Scalable and Sample-Efficient Active Learning for Graph-Based Classification

Kevin Miller

University of California, Los Angeles
Advisor: Dr. Andrea L. Bertozzi
Supported by NDSEG Research Fellowship

IMA Data Science Seminar

October 5, 2021

Overview

A Case for Semi-Supervised Learning

Our technology-rich and connected world produces lots of **Data**...

- Unlabeled Data : Inputs
  - **Easy to Collect/Generate**
- Labeled Data : Inputs + Outputs ("Labels")
  - **Difficult to Collect/Generate**

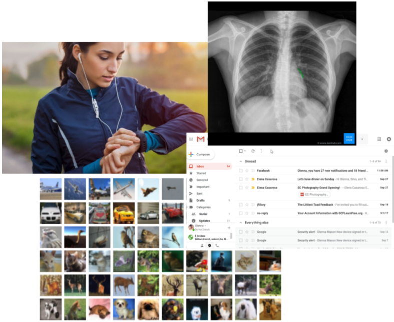


image credits: see references

Our technology-rich and connected world produces lots of **Data**...

- Unlabeled Data : Inputs
  - **Easy to Collect/Generate**
- Labeled Data : Inputs + Outputs ("Labels")
  - **Difficult to Collect/Generate**

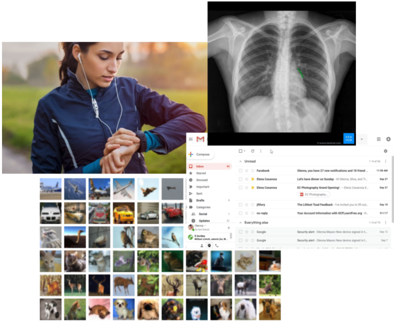


image credits: see references

Our technology-rich and connected world produces lots of **Data**...

- Unlabeled Data : Inputs
  - **Easy to Collect/Generate**
- Labeled Data : Inputs + Outputs ("Labels")
  - **Difficult to Collect/Generate**

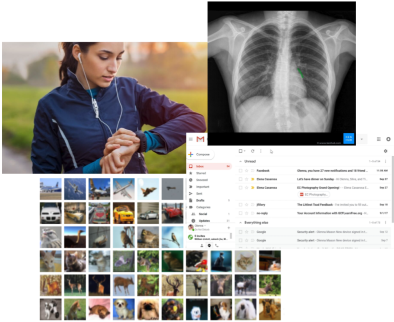
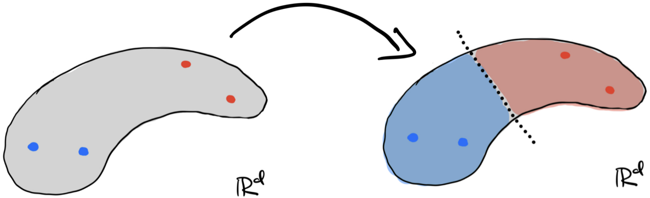
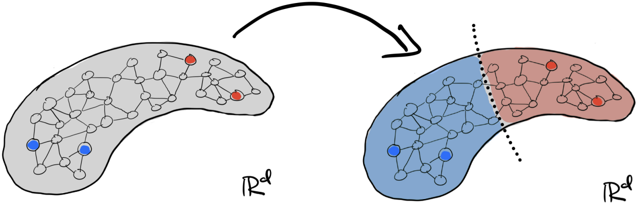


image credits: see references

Idea: *Given a small amount of labeled data, can I infer "accurate" labelings for the unlabeled data?*



Idea: *Given a small amount of labeled data* **and a similarity graph created from all inputs**, *can I infer "accurate" labelings for the unlabeled data?*



Great, you've leveraged using both labeled and unlabeled data!...

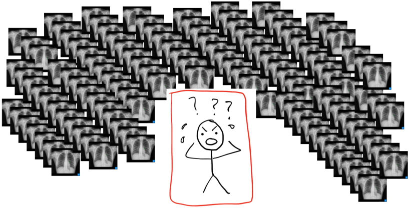
Why not try to improve?

Great, you've leveraged using both labeled and unlabeled data!...

Why not try to improve?

- Hand-label **the entire dataset**...

  COSTLY

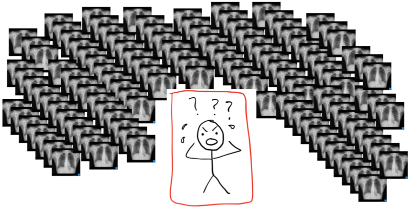


Great, you've leveraged using both labeled and unlabeled data!...

Why not try to improve?

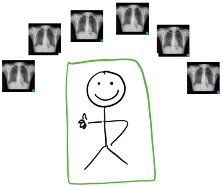
- Hand-label **the entire dataset**...

  COSTLY

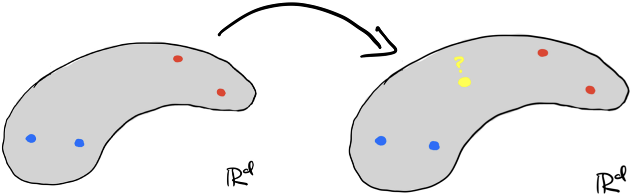


- Hand-label **only a few more**?

