Collaborative Image Triage with Humans and Computer Vision

Addison Bohannon
Applied Math, Statistics, & Scientific Computing

Advisors:

Vernon Lawhern
Army Research Laboratory

Brian Sadler
Army Research Laboratory

May 3, 2016

Outline

Motivation

We want to triage a large database of unlabeled images:

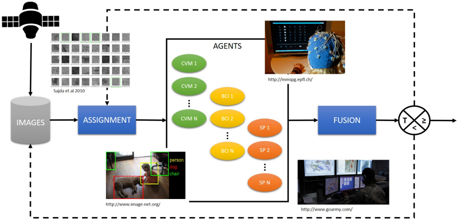
- Our purpose is motivated by DOD imagery intelligence requirements, but other people are interested in this and similar problems:
  - Google Images, Facebook, Galaxy Zoo, fold.it
- This could be fully automated by computer vision algorithms, but they require:
  - Training data (lots) and time (lots); or
  - Knowledge of the generating process of the data
- This could be done by humans, but...
  - Humans take a lot of time to classify images
  - Task may require expertise or security clearance
  - Humans require salary, benefits, pension, etc.

Related Work

How to triage a large image database

- Human augmentation
  - Rapid Serial Visual Presentation (RSVP) for image labeling [Bigdely-Shamlo et al., 2008]
- Human-machine systems
  - Serialize RSVP analyst and computer vision (CV) algorithm [Sajda et al., 2010]
  - Automate image labeling with CV which can query a human analyst for binary decisions [Joshi et al., 2012]
- Crowd-sourcing
  - Intelligent control of a system which dynamically scales human participants [Kamar et al., 2012]
  - Homogeneous human agents whose voting reliability is learned [Karger et al., 2014]
  - Heterogeneous human agents intelligently assigned heterogeneous tasks [Ho et al., 2013]

Research Objective



- **Goal:** To design and implement in software an image triage system which leverages an ensemble of heterogeneous agents to achieve the accuracy of a naive parallel implementation in significantly less wall time.
- **Problem Statement:**
  - How to optimally distribute images among agents?
  - How to combine responses from multiple agents?
  - How to design a software system which can support heterogeneous image labeling interfaces in parallel?

Schedule

- Develop Joint Classification Module (Summer 2015)
  - Implement Spectral Meta-Learner algorithm
- Develop Assignment Module (15 OCT - 4 DEC)
  - Implement branch and bound algorithm (6 NOV)
  - Validate branch and bound algorithm (25 NOV)
  - Mid-year review (14 DEC)
- Build Image Labeling System (25 JAN - 26 FEB)
  - Build base classes
  - Develop message-passing interface
  - Integrate all components into a system (26 FEB)
- Test Image Labeling System (26 FEB - 15 APR)
  - Testing (1 APR)
- Conclusion (15 APR - 13 MAY)
  - Final presentation (3 MAY)
  - Final report (13 May)

Generalized Assignment Problem

On iteration k, we seek the optimal assignment of n images among m agents–with a fixed budget, b_j^k, and reliability, r_j^k–where each assignment has a unique value, v_{ji}^k, and cost, c_{ji} [Kundakcioglu and Alizamir, 2008]:

$$Z = \max_{\mathbf{x}} \sum_{i \in I} \sum_{j \in J} v_{ji}^k x_{ji} \quad \text{s.t.} \tag{1}$$

1. $\sum_{i \in I} c_{ji} x_{ji} \leq b_j^k, \; j \in J$
2. $\sum_{j \in J} x_{ji} = 1, \; i \in I$
3. $x_{ji} \in \{0, 1\}$
4. $c_{ji}, b_j^k \in \mathbb{Z}_+$
5. $v_{ji}^k = r_j^k - s_i^k + \max_{i \in I} s_i^k$

