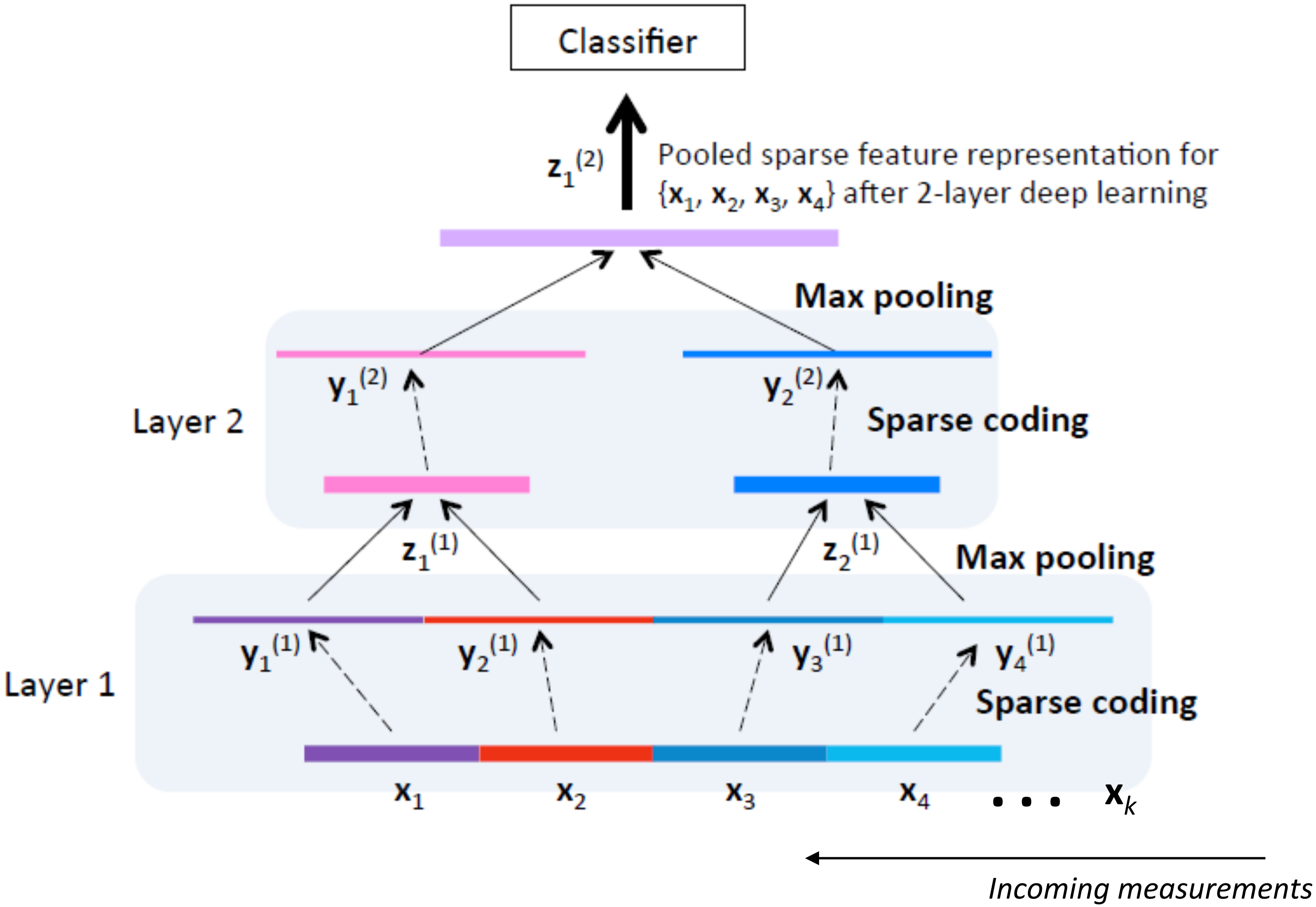
$z_1 = \max(y_{1,1}, y_{2,1}, \ldots, y_{L,1})$

$z_2 = \max(y_{1,2}, y_{2,2}, \ldots, y_{L,2})$

$\vdots$

- What do we do when we have too many of same kinds?
  - Need to summarize over them
- Max pooling
  - Translation-invariant subsampling of multiple feature vectors
  - Popular in CNN for image recognition