  DOABLE



Idea: *Given a small amount of labeled data, which unlabeled points would "best help" my semi-supervised learning classifier?*



Overview

1 Motivation

2 Problem Formulation and Graph-Based SSL Model

3 Model Change Active Learning

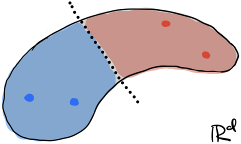
4 Further Insights and Applications

Observe *labeled data* $\mathcal{D}_\ell = \{(\mathbf{x}_i, y_i)\}_{i\in\mathcal{L}}$ and *unlabeled data* $\mathcal{X}_\mathcal{U} = \{\mathbf{x}_j\}_{j\in\mathcal{U}}$.

- $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} = \mathcal{X}_\mathcal{L} \cup \mathcal{X}_\mathcal{U}$
- $\mathcal{L}$: labeled indices, $\mathcal{U}$: unlabeled indices

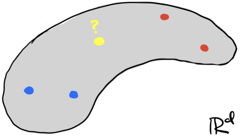
Semi-Supervised Learning

Given labeled data $\mathcal{L}$, can we accurately infer the labelings on $\mathcal{U}$?

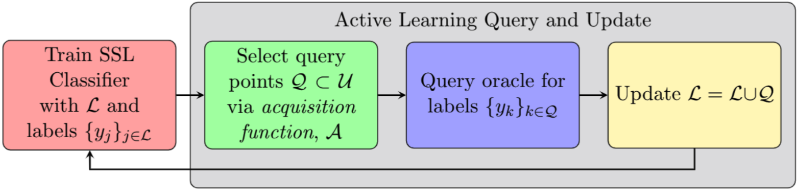


Active Learning

Given labeled data $\mathcal{L}$, can we judiciously "choose" unlabeled points $\mathcal{Q} \subset \mathcal{U}$ to label that will improve the output of the SSL model?



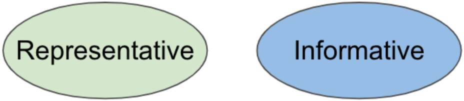
Active Learning Loop



Acquisition Function: *Criterion that quantifies the utility of labeling an unlabeled point* $k \in \mathcal{U}$.

Balancing Query Characteristics

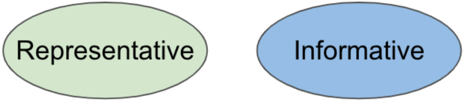
Active Learning – select "useful" points to label that will improve your classifier



- **Representative** : "looks" representative of the data
- **Informative** : help to refine the classifier's decision boundary

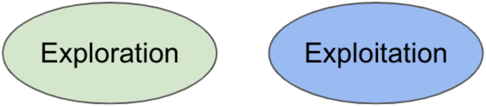
Balancing Query Characteristics

Active Learning – select "useful" points to label that will improve your classifier



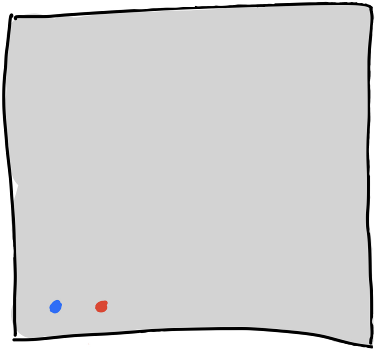
- **Representative** : "looks" representative of the data
- **Informative** : help to refine the classifier's decision boundary

---

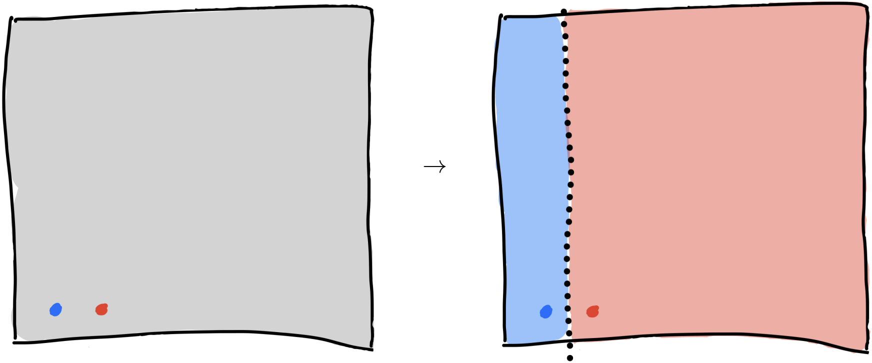


- **Exploration** : "explore" the inherent geometric/clustering structure
- **Exploitation** : "exploit" the classification structure that have learned so far

