Pareto Optimal Streaming Unsupervised Classification

Sanjay Shakkottai[1]

Joint with Soumya Basu[1], Steven Gutstein[2], Brent Lance[2]

1. UT Austin, 2. ARL

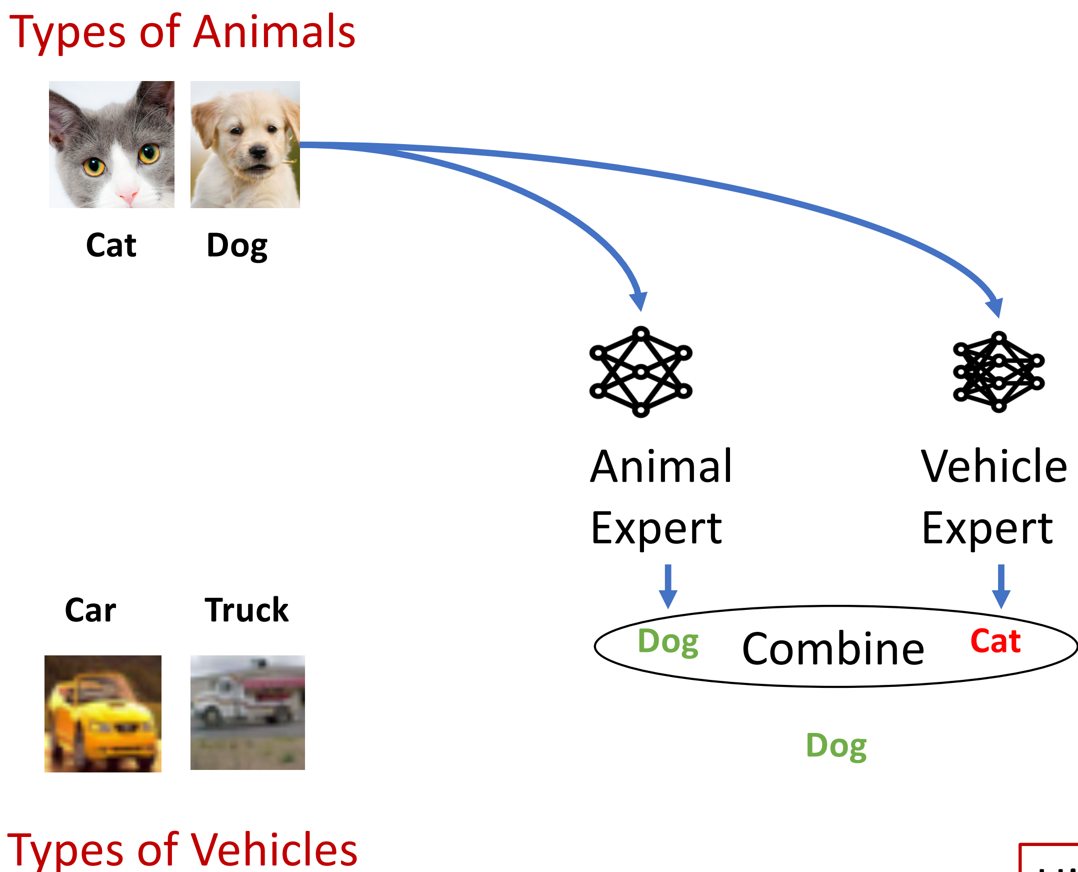
Streaming Unsupervised Learning

Tasks: Images arriving as a stream for classification

Agents: Human agents, Neural-net classifiers

