KDD2021

Fruit-fly Inspired Neighborhood Encoding for Classification

Kaushik Sinha[1,2], Parikshit Ram[3]

[1]Department of Electrical Engineering and Computer Science, Wichita State University

[2]Institute for Foundations of Machine Learning

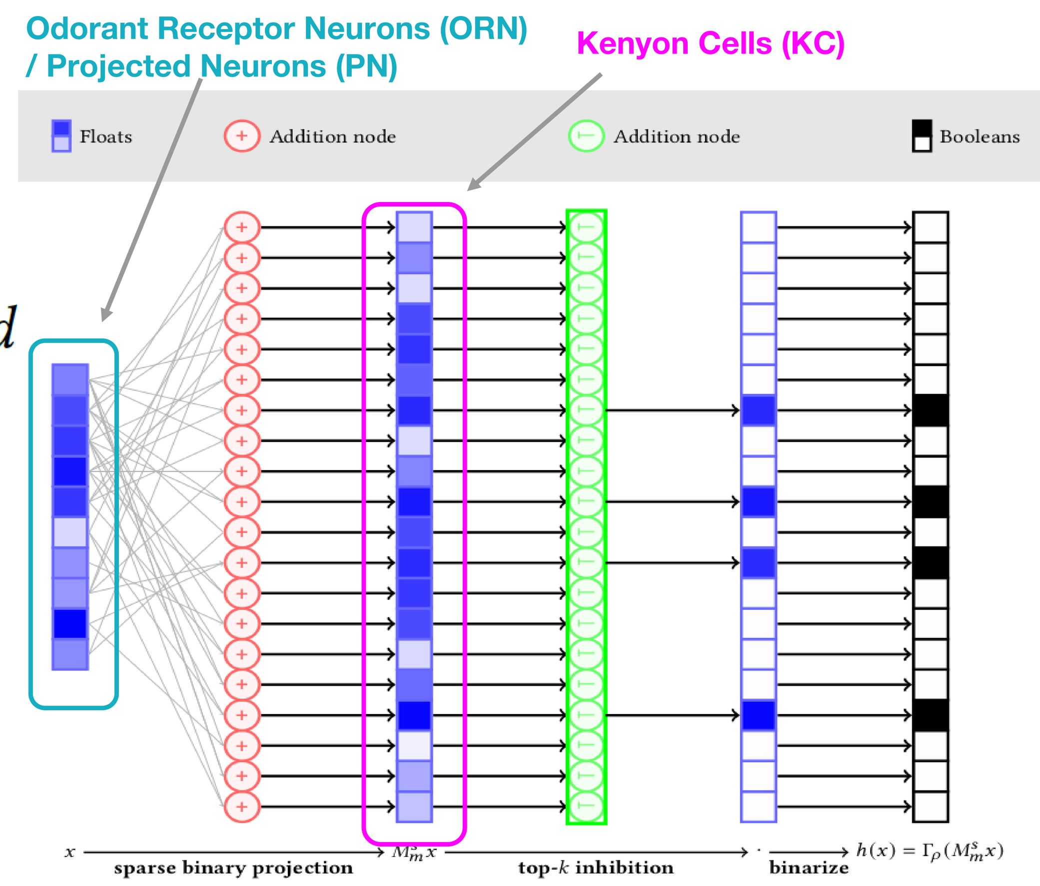
[3]IBM Research AI

Wichita State University

IFML Institute for Foundations of Machine Learning

FlyHash

- Mapping $h(x) = \Gamma_\rho(M_m^s x)$
- Random sparse binary lifting matrix with $s \ll d$ zero entries/row $M_m^s \in \{0,1\}^{m \times d}$
- Winner-take-all setting $\rho \ll m$ largest entries to 1 and rest to 0 $\Gamma_\rho : \mathbb{R}^m \to \{0,1\}^m$



FBF: Fly Bloom Filter

- Each class FBF $w \in \{0,1\}^m$ summarizes data points from that class
- Starting with an all ones vector, update is done as follows:

$$w \leftarrow w \wedge (w \oplus h(x_{\text{in}})) = w \wedge \overline{h(x_{\text{in}})}$$

- FBF for each class attempts to assign
  - low novelty scores to data points from the same class
  - high novelty scores to data points from different classes

Motivation

- Can we **reprogram FBF to the supervised learning setting** and devise a classifier based on the simple learning dynamics of the FBF?
- Will such a supervised classification scheme be **useful and competitive when learning needs to happen with a single pass**?
- What **generalization guarantees would such a learner have**?