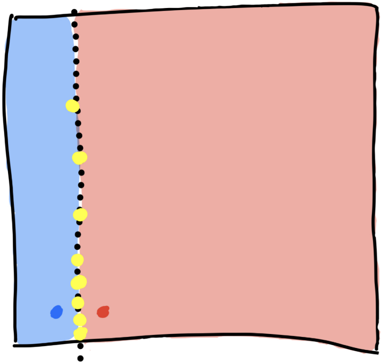
Exploration vs Exploitation Balance



Exploration vs Exploitation Balance

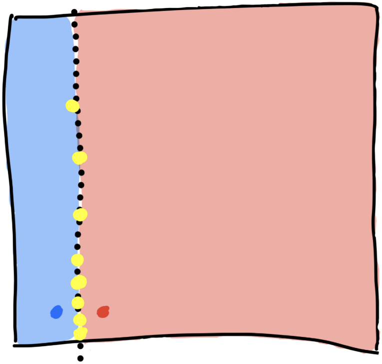


Potential SSL Classifier

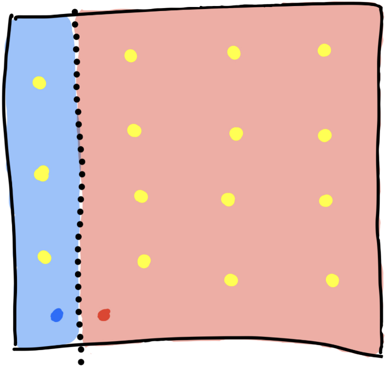


Exploitation

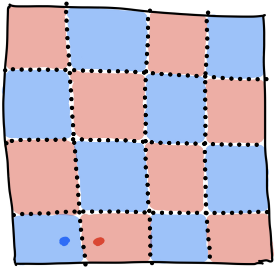
Exploration vs Exploitation Balance



Exploitation

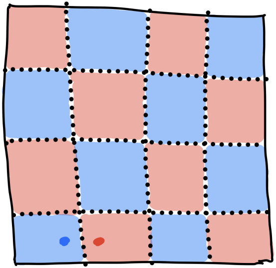


Exploration

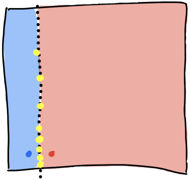


Ground Truth Boundaries

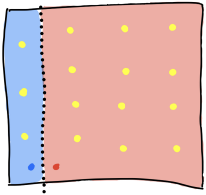
Exploration vs Exploitation Balance



Ground Truth Boundaries



Exploitation ✗



Exploration ✓

Given data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, construct *similarity graph* $G(Z, W)$, where

- $Z = \{1, 2, \ldots, N\}$
- $W_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$
- $d_i = \sum_{j \in Z} W_{ij}$
- degree matrix $D = \mathrm{diag}(d_1, d_2, \ldots, d_N)$

Graph Laplacians

- $L = D - W$, *unnormalized*
- $L_n = I - D^{-1/2} W D^{-1/2}$, *normalized*
- $L_{rw} = I - D^{-1} W$, *random walk*

Given data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, construct *similarity graph* $G(Z, W)$, where

- $Z = \{1, 2, \ldots, N\}$
- $W_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$
- $d_i = \sum_{j \in Z} W_{ij}$
- degree matrix $D = \mathrm{diag}(d_1, d_2, \ldots, d_N)$

Graph Laplacians

- $L = D - W$, *unnormalized*
- $L_n = I - D^{-1/2} W D^{-1/2}$, *normalized*
- $L_{rw} = I - D^{-1} W$, *random walk*

Useful Properties:

- Positive, semi-definite operators
- Eigenvectors encode clustering structure

Consider family of graph-based SSL models, using a perturbed *graph Laplacian* $L_\tau = L + \tau^2 I$:

$$\hat{\mathbf{u}} = \operatorname*{arg\,min}_{\mathbf{u}\in\mathbb{R}^N} \frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u}\rangle + \sum_{j\in\mathcal{L}} \ell(u_j, y_j) =: \operatorname*{arg\,min}_{\mathbf{u}\in\mathbb{R}^N} J_\ell(\mathbf{u};\mathbf{y}), \tag{1}$$

for different loss functions ℓ with parameter γ:

- $\ell(x,y) = (x-y)^2/2\gamma^2$, (Regression)
- $\ell(x,y) = \ln(1+e^{-xy/\gamma})$, (Logistic)
- $\ell(x,y) = -\ln \Psi_\gamma(xy)$, (Probit)

where $\Psi_\gamma(t) = \int_{-\infty}^{t} \psi_\gamma(s)ds$ is CDF of log-concave PDF $\psi_\gamma(s)$.

Multiclass Graph-Based SSL

With perturbed graph Laplacian L_τ and n_c the number of classes,

$$\hat{U} = \underset{U \in \mathbb{R}^{N \times n_c}}{\arg\min} \ \frac{1}{2}\langle U, L_\tau U\rangle_F + \sum_{j \in \mathcal{L}} \ell(\mathbf{u}^j, \mathbf{y}^j) =: \underset{U \in \mathbb{R}^{N \times n_c}}{\arg\min} \ \mathcal{J}_\ell(U; Y),$$

for different loss functions ℓ with parameter γ:

- $\ell(\mathbf{s}, \mathbf{t}) = \frac{1}{2\gamma^2}\|\mathbf{s} - \mathbf{t}\|_2^2$, (Multiclass Gaussian Regression)
- $\ell(\mathbf{s}, \mathbf{t}) = -\sum_{c=1}^{n_c} t_c \ln(s_c)$, (Cross-Entropy)

Optimizer $\hat{\mathbf{u}}$ can be viewed as *maximum a posteriori* (MAP) estimator

$$\arg\min_{\mathbf{u}} J_\ell(\mathbf{u};\mathbf{y}) \iff \arg\max_{\mathbf{u}} \exp(-J_\ell(\mathbf{u};\mathbf{y}))$$

$$= \arg\max_{\mathbf{u}} \underbrace{\exp\left(-\frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u}\rangle\right)}_{prior} \underbrace{\exp\left(-\sum_{j\in\mathcal{L}} \ell(u_j, y_j)\right)}_{likelihood}$$

$$= \arg\max_{\mathbf{u}} \mathbb{P}(\mathbf{u}|\mathbf{y})$$

for a posterior distribution $\mathbb{P}(\mathbf{u}|\mathbf{y}) \propto \exp(-J_\ell(\mathbf{u};\mathbf{y}))$.