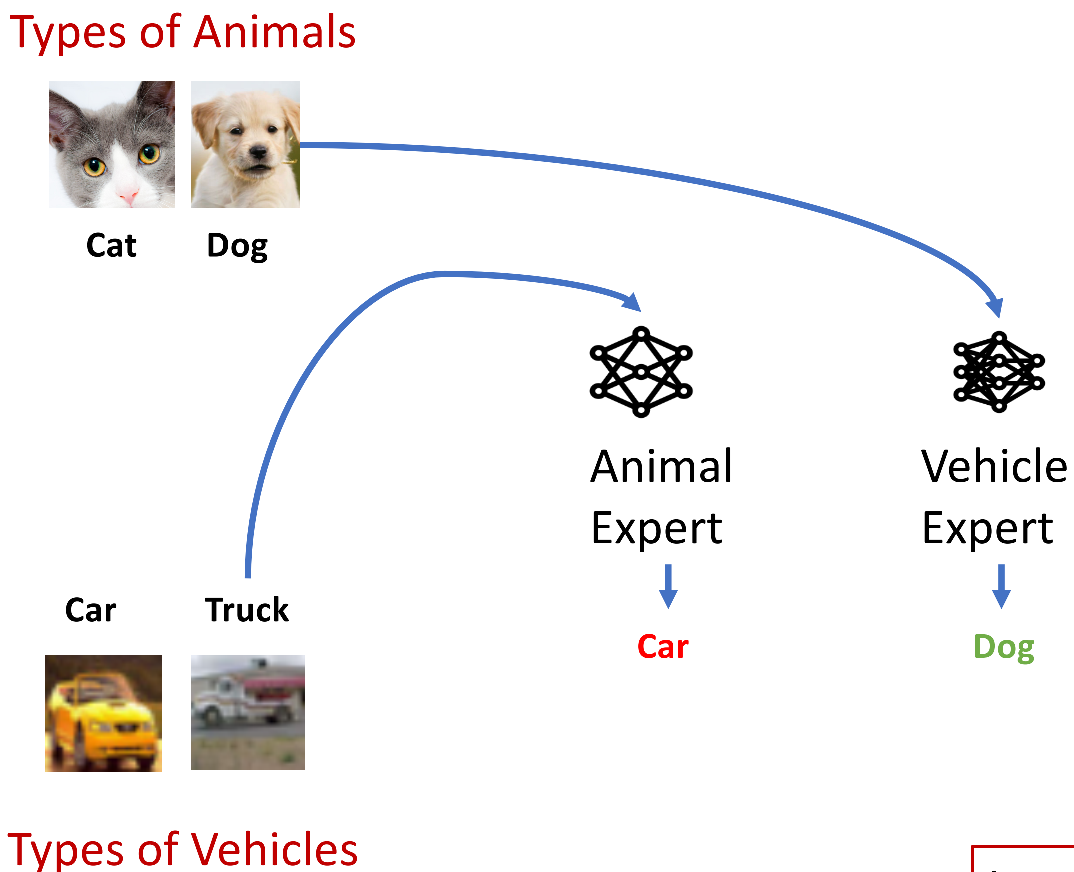
- **Different expertise**: Unknown confusion matrices
- **Frozen labeling**: Label for each image fixed
- **Processing Rate**: One image/round

Image credits: CIFAR-10, A. Krizhevsky, 2009; thenounproject.com, (NNs - K. M. Synstad; Faces - A. Selimov)