Summarizing Deep Feature Learning



Enhancements

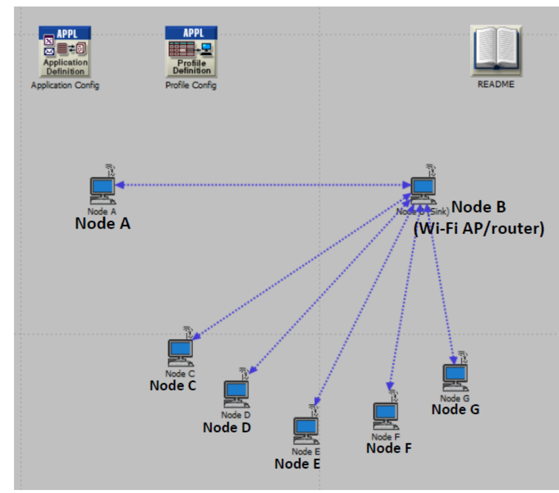
- Incoherent dictionary atoms
  - Force: $\|D^{\mathrm{T}}D\| = I$ with new constraint
    - $\min \|\mathbf{X} - D\mathbf{Y}\|_F^2 + \gamma \|D^{\mathrm{T}}D - I\|_F^2$ s.t. $\|\mathbf{y}_k\|_0 \leq M' \ \forall k$

- Relax sparsity due to distortions resulted by incoherent dictionary training
  - Use $M' > M$ for OMP

- Overlapping max pooling
  - $\mathbf{z}_1 = \texttt{max\_pool}(\mathbf{y}_1, \ldots, \mathbf{y}_L)$, $\mathbf{z}_2 = \texttt{max\_pool}(\mathbf{y}_5, \ldots, \mathbf{y}_{L+4})$, …
    - Instead of $\mathbf{z}_2 = \texttt{max\_pool}(\mathbf{y}_{L+1}, \ldots, \mathbf{y}_{2L})$, …

Evaluation

- Simulated 7 Wi-Fi nodes in OPNET Modeler
  - 10 distinct flow patterns generated at source
    - Mixed with various other flows including RTP/UDP/IP, HTTP, ftp, interactive DB transactions



- Schemes
  - ARMAX
  - Naïve Bayes
  - GMM with $K = 10$ & linear 1-vs-all SVMs
  - Proposed baseline
    - 2 layers & linear 1-vs-all SVMs
  - Proposed baseline + 3 enhancements
  - Implemented in MATLAB

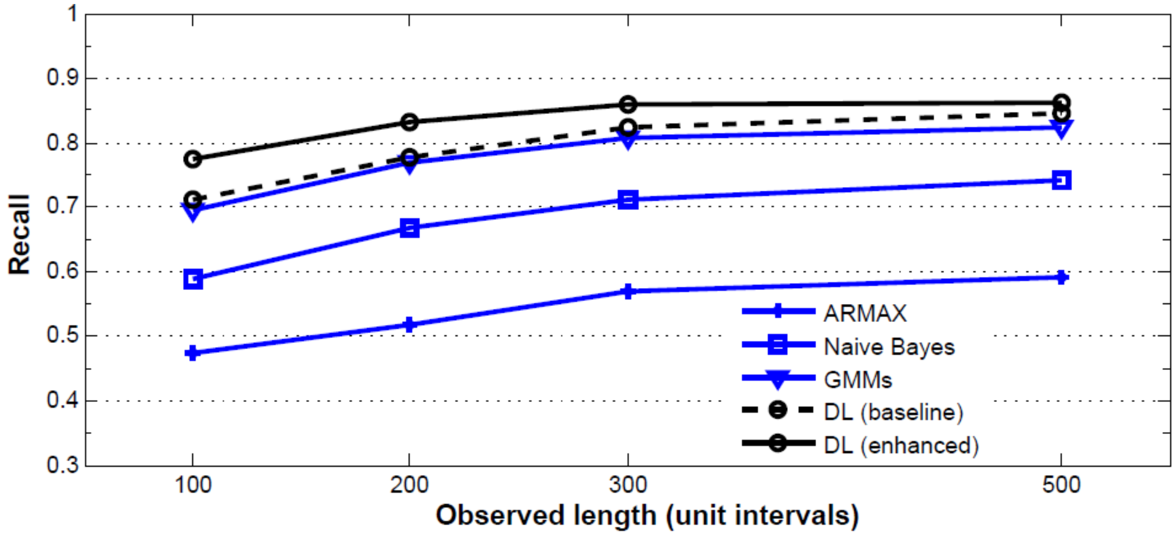
- Metrics
  - Classification recall (true positive rate) and false alarm rate