- Different loss functions give different likelihoods

Harmonic Functions (HF) Model – AKA "Laplace Learning"

Assuming hard constraints for labeling[1], have conditional distribution:

$$\mathbf{u}_{\mathcal{U}}|\mathbf{y} \sim \mathcal{N}(\mathbf{u}_{hf}, L_{\mathcal{U},\mathcal{U}}^{-1}), \quad \mathbf{u}_{hf} = -L_{\mathcal{U},\mathcal{U}}^{-1} L_{\mathcal{U},\mathcal{L}} \mathbf{y}$$

with $\mathbf{u}_{\mathcal{L}} = \mathbf{y}$.

Gaussian Regression (GR) Model

With $\ell(x, y) = (x - y)^2 / 2\gamma^2$, then likelihood/prior/posterior is Gaussian.

$$\begin{aligned} \mathbb{P}(\mathbf{u}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u}\rangle\right) \exp\left(-\frac{1}{2\gamma^2} \sum_{j \in \mathcal{L}} (u_j - y_j)^2\right) \\ &\sim \mathcal{N}(\hat{\mathbf{u}}, C), \; \hat{\mathbf{u}} = \frac{1}{\gamma^2} C P^T \mathbf{y}, \; C^{-1} = L + \frac{1}{\gamma^2} P^T P, \end{aligned}$$

where $P : \mathbb{R}^N \to \mathbb{R}^{|\mathcal{L}|}$ is projection onto labeled set $\mathcal{L}$.

[1] Does not actually rigorously fit into Bayesian framework like others

Overview

Look-Ahead model with index k and label y_k:

$$\hat{\mathbf{u}}^{+k,y_k} := \operatorname*{arg\,min}_{\mathbf{u}\in\mathbb{R}^N} J^k(\mathbf{u};\mathbf{y},y_k) = \operatorname*{arg\,min}_{\mathbf{u}\in\mathbb{R}^N} \frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u}\rangle + \sum_{j\in\mathcal{L}} \ell(u_j, y_j) + \overbrace{\ell(u_k, y_k)}^{plus\ \ k}$$

- "hypothetical model", with $k \in \mathcal{U}$ and label y_k

For Gaussian model, look-ahead posterior distribution's parameters from the current posterior distribution

- *without expensive model retraining* – **rank-one updates**

GR: $\hat{\mathbf{u}}^{+k,y_k} = \hat{\mathbf{u}} + \frac{(y_k - \hat{u}_k)}{\gamma^2 + C_{kk}} C_{:,k}, \quad C^{+k,y_k} = C - \frac{1}{\gamma^2 + C_{kk}} C_{:,k} C_{:,k}^T$

Model Change: *How much would labeling $k \in \mathcal{U}$ change the classifier if we added it to the labeled set with pseudo-label $\hat{y}_k$?*

$$k^* = \underset{k \in \mathcal{U}}{\arg\max}\ \mathcal{A}(k) = \underset{k \in \mathcal{U}}{\arg\max}\ \|\hat{\mathbf{u}}^{+k,\hat{y}_k} - \hat{\mathbf{u}}\|_2$$

[2] Cai, Zhang, and Zhou, "Maximizing Expected Model Change for Active Learning in Regression", 2013; Karzand and Nowak, "MaxiMin Active Learning in Overparameterized Model Classes", 2020.

Model Change: *How much would labeling $k \in \mathcal{U}$ change the classifier if we added it to the labeled set with pseudo-label $\hat{y}_k$?*

$$k^* = \underset{k \in \mathcal{U}}{\arg\max}\ \mathcal{A}(k) = \underset{k \in \mathcal{U}}{\arg\max}\ \|\hat{\mathbf{u}}^{+k,\hat{y}_k} - \hat{\mathbf{u}}\|_2$$

Similar idea to previous works[2], but applied to a more general family of classifiers.

[2] Cai, Zhang, and Zhou, "Maximizing Expected Model Change for Active Learning in Regression", 2013; Karzand and Nowak, "MaxiMin Active Learning in Overparameterized Model Classes", 2020.

Model Change: *How much would labeling $k \in \mathcal{U}$ change the classifier if we added it to the labeled set with pseudo-label $\hat{y}_k$?*

$$k^* = \underset{k \in \mathcal{U}}{\arg\max}\ \mathcal{A}(k) = \underset{k \in \mathcal{U}}{\arg\max}\ \|\hat{\mathbf{u}}^{+k,\hat{y}_k} - \hat{\mathbf{u}}\|_2$$

Similar idea to previous works[2], but applied to a more general family of classifiers.

Other Acquisitions Using Look-Ahead:

- VOpt (Ji and Han, 2012): $\min\ Tr[C^{+k,y_k}]$
- Error Bound (Ji and Han, 2012): $\min\ Tr[(C^{+k,y_k})^2]$
- EER (Zhu et al, 2003): minimize expected error of look-ahead

All these use Gaussian models, i.e. look-ahead updates exact

[2] Cai, Zhang, and Zhou, "Maximizing Expected Model Change for Active Learning in Regression", 2013; Karzand and Nowak, "MaxiMin Active Learning in Overparameterized Model Classes", 2020.

Non-Gaussian Posterior?

When likelihood not Gaussian, posterior $\mathbb{P}(\mathbf{u}|\mathbf{y})$ *is non-Gaussian..*

Problems:

- model classifier as mean $\mu = \mathbb{E}_{\mathbf{u}\sim\mathbb{P}}[\mathbf{u}]$? or MAP estimator $\hat{\mathbf{u}} = \arg\max \mathbb{P}(\mathbf{u}|\mathbf{y})$?
- compute mean, μ, and covariance $C = \mathbb{E}_{\mathbf{u}\sim\mathbb{P}}\left[(\mathbf{u}-\mu)(\mathbf{u}-\mu)^T\right]$? (*potentially expensive!*)
- Look-ahead updates??