Safest Strategy:
Send each image (task) to both the Classifiers

High accuracy, but low throughput

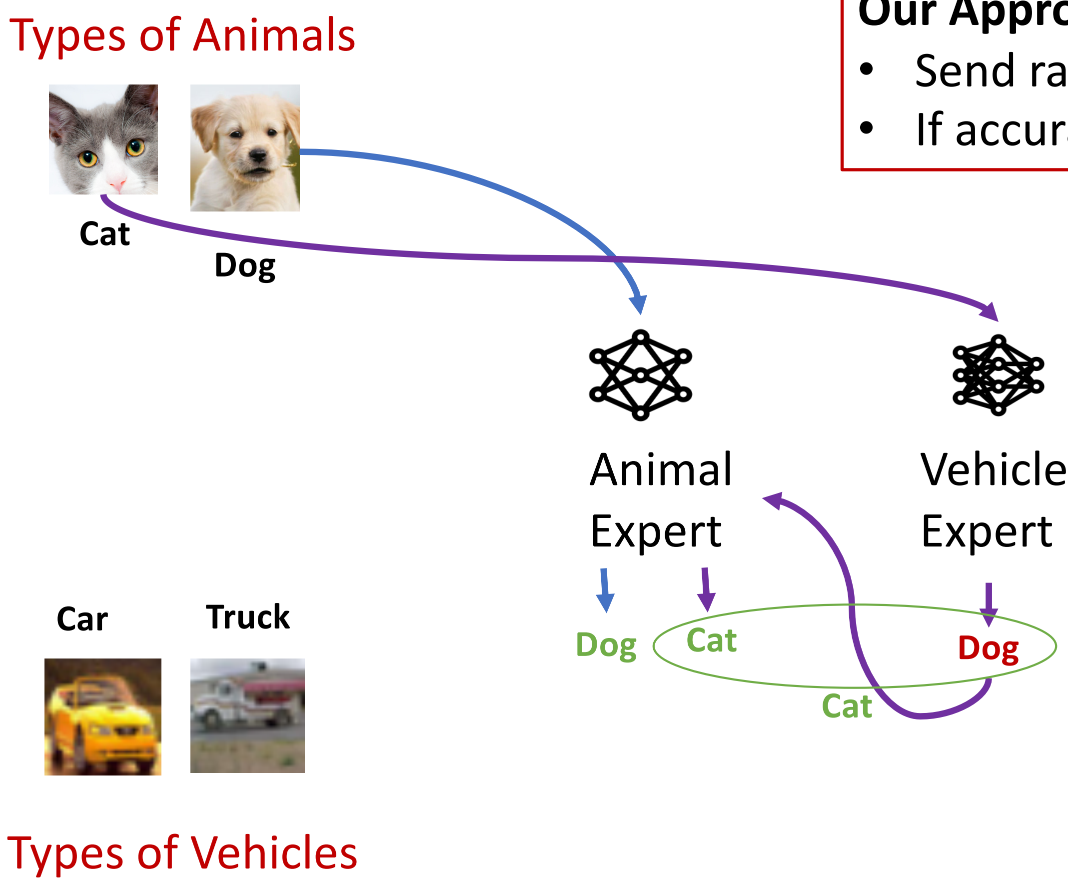


Fastest Strategy:
Send each image (task) randomly to one Classifier

Lower accuracy, but high throughput

Our Approach: Dynamic Routing

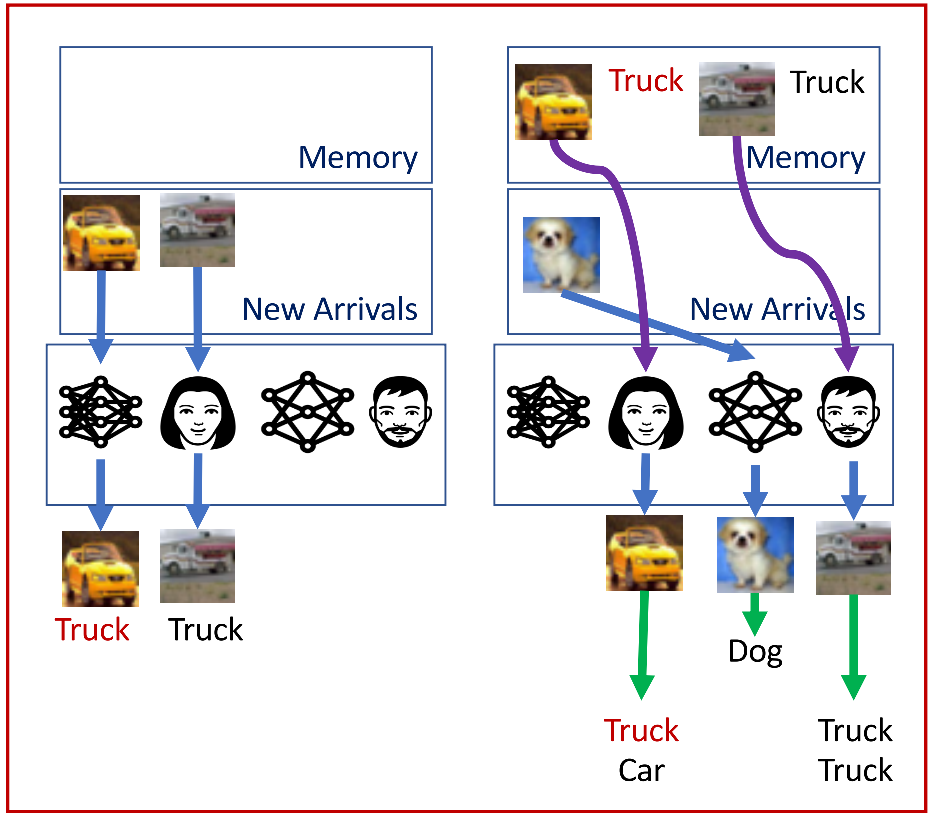
- Send randomly to one classifier
- If accuracy is low send to the remaining



Desired Accuracy
High Speed

Resource Allocation and Label Aggregation

- Sequential routing to subset of agents
- Collected labels are continually aggregated
- Aggregation using estimated confusion matrices



Streaming Dawid-Skene Model

- Time slotted system, **K** classes of images, and **M** classifiers
- Independent identically distributed arrivals with rate $\boldsymbol{\lambda}$ **images/timeslot**
- Classifier characterized by its confusion matrix

A classifier's labels for a specific image is **frozen**. Repeatedly sending an image to the same classifier does not result in new labels.