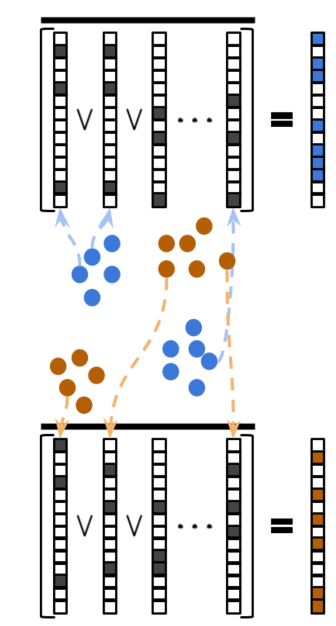
Contributions

- **Design of a novel FBF based FlyHash Bloom Filter Classifier -- FBFC**
  - ***Learning:*** additions only operations, single-pass, no loss-minimizing optimization
  - ***Inference:*** efficient FlyHash followed by a sparse binary additions-only dot-product
- **A thorough empirical comparison** of FBFC to standard classifiers
  - on **over 71 datasets** demonstrating significant gains over single-pass baselines
- A **theoretical examination** of the proposed scheme
  - establish **conditions under which FBFC agrees with the nearest-neighbor classifier**, thereby inheriting its generalization guarantees
- How the FBFC can **provide insights into the problem structure**
  - in terms of a **class hierarchy** in classification problem

FBFC Learning

TrainFBFC: $(S, m, \rho, s) \to (M_m^s, \{w_l, l \in [L]\})$
Sample $M_m^s \in \{0,1\}^{m \times d}$ with s NNZ/row
Initialize $w_1, \dots, w_L \leftarrow \mathbf{1}_m \in \{0,1\}^m$
for $(x, y) \in S$ **do**
  $h(x) \leftarrow \Gamma_\rho(M_m^s x)$
  $w_y \leftarrow w_y \wedge \overline{h(x)}$
end
return $(M_m^s, \{w_l, l \in [L]\})$
end

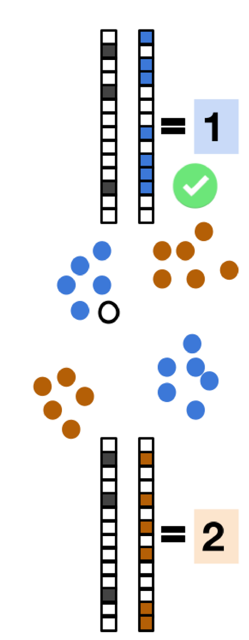
$$w_l = \mathbf{1}_m \bigwedge_{(x,y) \in S:\ y=l} \overline{(h(x))}$$



- Learns separate FBFs (encoding) for each class
- The learning scheme is online
  - an example *can be used in isolation to update the model*
  - an example *does not need to be seen more than once*

FBFC Inference

InferFBFC: $(x, M_m^s, \rho, \{w_l, l \in [L]\}) \to \hat{y}$
  $h(x) \leftarrow \Gamma_\rho(M_m^s x)$
  $\hat{y} \leftarrow \operatorname{argmin}_{l \in [L]} (1/\rho) w_l^\top h(x)$
  return $\hat{y}$
end



- Each class FBF ensure by construction
  - points similar to points in class are treated as inliers
  - points from other classes do not affect the class encoding
- Compute novelty score for a test point on all FBFs and select class with lowest novelty score

Robust Learning of FBFC*

- **Problem:** Single mislabeled example can damage class' binary FBF
- **Remedy:** Non-binary FBF to capture neighborhood more robustly
  - FBF coordinates decayed at a controlled rate

TrainNBFBFC: $(S, m, \rho, s, c) \to (M_m^s, \{w_l, l \in [L]\})$
Sample $M_m^s \in \{0,1\}^{m \times d}$ with s NNZ/row
Initialize $z_1, \dots, z_L \leftarrow \mathbf{0}_m \in \mathbb{R}^m$
for $(x, y) \in S$ **do**
  $h(x) \leftarrow \Gamma_\rho(M_m^s x)$
  $z_y \leftarrow z_y + h(x)$
end
$w_l \leftarrow (1-c)^{\odot z_l}, l \in [L]$
return $(M_m^s, \{w_l, l \in [L]\})$
end

$$w_{lj} = (1-c)^{|\{(x,y) \in S:\ y=l \text{ and } (h(x))_j = 1\}|}, l \in [L], j \in [m]$$