With non-Gaussian models, we lose these nice properties. *What to do?*

Non-Gaussian Posterior?

When likelihood not Gaussian, posterior $\mathbb{P}(\mathbf{u}|\mathbf{y})$ *is non-Gaussian..*

Problems:

- model classifier as mean $\mu = \mathbb{E}_{\mathbf{u}\sim\mathbb{P}}[\mathbf{u}]$? or MAP estimator $\hat{\mathbf{u}} = \arg\max \mathbb{P}(\mathbf{u}|\mathbf{y})$?
- compute mean, μ, and covariance $C = \mathbb{E}_{\mathbf{u}\sim\mathbb{P}}\left[(\mathbf{u}-\mu)(\mathbf{u}-\mu)^T\right]$? (*potentially expensive!*)
- Look-ahead updates??

With non-Gaussian models, we lose these nice properties. *What to do?*

Let's approximate with Gaussian, and see what happens!

Laplace approximation is a popular technique for approximating non-Gaussian distributions $\mathbb{P}$ with a Gaussian distribution.

$$\mathbf{x} \sim \mathcal{N}(\hat{\mathbf{x}}, \hat{C}), \quad \hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\max}\ \mathbb{P}(\mathbf{x}), \quad \hat{C} = \left(-\nabla^2 \ln(\mathbb{P}(\mathbf{x}))|_{\mathbf{x}=\hat{\mathbf{x}}}\right)^{-1},$$

where

- $\hat{\mathbf{x}}$: MAP estimator of $\mathbb{P}$
- $\hat{C}$: Hessian matrix of the negative-log density of $\mathbb{P}$, evaluated at $\hat{\mathbf{x}}$

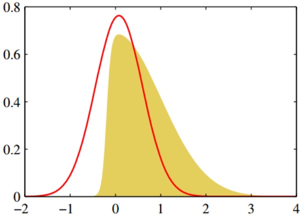


photo credit : http://wiljohn.top/2019/04/14/PRML4-4/

Laplace Approximation - Binary

$$\mathbf{u}|\mathbf{y} \sim \mathcal{N}(\hat{\mathbf{u}}, C_{\hat{\mathbf{u}}}), \qquad \hat{\mathbf{u}} = \operatorname*{arg\,min}_{\mathbf{u} \in \mathbb{R}^N} \; J_\ell(\mathbf{u}; \mathbf{y}),$$

and then calculate covariance of Laplace Approximation $C_{\hat{\mathbf{u}}}$

$$C_{\hat{\mathbf{u}}} = \left(\nabla^2_{\mathbf{u}} J_\ell(\hat{\mathbf{u}}; \mathbf{y})\right)^{-1} = \left(L + \sum_{j \in \mathcal{L}} F'(\hat{u}_j, y_j)\mathbf{e}_j\mathbf{e}_j^T\right)^{-1},$$

where

$$F(x, y) := \frac{\partial \ell}{\partial x}(x, y), \;\; F'(x, y) := \frac{\partial^2 \ell}{\partial x^2}(x, y).$$

How to approximate look-ahead model update, $\hat{\mathbf{u}}^{+k,\hat{y}_k} = \arg\min J_\ell^{k,\hat{y}_k}$?

- have $C_{\hat{\mathbf{u}}}$ (i.e. *inverse Hessian* evaluated at MAP estimator $\hat{\mathbf{u}}$)

Approximate Look-Ahead Update - Binary

How to approximate look-ahead model update, $\hat{\mathbf{u}}^{+k,\hat{y}_k} = \arg\min J_\ell^{k,\hat{y}_k}$?

- have $C_{\hat{\mathbf{u}}}$ (i.e. *inverse Hessian* evaluated at MAP estimator $\hat{\mathbf{u}}$)

Try one step of Newton's method, *starting at* $\hat{\mathbf{u}}$:

$$\begin{aligned}
\tilde{\mathbf{u}}^{+k,\hat{y}_k} &= \hat{\mathbf{u}} - \left(\nabla^2_{\mathbf{u}} J_\ell^{k,\hat{y}_k}(\hat{\mathbf{u}}; \mathbf{y}, \hat{y}_k)\right)^{-1} \left(\nabla_{\mathbf{u}} J_\ell^{k,\hat{y}_k}(\hat{\mathbf{u}}; \mathbf{y}, \hat{y}_k)\right) \\
&= \dots \\
&= \hat{\mathbf{u}} - \frac{F(\hat{u}_k, \hat{y}_k)}{1 + F'(\hat{u}_k, \hat{y}_k)[C_{\hat{\mathbf{u}}}]_{kk}} [C_{\hat{\mathbf{u}}}]_{:,k}
\end{aligned}$$

where

$$F(x,y) := \frac{\partial \ell}{\partial x}(x,y), \;\; F'(x,y) := \frac{\partial^2 \ell}{\partial x^2}(x,y).$$

Approximate Look-Ahead Update - Binary

How to approximate look-ahead model update, $\hat{\mathbf{u}}^{+k,\hat{y}_k} = \arg\min J_\ell^{k,\hat{y}_k}$?