Example: Trained Neural Network

Pareto Optimality

- **Causal Policy:**
  Routing and aggregation are randomized functions of observed history
- $\mathbf{Confidence} = \mathbb{P}[\text{True label} = \text{final label} \mid \text{History}\,]$

Threshold Accuracy ($\boldsymbol{\theta}$)

Accurately labeled with Confidence $\geq \theta$

OR

Sent to ALL the classifiers

Final label = f(History, independent randomness)

- **Arrival Rate ($\boldsymbol{\lambda}$) vs Threshold Accuracy ($\boldsymbol{\theta}$):**
  Higher θ requires more classifiers per image on average
  Higher the threshold θ the lower the arrival rate we can support

- **Goal**: Achieve the "best" trade-off between throughput ($\boldsymbol{\lambda}$) and accuracy ($\boldsymbol{\theta}$)

Prior Work: One-shot Unsupervised Learning

- Batch of samples without knowledge of true label (no ground truth)
- Fixed classifiers with unknown confusion matrices
- **Aim:** Combine label of all the classifiers
- **EM:** Dawid et al. JRSS'1979, Liu et al. Neurips'12, Zhang et al. Neurips'14
- **Majority Voting:** Li et al. Stat'14, Parisi et al. PNAS'14
- **Learning Confusion Matrices:** Zhang et al. Neurips'14, Jain et al. COLT'14

- Differences - In our setting:
  - Streaming classification
  - Rate vs accuracy tradeoff

Prior Work: Budgeted Crowdsourcing

- Finite batch of samples at beginning of time
- Classifiers from crowd arrive online (effectively randomized classifiers)
- Fixed budget of total number of classifier actions (over all samples)
- **Aim:** Achieve maximum accuracy given budget
- **Optimal accuracy rates in budgeted crowdsourcing:**
  - Karger et al. Neurips'11, Khetan et al. Neurips'16

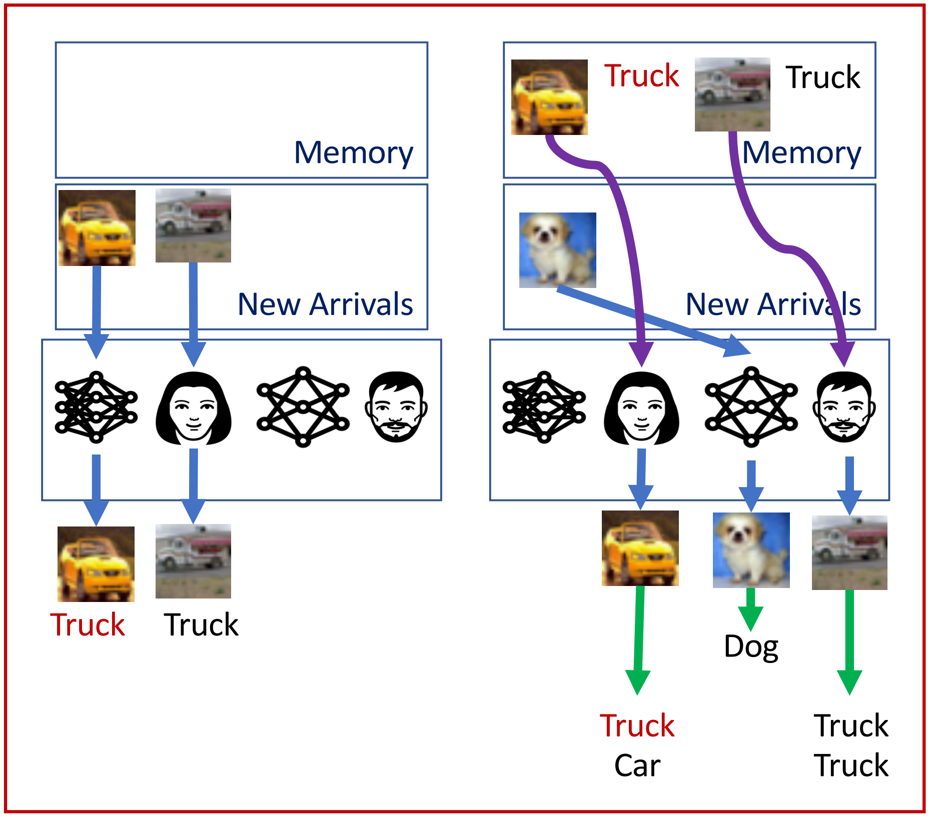
- Differences - In our setting:
  - Samples arrives online
  - Frozen classifiers