- 0-1 integer linear problem
- NP-hard
- Known solution techniques

Branch and Bound Algorithm

Algorithm 1: Branch & Bound

Data: Z_0
Result: x
$Z = Z_0, \; queue = p_0$;
while $queue \neq \emptyset$ **do**
  Select $p^i \in queue$
  for $j \in J$ **do**
    $Z^i_j = bound(p^i_j)$;
    if $Z^i_j > Z$ **then**
      if x_j *is feasible* **then**
        $x = x^i_j, \; Z = Z^i_j$
      else
        add p^i_j to $queue$
      end
    end
  end
end

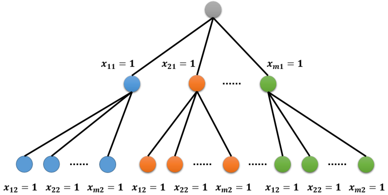


Figure: Visualization of branch and bound (B&B) algorithm. Nodes along the *m*-nary search tree represent sub-problems ($p^i_j \sim x_{ji} = 1$).

Bounding Function

We introduce the dual problem [Fisher, 2004],

$$d(\boldsymbol{\lambda}) = \max_{\mathbf{x}} \sum_{i \in I} \sum_{j \in J} v_{ji} x_{ji} - \sum_{i \in I} \lambda_i \Big(1 - \sum_{j \in J} x_{ji}\Big),$$

to define our bounding function,

$$\min_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}) \geq Z \geq Z_{feasible}.$$

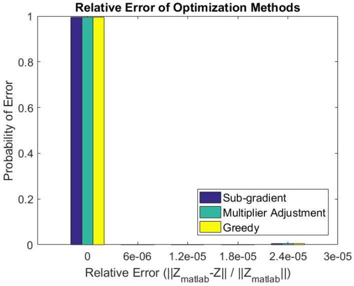
Then, we solve the saddle-point problem directly via sub-gradient descent [Boyd and Vandenberghe, 2004]:

$$\mathbf{x}^{k+1} = \arg\max_{\mathbf{x}} \sum_{i \in I} \sum_{j \in J} (v_{ji} - \lambda_i^k) x_{ji} \quad \text{s.t.} \quad \sum_{i \in I} c_{ji} x_{ji} \leq b_j$$

$$\lambda_i^{k+1} = \lambda_i^k + \alpha_k \left(1 - \sum_{j \in J} x_{ji}\right)$$

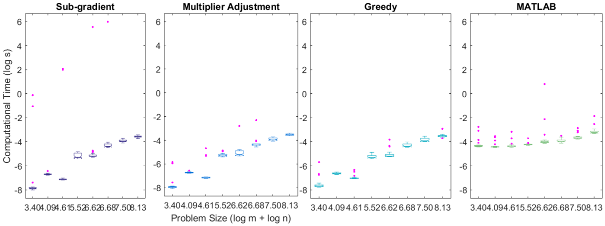
Validation

Generalized Assignment Problem Solvers



Feasibility

Solver	Probability
Sub-gradient	1.0
Multiplier	1.0
Greedy	1.0
MATLAB	0.07



Time Complexity

Maximum Likelihood Estimation

Spectral Meta-Learner

Consider the set of decisions from m agents for a single image i, $\mathbf{A}^i : \{-1, 1\}^m \to \mathbb{R}$. We seek the decision rule which maximizes $\mathbb{P}(d(\mathbf{A}^i) = y_i)$:

$$d(\mathbf{a}^i) = \arg\max_{y_i \in \{-1,1\}} \sum_{j \in J} \log \mathbb{P}_{A^i_j | Y}(a^i_j | y_i),$$

where $Y : \{-1, 1\} \to \mathbb{R}$ is the true label of an image [Dawid and Skene, 1979]. Let $\pi_j = \frac{1}{2}(\psi_j + \eta_j)$, where $\psi_j = \mathbb{P}(a_j = 1 | y_i = 1)$ and $\eta_j = \mathbb{P}(a_j = -1 | y_i = -1)$, then the decision rule is equivalent to

$$d(\mathbf{a}^i) = \text{sign} \sum_{j=1}^{m} a^i_j \left(\log \alpha_j + \log \beta_j\right),$$

where $\alpha_j = \frac{\psi_j \eta_j}{(1-\psi_j)(1-\eta_j)}$ and $\beta_j = \frac{\psi_j(1-\psi_j)}{\eta_j(1-\eta_j)}$ [Parisi et al., 2014].