- have $C_{\hat{\mathbf{u}}}$ (i.e. *inverse Hessian* evaluated at MAP estimator $\hat{\mathbf{u}}$)

Try one step of Newton's method, *starting at* $\hat{\mathbf{u}}$:

$$\begin{aligned}
\tilde{\mathbf{u}}^{+k,\hat{y}_k} &= \hat{\mathbf{u}} - \left(\nabla^2_{\mathbf{u}} J_\ell^{k,\hat{y}_k}(\hat{\mathbf{u}};\mathbf{y},\hat{y}_k)\right)^{-1}\left(\nabla_{\mathbf{u}} J_\ell^{k,\hat{y}_k}(\hat{\mathbf{u}};\mathbf{y},\hat{y}_k)\right) \\
&= \dots \\
&= \hat{\mathbf{u}} - \frac{F(\hat{u}_k,\hat{y}_k)}{1+F'(\hat{u}_k,\hat{y}_k)[C_{\hat{\mathbf{u}}}]_{kk}}[C_{\hat{\mathbf{u}}}]_{:,k}
\end{aligned}$$

where

$$F(x,y) := \frac{\partial \ell}{\partial x}(x,y),\ \ F'(x,y) := \frac{\partial^2 \ell}{\partial x^2}(x,y).$$

Simple update!

$*$ GR: this reduces to the **exact** look-ahead update!

Model Change Acquisition Function

Employ approximate update:

$$\begin{aligned}\mathcal{A}(k) &= \|\hat{\mathbf{u}}^{k,\hat{y}_k} - \hat{\mathbf{u}}\|_2 \approx \left\|\tilde{\mathbf{u}}^{k,\hat{y}_k} - \hat{\mathbf{u}}\right\|_2 \\ &= \left\|\frac{F(\hat{u}_k, \hat{y}_k)}{1 + F'(\hat{u}_k, \hat{y}_k)[C_{\hat{\mathbf{u}}}]_{kk}}[C_{\hat{\mathbf{u}}}]_{:,k}\right\|_2 \\ &= \left|\frac{F(\hat{u}_k, \hat{y}_k)}{1 + F'(\hat{u}_k, \hat{y}_k)[C_{\hat{\mathbf{u}}}]_{kk}}\right| \|[C_{\hat{\mathbf{u}}}]_{:,k}\|_2 .\end{aligned}$$

Employ approximate update:

$$\begin{aligned}\mathcal{A}(k) &= \|\hat{\mathbf{u}}^{k,\hat{y}_k} - \hat{\mathbf{u}}\|_2 \approx \left\|\tilde{\mathbf{u}}^{k,\hat{y}_k} - \hat{\mathbf{u}}\right\|_2 \\ &= \left\| \frac{F(\hat{u}_k, \hat{y}_k)}{1 + F'(\hat{u}_k, \hat{y}_k)[C_{\hat{\mathbf{u}}}]_{kk}} [C_{\hat{\mathbf{u}}}]_{:,k} \right\|_2 \\ &= \left| \frac{F(\hat{u}_k, \hat{y}_k)}{1 + F'(\hat{u}_k, \hat{y}_k)[C_{\hat{\mathbf{u}}}]_{kk}} \right| \left\|[C_{\hat{\mathbf{u}}}]_{:,k}\right\|_2 .\end{aligned}$$

Problem: $C_{\hat{\mathbf{u}}} = \left(L + \sum_{j\in\mathcal{L}} F'(\hat{u}_j, y_j)\mathbf{e}_j\mathbf{e}_j^T\right)^{-1} \in \mathbb{R}^{N\times N} \dots$

Consider only first $M < N$ eigenvalues and eigenvectors of graph Laplacian, L:

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_M, \quad \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M.$$

- $\Lambda_\tau = \mathrm{diag}\left(\lambda_1 + \tau^2, \ldots, \lambda_M + \tau^2\right)$
- $V = [\mathbf{v}_1 \ \ \mathbf{v}_2 \ \ \ldots \ \ \mathbf{v}_M] \in \mathbb{R}^{N \times M}$
- $\boldsymbol{\alpha} \in \mathbb{R}^M$ (binary), $A \in \mathbb{R}^{M \times n_c}$ (multiclass)

Consider only first $M < N$ eigenvalues and eigenvectors of graph Laplacian, L:

$$0 = \lambda_1 \le \lambda_2 \le \ldots \le \lambda_M, \quad \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M.$$

- $\Lambda_\tau = \mathrm{diag}\left(\lambda_1 + \tau^2, \ldots, \lambda_M + \tau^2\right)$
- $V = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \ldots \quad \mathbf{v}_M] \in \mathbb{R}^{N \times M}$
- $\boldsymbol{\alpha} \in \mathbb{R}^M$ (binary), $A \in \mathbb{R}^{M \times n_c}$ (multiclass)

Binary: $(\mathbf{u} = V\boldsymbol{\alpha})$

$$J_\ell(\mathbf{u}; \mathbf{y}) = \frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u} \rangle + \sum_{j \in \mathcal{L}} \ell(u_j, y_j)$$

$$\to \frac{1}{2}\langle \boldsymbol{\alpha}, \Lambda_\tau \boldsymbol{\alpha} \rangle + \sum_{j \in \mathcal{L}} \ell(\mathbf{e}_j^T V \boldsymbol{\alpha}, y_j) =: \tilde{J}_\ell(\boldsymbol{\alpha}; \mathbf{y}),$$

Consider only first $M < N$ eigenvalues and eigenvectors of graph Laplacian, L:

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_M, \quad \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M.$$