Prior Work: Information Processing Networks

- Fixed but randomized processors
- Accuracy can be arbitrarily high
- Streaming arrival of samples
- Exogenous departure from processors
- **Aim:** Support maximum arrival rate
- **Capacity region characterization**: Shah et al. Allerton'17

- Differences - In our setting:
  - Endogenous departures
  - Frozen classifiers

How Does Belief Evolve? (1/2)



$$\overbrace{Schedule(t) \to Labels(t)}^{Events(t)}$$

$$\mathbf{History(t)} = \bigcap_{s=1}^{t} \boldsymbol{Events(s)}$$

$$\mathbb{P}\left[\cap_{image\ j}\ \mathbf{T_j} = \mathbf{k(j)} \middle| \mathbf{History(t)}\right]$$

Belief at time t

$\mathbf{T_j}$: true label of image j

How Does Belief Evolve? (2/2)

$\mathbf{T_j}$: true label of image j
$\boldsymbol{L(i,j)}$: Label of image j from classifier i
$\boldsymbol{Cl(j,t)}$: Classifiers assigned to image j up to time t

Is this true?

$$\mathbb{P}\left[\cap_{\boldsymbol{image}\ \mathbf{j}} \mathbf{T_j} = \mathbf{k(j)} \middle| \mathbf{History(t)}\right] = \frac{\mathbf{1}}{\boldsymbol{Z'}} \prod_{\boldsymbol{image}\ \mathbf{j}} \mathbb{P}\left[\mathbf{T_j} = \mathbf{k(j)} \middle| \cap_{\boldsymbol{i \in Cl(j,t)}} \boldsymbol{L(i,j)} = \boldsymbol{l(i,j)}\right]$$

The conditional belief on the true label an image potentially depends on entire history of all co-existing images and their labels

- The set of labels acquired by an image until time t depends on the set of classifiers that the image was scheduled to, until time t
- This set of classifiers depends on the scheduling decisions until time t
- These scheduling decisions depends on the entire past history of labels of all images in the system until time t

Product Form of Belief Evolution

- Distribution of history depends on the the true label only through the collected labels of the sample

$\mathbf{T_j}$: true label of image j
$\boldsymbol{L(i,j)}$: Label of image j from classifier i
$\boldsymbol{Cl(j,t)}$: Classifiers assigned to image j up to time t

<u>Product form on the Belief Evolution</u>

$$\mathbb{P}\left[\cap_{\boldsymbol{image}\ \mathbf{j}}\ \mathbf{T_j} = \mathbf{k(j)} \middle| \mathbf{History(t)}\right] = \frac{\mathbf{1}}{\boldsymbol{Z'}} \prod_{\boldsymbol{image}\ \mathbf{j}} \mathbb{P}\left[\mathbf{T_j} = \mathbf{k(j)} \middle| \cap_{\boldsymbol{i \in Cl(j,t)}} \boldsymbol{L(i,j)} = \boldsymbol{l(i,j)}\right]$$

$$= \frac{\mathbf{1}}{\boldsymbol{Z(t)}} \prod_{\boldsymbol{image}\ \mathbf{j}} \prod_{\mathbf{i} \in \boldsymbol{Cl(j,t)}} \mathbb{P}[\boldsymbol{L(i,j)} = \boldsymbol{l(i,j)} | \mathbf{T_j} = \mathbf{k(j)}]\mathbb{P}[\mathbf{T_j} = \mathbf{k(j)}] \quad \textbf{(Bayes Form)}$$