Joint Classification

This provides three results:

1. Class label of each image, $\text{sign}(d(\mathbf{a}^i))$
2. Confidence of the MLE estimate of each image, $s_i = |d(\mathbf{a}^i)|$
3. Reliability of each agent, $r_j = \pi_j = \frac{1}{2}(\psi_j + \eta_j)$

Software Map

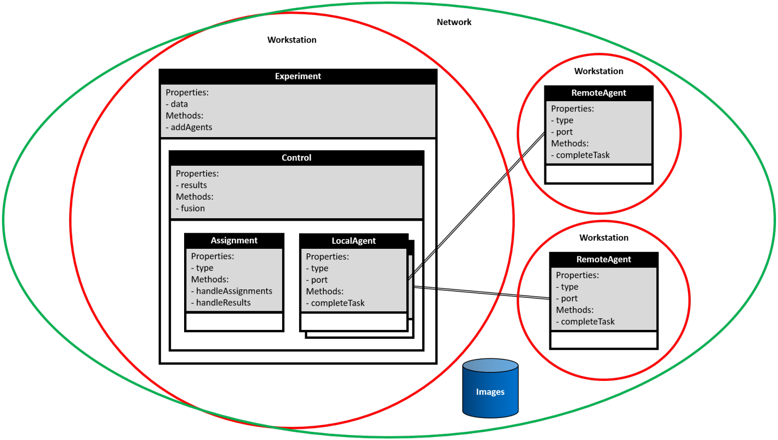


Figure: Visualization of the software design of the image triage system. Architecture prioritizes software **flexibility** and independent operation for a network of **distributed** agents.

Process Flow

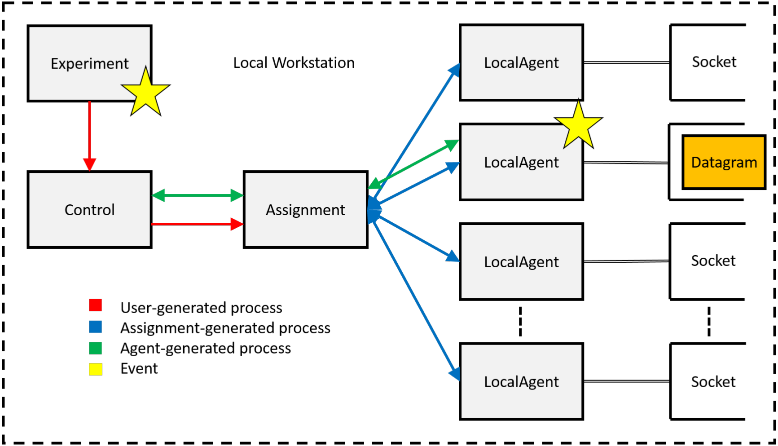


Figure: Visualization of process flow on central server. Asynchronous read operations facilitate **parallel** classification among distributed agents.

Convergence Considerations

The following methods are implemented to address instability in the system as a result of feedback[1]:

- Soft barrier to duplicate assignment, $v_{ji} = 0$
- Dynamic budget, $b_j^k = \frac{L_k}{\mu_j}$
- Monotonically increasing interval length, $L_{k+1} \geq L_k$
- Maximum interval length, $L_k \leq L_{max}$
- Alternative stopping condition (pseudo-infeasibility)

Definition

*The system achieves **convergence** when all images achieve threshold confidence, or the alternative stopping condition is reached.*

[1] L is the interval length, and μ_j is the throughput rate of an agent.