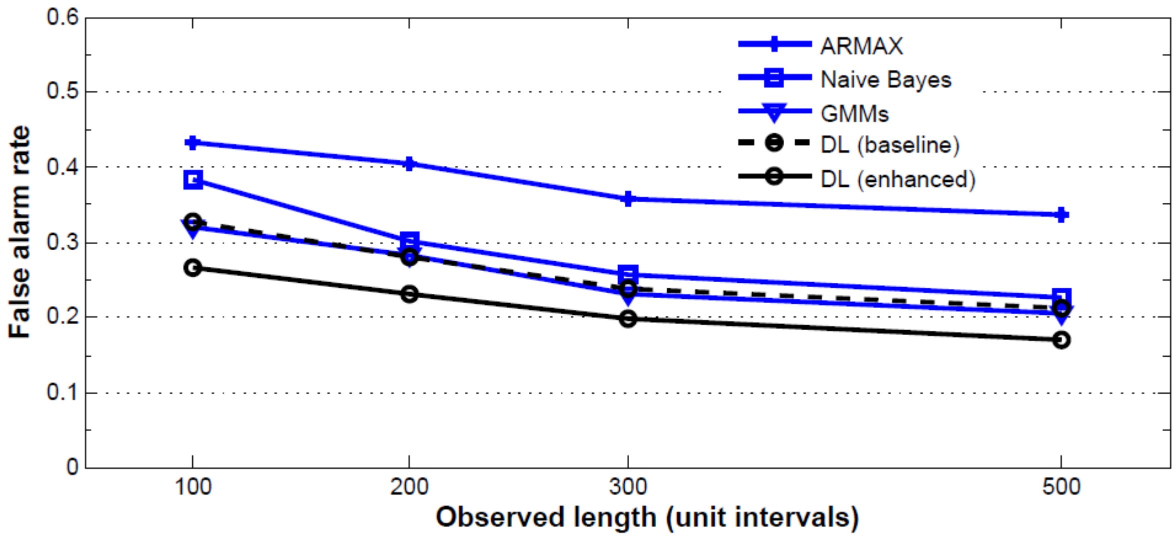
Flow Patterns and Nodes

Pattern	Flow type	Generative triplet $\langle t_r, s_r, t_g \rangle$
$\mathbf{f}_1$	Constant	$\langle 2, 100, 4 \rangle$
$\mathbf{f}_2$	Constant	$\langle 2, 500, 2 \rangle$
$\mathbf{f}_3$	Constant	$\langle 5, 200, 5 \rangle$
$\mathbf{f}_4$	Constant	$\langle 10, 200, 10 \rangle$
$\mathbf{f}_5$	Stochastic	$\langle \mathrm{Exp}(1), \mathrm{Pareto}(100, 2), \mathrm{Exp}(0.1) \rangle$
$\mathbf{f}_6$	Stochastic	$\langle \mathrm{Exp}(0.5), \mathrm{Pareto}(40, 1), \mathrm{Exp}(0.25) \rangle$
$\mathbf{f}_7$	Stochastic	$\langle \mathrm{U}(4, 10), \mathrm{Pareto}(100, 2), \mathrm{Exp}(0.5) \rangle$
$\mathbf{f}_8$	Stochastic	$\langle \mathrm{N}(10, 5), \mathrm{Pareto}(40, 1), \mathrm{N}(10, 5) \rangle$
$\mathbf{f}_9$	Mixed	$\langle 1, \mathrm{Pareto}(100, 2), 1 \rangle$
$\mathbf{f}_{10}$	Mixed	$\langle 1, \mathrm{Pareto}(100, 2), \mathrm{Exp}(0.25) \rangle$

Node	Role	Main networking activity
A	Flow source	Transmits $\mathbf{f}_i$
B	Receiver	Intercepts flows as Wi-Fi router/AP
C	Flow source	Transmits $\mathbf{f}_j \; \forall \; j \neq i$
D	Flow source	Multimedia streaming over RTP/UDP/IP
E	Flow dest.	HTTP with page size $\sim$ U[10,400] B
F	Flow dest.	ftp file transfer with size 50000 B
G	Flow dest.	DB access with inter-arrival $\sim$ Exp(3) sec

Classification Performance





Burst and Interarrival Prediction Errors

Scheme	Origin run size prediction error	Origin gap size prediction error
ARMAX	45.9%	36.7%
Naïve Bayes	37.5%	24.6%
GMM (K = 10)	31.3%	18.1%
Proposed (baseline)	28.3%	16.2%
Proposed (enhanced)	22.8%	11.4%

Conclusion

- Simply, we have created *inverse mapping*
  - Measured pattern ⟶ origin pattern (prequalified)
  - This mapping consists of deep feature learner & classifier

- Deep learning
  - Start with small features, aggregate up, and broaden coverage
  - Can learn invariances and changes introduced by CSMA
    - Arbitrary mix of flows, retransmissions, loss of data

- Future directions
  - Explore other (dis)similarity metrics (*e.g.*, DTW)
  - Sparse packet sampling, multiple hops
  - Test on real Wi-Fi data
  - Other inference applications in networking (*e.g.*, protocols)