Product Form of Belief Evolution

Induction on t: $\mathbb{P}[\boldsymbol{History(t)} | \cap_{\mathbf{j}} \boldsymbol{T_j} = \boldsymbol{k(j)}]$

$$= \frac{\mathbf{1}}{\boldsymbol{Z'(t)}} \prod_{j} \prod_{i \in Cl(j,t)} \mathbb{P}[\boldsymbol{L(i,j)} = \boldsymbol{l(i,j)} | \boldsymbol{T_j} = \boldsymbol{k(j)}]$$

$$\overbrace{Sch(t) \rightarrow Labels(t)}^{\boldsymbol{Events(t)}}$$

$$\mathbb{P}[History(t+1) | \cap_{\mathbf{j}} T_j = k(j)] = \mathbb{P}\,[Events(t+1) \cap History(t) | \cap_{\mathbf{j}} T_j = k(j)]$$

$$= \mathbb{P}[\,Events(t+1) | \,History(t), \cap_{\mathbf{j}} T_j = k(j)]\, \mathbb{P}[History(t) | \cap_{\mathbf{j}} T_j = k(\mathrm{j})]$$

$$= \mathbb{P}[Sch\,(t+1) | \,History(t), \cancel{\cap_{\mathbf{j}} T_j = k(j)}] \times \mathbb{P}[Labels(t+1) | \cap_{\mathbf{j}} T_j = k, Sch\,(t)]$$

$$\times \frac{1}{Z'(t)} \prod_j \prod_{i \in Cl(j,t)} \mathbb{P}[L(i,j) = l(i,j) | T_j = k(j)]$$

Causal Policy

continued….

Product Form of Belief Evolution

Induction on t: $\mathbb{P}[\boldsymbol{History(t)} \mid \cap_{\mathbf{j}} \boldsymbol{T_j} = \boldsymbol{k(j)}]$

$$= \frac{\mathbf{1}}{\boldsymbol{Z'(t)}} \prod_{\boldsymbol{j}} \prod_{\boldsymbol{i \in Cl(j,t)}} \mathbb{P}[\boldsymbol{L(i,j)} = \boldsymbol{l(i,j)} \mid \boldsymbol{T_j} = \boldsymbol{k(j)}]$$

$$= \mathbb{P}\left[\cap_{(i,j) \in Sch(t+1)} L(i,j) = l(i,j) \middle| \cap_{\mathbf{j}} T_j = k\right]$$

$Sch(t)$ is a set of (classifier, image) pairs

$$\times \frac{\mathbb{P}[Sch\ (t+1) \mid History(t)]}{Z'(t)} \prod_j \prod_{\boldsymbol{i \in Cl(j,t)}} \mathbb{P}[L(i,j) = l(i,j) \mid T_j = k(j)]$$

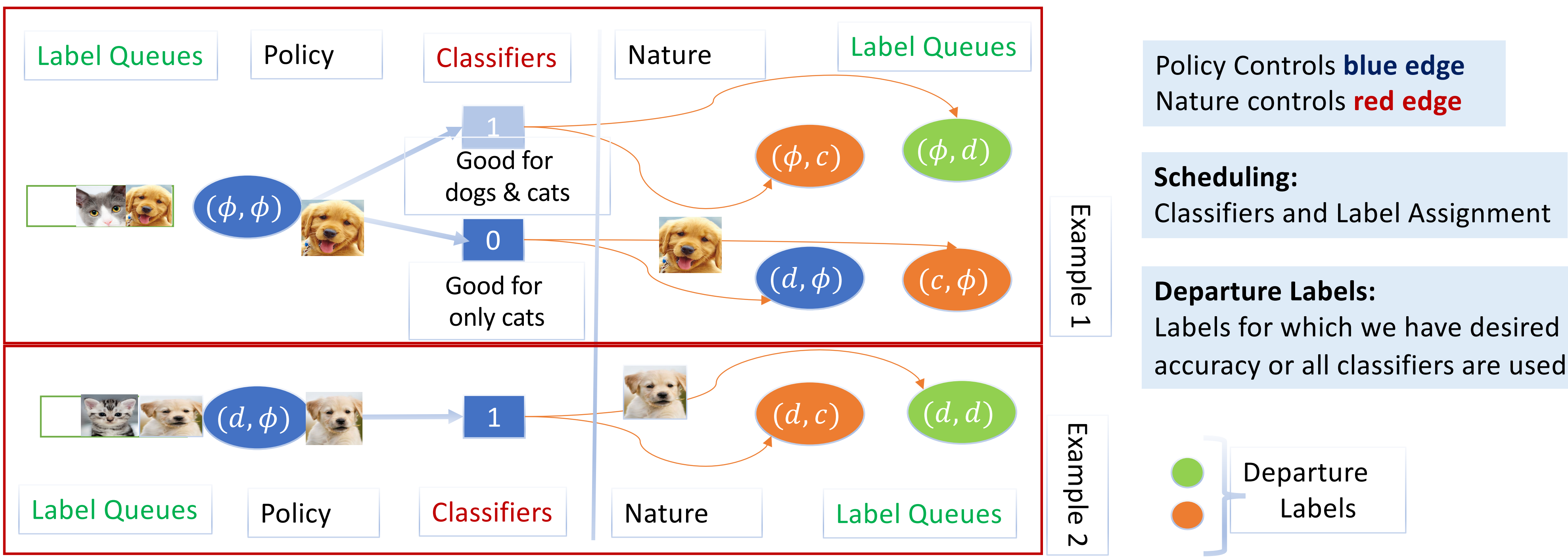
$$= \prod_{(i,j) \in Sch(t+1)} \mathbb{P}[L(i,j) = l(i,j) \mid T_j = k(j)]$$