Simulation Set-up I

- Software: MATLAB R2015a
- Hardware: Unix-based desktop, two Intel Xeon 2.67 GHz processors, 8 cores (independent instance of MATLAB for each agent)
- Data: Simulated, 30 trials, 6 agents, 200 images

Type	Accuracy (p_j)	Cost (c_{ji})	Service Time (μ_j)
CV	0.75	1	0.01s
RSVP	0.85	1	0.1s
Human	0.95	1	1.0s

Table: Properties of agents used for all simulations. Labels generated by Bernoulli process, $f_{A_j|Y}(a_j|y) \sim bern(p_j)$. Service times generated by exponential random variable, $T_j \sim exp(\mu_j)$

Simulation Set-up II

- Assignment conditions:
  - Naive (control) - all images assigned to all agents in parallel in a single batch.
  - GAP-2 - images assigned in parallel according to GAP; images classified if confidence meets or exceeds two, $s_i \geq 2$.
  - GAP-3 - same as GAP-2, $s_i \geq 3$.
  - GAP-4 - same as GAP-2, $s_i \geq 4$.
- Agent ensembles:
  - Computer vision ($CV \times 6$)
  - Mixed ($CV \times 2$, $RSVP \times 2$, $H \times 2$)
  - Human ($H \times 6$)

Expected Performance of Naive Assignment

- Balanced accuracy [Parisi et al., 2014]

$$R \geq \max_{j \in J} R_j - \epsilon(|J|)$$

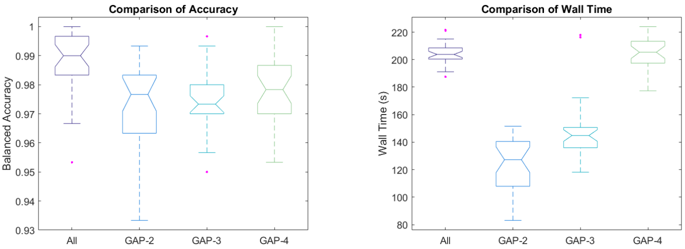
- Wall time

$$\begin{aligned} f_T(t) = \frac{\partial}{\partial t}\mathbb{P}(\max_{j \in J} T_j \leq t) &= \frac{\partial}{\partial t}\mathbb{P}(T_1 \leq t, \ldots, T_m \leq t) \\ &= \frac{\partial}{\partial t}\mathbb{P}(T_1 \leq t) \cdots \mathbb{P}(T_m \leq t) \\ &= \frac{\partial}{\partial t}\prod_{j \in J} F_{T_j}(t) \\ &= \left(\prod_{j \in J} F_{T_j}(t)\right) \sum_{j \in J} \frac{f_{T_j}(t)}{F_{T_j}(t)} \end{aligned}$$

Analytical Results of Naive Assignment

Agent Ensemble	**Accuracy (π)**	**Wall Time (T)**
CV	0.75	2.2 ± 0.1s
Mixed	0.95	208.0 ± 12.0s
Human	0.95	218.3 ± 9.7s

Table: Analytical Results of naive assignment condition across agent ensembles. These results provide a performance ceiling to which we can compare the simulation results of the mixed ensemble GAP assignment conditions.



(a) Balanced Accuracy (b) Wall Time

Figure: One-way analysis of variance (ANOVA) of the performance of heterogeneous agent ensembles across assignment conditions reveals significance in both the balanced accuracy ($F(3, 116) = 8.8$, $p = 2.6 \times 10^{-5}$) and wall time ($F(3, 116) = 186.5$, $p < 1.0 \times 10^{-9}$).

Assignment Conditions Results (Mixed Ensemble)

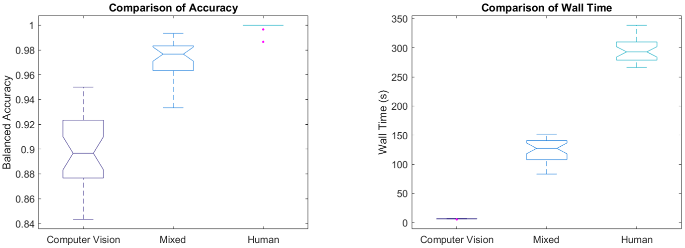
Summary Statistics

Condition	Accuracy	Wall Time	Assignments
Naive	0.988 ± 0.011	204.1 ± 7.9	1200
GAP-2	$0.974 \pm 0.014^{*,**}$	124.1 ± 19.3*	879.9 ± 16.3*
GAP-3	$0.975 \pm 0.011^{*,**}$	147.9 ± 21.8*	983.1 ± 15.1*
GAP-4	$0.978 \pm 0.011^{*,**}$	204.4 ± 12.3	1047.6 ± 6.4*