Backup Slides

Metrics

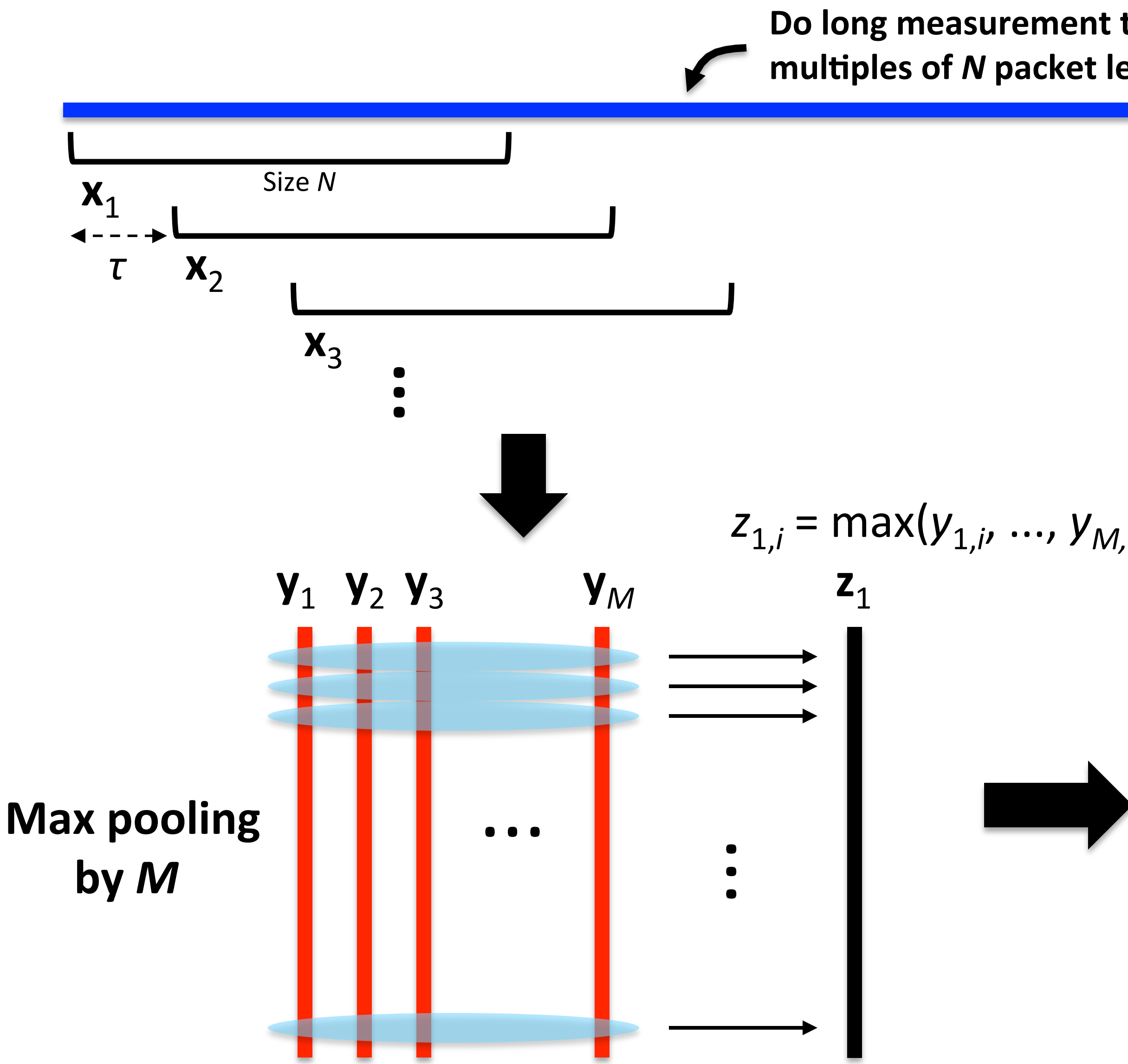
$$Recall = \frac{\sum \text{True positives}}{\sum \text{True positives} + \sum \text{False negatives}}$$

$$False\ alarm = \frac{\sum \text{False positives}}{\sum \text{False positives} + \sum \text{True negatives}}$$

For multiple hypothesis testing, false discovery rate (FDR) could be used instead of false alarm rate

$$FDR = \frac{\sum \text{False positives}}{\sum \text{False positives} + \sum \text{True positives}}$$

Feature Extraction and Pooling Details



To next layer:

$$\mathbf{x}_j^{(\mathrm{I}+1)} = \mathbf{z}_j^{(\mathrm{I})}$$