After scheduling, new label of image j depends only on true label of j

$$\times \frac{\mathbb{P}[Sch\ (t+1) \mid History(t)]}{Z'(t)} \prod_j \prod_{\boldsymbol{i \in Cl(j,t)}} \mathbb{P}[L(i,j) = l(i,j) \mid T_j = k(j)]$$

Network of Label Queues

- **Partial Label**: Observed label tuple, e.g. (cat, none), (dog, cat), (none, none)
- **Label Queues:** Binary classification (dogs vs cats) with 2 classifiers



Expected MaxWeight for Agent Selection

- Let $Q_\ell(t)$ be queue length of label queue ℓ at time t
- Under assignment $\boldsymbol{S}$
  - Departure from queue ℓ, $D_\ell(\boldsymbol{S})$
  - Arrival into queue ℓ, $A_\ell(\boldsymbol{S})$
- Choose assignment according to Expected MaxWeight:

$$\boldsymbol{S}(t) = argmax \sum_{\ell} Q_\ell(t)(D_\ell(\boldsymbol{S}) - \mathbb{E}[A_\ell(\boldsymbol{S})])$$

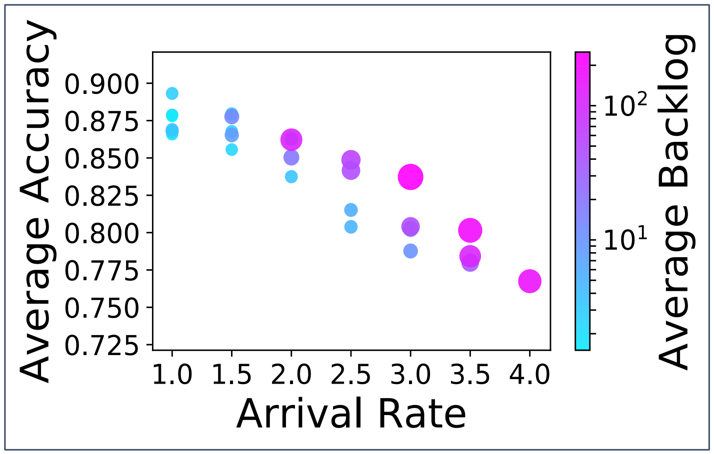
- Belief is only dependent on the collected labels
- Expectation on Nature's choice does not vary over time
  - Depends only on labels and confusion matrix

Performance Guarantees

- **Pareto region:**
  Set of (Arrival rate $(\boldsymbol{\lambda})$, threshold $(\boldsymbol{\theta})$) tuple so that the sum of expected queue length remains bounded under some causal policy
- Pareto region is characterized by a **maximum network flow** problem
- **Pareto Optimality:**

For any **arrival rate $\boldsymbol{\lambda}$** and **threshold $\boldsymbol{\theta}$, for any $\boldsymbol{\epsilon > 0}$,**
if $(\boldsymbol{\lambda} + \boldsymbol{\epsilon}, \boldsymbol{\theta} + \boldsymbol{\epsilon})$ lies in the Pareto region
then Expected Backpressure + online tensor decomposition
support $(\boldsymbol{\lambda}, \boldsymbol{\theta})$ with bounded expected memory.