Table: Performance of heterogeneous agent ensemble across assignment conditions (* significantly different from naive assignment condition under multiple comparisons test, $p < 0.001$; ** achieved or exceeded the expected accuracy of the naive condition, one-sample T-test, $p < 0.001$). The mean of the GAP-2 condition achieves a $1.6\times$ speed-up over the mean of the naive condition, while the GAP-3 achieves a $1.4\times$ speed-up.

Agent Ensemble Results (GAP-2 Assignment)

Analysis of Variance



(a) Balanced Accuracy (b) Wall Time

Figure: ANOVA of the performance of GAP-2 assignment condition across agent ensembles reveals significance in both balanced accuracy ($F(2, 87) = 255.47$, $p < 1.0 \times 10^{-9}$) and wall time ($F(2, 87) = 2667.44$, $p < 1.0 \times 10^{-9}$).

Agent Ensemble Results (GAP-2 Assignment)

Summary Statistics

Ensemble	Accuracy	Wall Time	Assignments
CV	0.898 ± 0.030	6.3 ± 0.3s	913.8 ± 13.8
Mixed	0.974 ± 0.014	124.1 ± 19.3s	879.9 ± 16.3
Human	0.999 ± 0.003	294.2 ± 18.3s	770.1 ± 7.2

Table: Performance of GAP-2 assignment condition across all agent ensembles. The balanced accuracy and wall time of all ensembles are significantly different from all other ensembles under a multiple comparisons test, $p < 1.0 \times 10^{-9}$.

Conclusions

- For naive assignment, a mixed ensemble increases the lower bound of accuracy over that of a computer vision ensemble
  - Results in a $100\times$ increase in wall time
- GAP conditions achieve or exceed the lower bound of accuracy for the naive mixed ensemble
  - Represent a significant speed-up over the naive parallel implementation (GAP-2: $1.6\times$, GAP-3: $1.4\times$)
  - Achieves rapid convergence by making fewer assignments
- In simulation, the mixed ensemble naive assignment condition significantly exceeds its lower bound (one-sample T-test, $p < 1.0 \times 10^{-9}$)
  - Simulated agents achieve true conditional independence
  - Unlikely to happen in real-world application
  - Indicates an increased importance of independent agents such as humans

References I

Bigdely-Shamlo, N., Vankov, A., Ramirez, R. R., and Makeig, S. (2008). Brain activity-based image classification from rapid serial visual presentation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 16(5):432–441.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28.

Fisher, M. L. (2004). The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, 50(12_supplement):1861–1871.

Ho, C.-j., Jabbari, S., and Vaughan, J. W. (2013). *Adaptive Task Assignment for Crowdsourced Classification*.

Joshi, A. J., Porikli, F., and Papanikolopoulos, N. P. (2012). Scalable active learning for multiclass image classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2259–2273.

References II

Kamar, E., Hacker, S., and Horvitz, E. (2012). Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 467–474. International Foundation for Autonomous Agents and Multiagent Systems.

Karger, D. R., Oh, S., and Shah, D. (2014). Budget-Optimal Task Allocation for Reliable Crowdsourcing Systems. *Operations Research*, 62(1):1–24.

Kundakcioglu, O. E. and Alizamir, S. (2008). Generalized Assignment Problem. In Floudas, C. A. and Pardalos, P. M., editors, *Encyclopedia of Optimization*, pages 1153–1162. Springer US.

Parisi, F., Strino, F., Nadler, B., and Kluger, Y. (2014). Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, 111(4):1253–1258.

Sajda, P., Pohlmeyer, E., Wang, J., Parra, L., Christoforou, C., Dmochowski, J., Hanna, B., Bahlmann, C., Singh, M., and Chang, S.-F. (2010). In a Blink of an Eye and a Switch of a Transistor: Cortically Coupled Computer Vision. *Proceedings of the IEEE*, 98(3):462–478.