- $\Lambda_\tau = \mathrm{diag}\left(\lambda_1 + \tau^2, \ldots, \lambda_M + \tau^2\right)$
- $V = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \ldots \quad \mathbf{v}_M] \in \mathbb{R}^{N \times M}$
- $\boldsymbol{\alpha} \in \mathbb{R}^M$ (binary), $A \in \mathbb{R}^{M \times n_c}$ (multiclass)

Binary: $(\mathbf{u} = V\boldsymbol{\alpha})$

$$J_\ell(\mathbf{u}; \mathbf{y}) = \frac{1}{2}\langle \mathbf{u}, L_\tau \mathbf{u}\rangle + \sum_{j \in \mathcal{L}} \ell(u_j, y_j)$$

$$\to \frac{1}{2}\langle \boldsymbol{\alpha}, \Lambda_\tau \boldsymbol{\alpha}\rangle + \sum_{j \in \mathcal{L}} \ell(\mathbf{e}_j^T V \boldsymbol{\alpha}, y_j) =: \tilde{J}_\ell(\boldsymbol{\alpha}; \mathbf{y}),$$

Multiclass: $(U = VA)$

$$\mathcal{J}_\ell(U; Y) = \frac{1}{2}\langle U, L_\tau U\rangle_F + \sum_{j \in \mathcal{L}} \ell(\mathbf{u}^j, \mathbf{y}^j)$$

$$\to \frac{1}{2}\langle A, \Lambda_\tau A\rangle_F + \sum_{j \in \mathcal{L}} \ell(\mathbf{e}_j^T V A, \mathbf{y}^j) =: \tilde{\mathcal{J}}_\ell(A; Y).$$

Using covariance matrix (i.e. *inverse Hessian*) $\tilde{C}_{\hat{\boldsymbol{\alpha}}} = \left(\nabla^2_{\boldsymbol{\alpha}} \tilde{J}_\ell(\hat{\boldsymbol{\alpha}}; \mathbf{y})\right)^{-1}$ of the spectral truncation setup, we can apply approximate update as before:

$$\begin{aligned}
\mathcal{A}(k) &= \|\hat{\mathbf{u}}^{k,\hat{y}_k} - \hat{\mathbf{u}}\|_2 \approx \left\|\tilde{\mathbf{u}}^{k,\hat{y}_k} - \hat{\mathbf{u}}\right\|_2 \\
&= \left\|V\left(\tilde{\boldsymbol{\alpha}}^{k,\hat{y}_k} - \hat{\boldsymbol{\alpha}}\right)\right\|_2 \\
&= \left\|\tilde{\boldsymbol{\alpha}}^{k,\hat{y}_k} - \hat{\boldsymbol{\alpha}}\right\|_2 \\
&= \dots \\
&= \left|\frac{F(\hat{u}_k, \hat{y}_k)}{1 + F'(\hat{u}_k, \hat{y}_k)(\mathbf{v}^k)^T \tilde{C}_{\hat{\boldsymbol{\alpha}}} \mathbf{v}^k}\right| \left\|\tilde{C}_{\hat{\boldsymbol{\alpha}}} \mathbf{v}^k\right\|_2,
\end{aligned}$$

where we recall that $V\boldsymbol{\alpha} = \mathbf{u}$, so that

$$\hat{u}_k = \mathbf{e}_k^T V \hat{\boldsymbol{\alpha}} = (\mathbf{v}^k)^T \hat{\boldsymbol{\alpha}},$$

where $\mathbf{v}^k \in \mathbb{R}^M$ is the k^{th} *row* of V.

Similar result for multiclass case, but a little lengthy to describe...

Similar result for multiclass case, but a little lengthy to describe...

$$\tilde{A}^{+k,\hat{y}_k} = \hat{A} - \underbrace{\left(\nabla_A^2 \tilde{\mathcal{J}}^{k,\hat{y}_k}(\hat{A}; Y, \hat{\mathbf{y}}^k)\right)^{-1} \left(\nabla_A \tilde{\mathcal{J}}^{k,\hat{y}_k}(\hat{A}; Y, \hat{\mathbf{y}}^k)\right)}_{\text{simplifies to be rank } n_c}$$

Pixel Classification

- Seek to classify the pixels into classes (e.g. water, dirt, grass, metal, etc)
- Noisy measurements, corrupted by weather and atmospheric effects

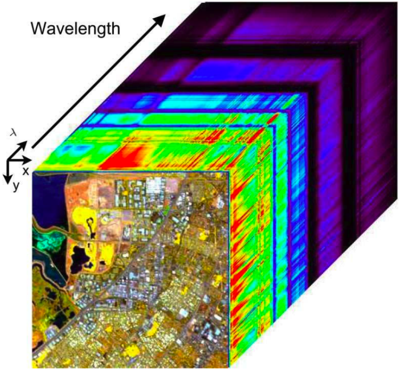


Figure 1: image credit: Christophe, Mailhes, & Duhamel (2009)

Pixel Classification

- Seek to classify the pixels into classes (e.g. water, dirt, grass, metal, etc)
- Noisy measurements, corrupted by weather and atmospheric effects

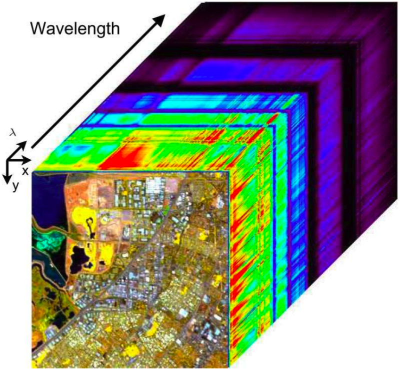


Figure 1: image credit: Christophe, Mailhes, & Duhamel (2009)

Apply active learning to incorporate human-in-the-loop to improve the accuracy of graph-based semi-supervised classification of pixels.