Neural Network Ensembles

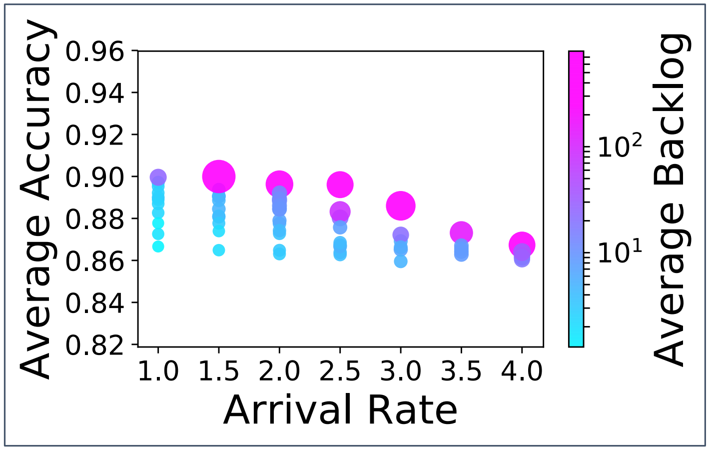


Pareto Region Exp 1

Experiment 1

- **6 Classifiers:** Three AlexNet, one VGG-19, and two ResNet-18
- **3 Labels:** Group 1 (airplanes, ships, trucks, cars), Group 2 (birds, frogs cats), and Group 3 (dogs, deer, horses)

Neural Network Ensembles



Pareto Region Exp 2

Improved Classifiers lead to improved Pareto Region

Experiment 2

- **6 Classifiers:** One VGG-11, one VGG-16, two VGG-19, and two ResNet-18
- **5 Labels:** Group 1 (airplanes, ships), Group 2(trucks, cars), Group 3(birds, frogs), Group 4 (cats, dogs) and Group 5(deer, horses)

Future Directions

- What fraction of Pareto Region can be covered by simple algorithms?

  - Algorithms without explicitly recovering parameters
    e.g. Majority voting, Routing over limited number of rounds

- Can we maximize threshold accuracy given an arrival rate?
- Can we maximize average accuracy inside Pareto Region?

Backup Slides

Online Learning of Confusion Matrices

- Explore (w.p. $1/t$ at time t) by sending one sample to all classifiers
- An $(\boldsymbol{\alpha}, \boldsymbol{\beta})$**-oracle**, with n exploration samples, outputs confusion matrices and true probability vector with L_∞ error at most $\mathrm{O}(n^{-\alpha})$ with probability at least $1 - \Omega\left(n^{-\beta}\right)$
- $(0.5 - \epsilon, 1 - \epsilon)$-oracle created by adapting tensor decomposition based one-shot unsupervised learning [Zhang et al. 2014]
  - Reinitialize only when a new explore sample is obtained
  - Ensure w.p. 1 the initialization is 'good' for tensor power method
  - Iteratively improve estimates between two exploration instances

Online Dawid-Skene Model

- Time slotted system, **K** classes of images, and **M** classifiers
- I.i.d. Arrival with rate $\boldsymbol{\lambda}$ **images/timeslot**
- **Classifier i :**
  - Confusion matrix $\boldsymbol{C_i}(\cdot,\cdot)$
  - **Speed** = 1 image / timeslot
- **Image j:** (M+1)-tuple $\left(L_j(1), \dots, L_j(M), T_j\right)$
  - **True label:** Label $T_j \sim p_g \in \Delta_{\mathrm{K}}$ chosen and fixed
  - **Label from Classifier i:** Label $L_j(i) \sim C_i\left(T_j, \ \cdot\ \right)$ chosen and fixed

A specific classifier labels a specific image deterministically

e.g. Trained Neural Network

Sufficiency of Compressed Causal Policy

- **Compressed History:**

Aggregate samples with identical collected labels, delete the sample ids

- **Compressed Causal Policy:**

Decisions are randomized functions of the compressed history

For any causal policy there is a compressed causal policy s.t.

- For all time t, the compressed history distribution is identical
- Threshold accuracy is attained is identical

Key Proof Idea:
The belief of a sample only depends on its collected labels.
The new labels and threshold accuracy only depend on the belief.