Robustness of FBFC vs FBFC* on Synthetic Data

Label noise level	1.0%	5.0%	10%	25%
FBFC accuracy (%)	56.6±0.8	54.5±0.9	52.0±1.3	44.0±1.0
FBFC* accuracy (%)	60.1±0.7	58.9±0.9	57.1±1.2	50.3±1.2
Rel. improvement (%)	6.1±0.7	8.1±0.9	9.8±1.2	14.2±1.8

Theoretical Guarantee

Runtime complexities

$O(nm \cdot \max\{s, \log \rho\})$	**Training time (Claim 1, 3)**
$O(m \cdot \max\{s, \log \rho, (\rho L/m)\})$	**Inference time (Claim 2)**

Learning Theoretic Properties

THEOREM 4. *Fix any $\delta \in (0,1)$, $s \ll d$, and $\rho \ll m$. Given a training set S as described above and a test example $x \in \mathcal{X}$, let x_{NN} be its closest point from S measured using ℓ_p metric for an appropriate choice of p. If (i) $\rho = \Omega(\log(1/\delta))$, (ii) $\|x - x_{NN}\|_p = O(1/s)$, and (iii) $m = \Omega(n\rho)$, then under mild conditions, with probability at least $1 - \delta$ (over the random choice of lifting matrix M), prediction of* FBFC *on x agrees with the prediction of* 1NNC *on x.*

Empirical Evaluation

Comparison against baselines

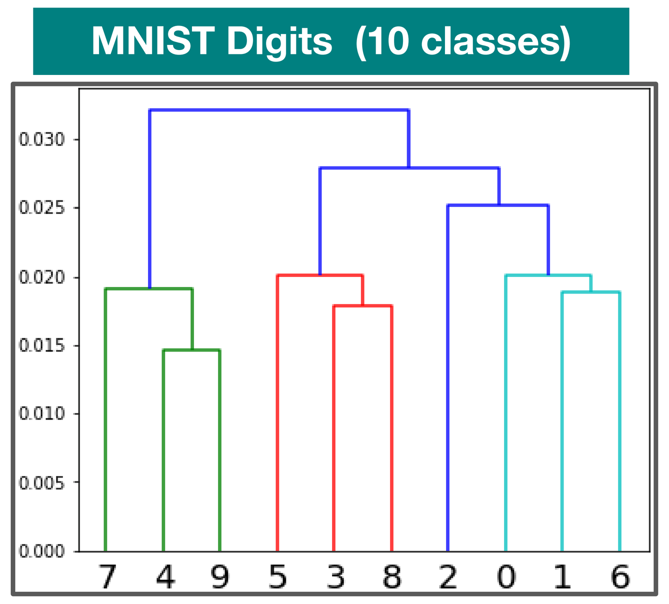
- **Baselines:** nearest-neighbor (kNNC, 1NNC), prototype based (CC1, CC), LSH based bloom filters (SBFC), linear (LR), multi-layered perceptrons (MLPC)
- Three groups of **71 OpenML data sets**: **All** (all 71 sets), **Group A** (kNNC best among baselines), **Group B** (LR best among baselines)
- Performance relative to FBFC*: **Fraction FBFC* wins** & **relative margin of improvement**

Method	All (71 sets)	**Group A** (37/71)	**Group B** (34/71)
kNNC	0.51 (▲0.05%) ○	0.24 (▼1.08%) ○	0.79 (▲3.98%) ●
1NNC	0.62 (▲2.21%) ●	0.38 (▼0.24%) ○	0.88 (▲12.1%) ●
CC1	0.87 (▲7.64%) ●	0.95 (▲11.8%) ●	0.79 (▲4.96%) ●
CC	0.38 (▼0.48%) ○	0.38 (▼0.31%) ○	0.38 (▼0.50%) ○
SBFC	0.99 (▲24.9%) ●	0.97 (▲24.7%) ●	1.00 (▲25.2%) ●
LR	0.58 (▲1.34%) ○	0.78 (▲3.39%) ●	0.35 (▼0.68%) ○
MLPC	0.73 (▲4.36%) ●	0.81 (▲6.57%) ●	0.65 (▲3.80%) ●
FBFC	0.82 (▲5.87%) ●	0.68 (▲0.73%) ●	0.97 (▲9.13%) ●

Problem insights

- **Class hierarchy** based on per-class FBF similarities

MNIST Digits (10 classes)



English Letters (26 classes)