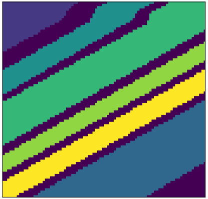


Figure 2: Salinas-A

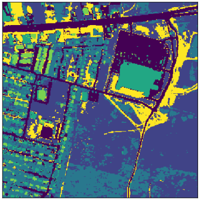


Figure 3: Urban

Graph Construction:

- 15 nearest neighbors, cosine similarity
- $M = 50$ eigenvalues

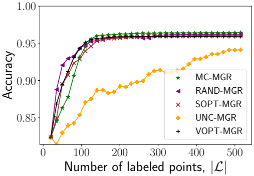
Experiments:

- Initially label 2 per class, select 500 points

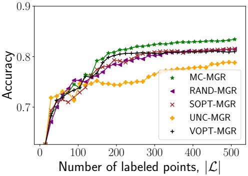
Acquisition Functions:

- Random
- Uncertainty
- VOpt (Ji and Han, 2012)
- Σ-Opt (Ma et al, 2013)
- **Model Change**

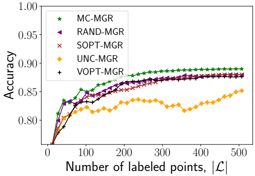
Multiclass GR Results:



(a) MNIST

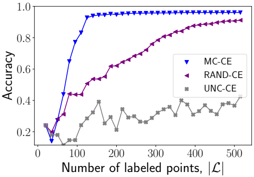


(b) Salinas A

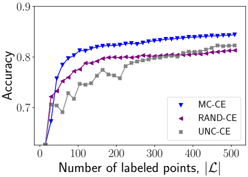


(c) Urban

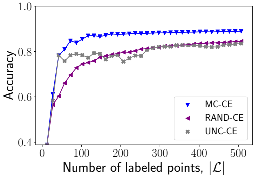
Cross-Entropy Results:



(d) MNIST



(e) Salinas A



(f) Urban

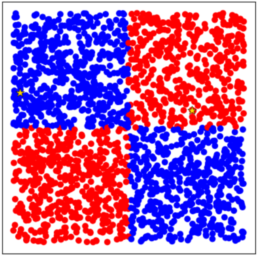
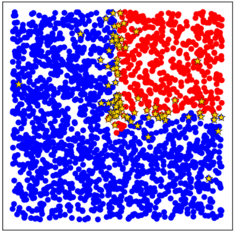


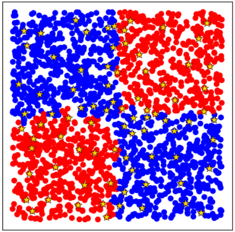
Figure 4: 2 x 2 Binary Checkerboard

2000 total points, 2 initially labeled points

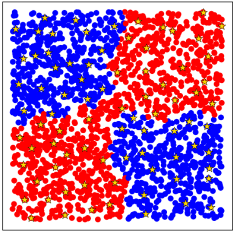
- Select 80 points sequentially via *Uncertainty, Model Change,* and *VOpt*.



Uncertainty

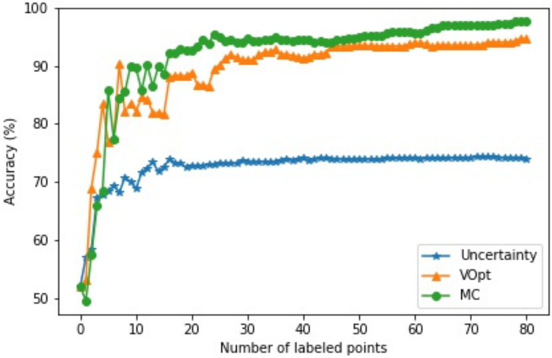


Model Change



VOpt

Accuracy Comparison



	Laplace Learning (HF)	Gaussian Regression
C	$L^{-1}_{\mathcal{U},\mathcal{U}}$	$\left(L + \frac{1}{\gamma^2} P^T P\right)^{-1}$
VOpt	$\frac{1}{C_{kk}} \lVert C_{:,k} \rVert_2^2$	$\frac{1}{\gamma^2 + C_{kk}} \lVert C_{:,k} \rVert_2^2$
Uncertainty	$\lvert \hat{u}_k - \hat{y}_k \rvert$	$\lvert \hat{u}_k - \hat{y}_k \rvert$
Model Change (MC)	$\frac{\lvert \hat{u}_k - \hat{y}_k \rvert}{C_{kk}} \lVert C_{:,k} \rVert_2$	$\frac{\lvert \hat{u}_k - \hat{y}_k \rvert}{\gamma^2 + C_{kk}} \lVert C_{:,k} \rVert_2$

	Laplace Learning (HF)	Gaussian Regression
C	$L^{-1}_{\mathcal{U},\mathcal{U}}$	$\left(L + \frac{1}{\gamma^2} P^T P\right)^{-1}$
VOpt	$\frac{1}{C_{kk}} \|C_{:,k}\|_2^2$	$\frac{1}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2^2$
Uncertainty	$\lvert\hat{u}_k - \hat{y}_k\rvert$	$\lvert\hat{u}_k - \hat{y}_k\rvert$
Model Change (MC)	$\frac{\lvert\hat{u}_k - \hat{y}_k\rvert}{C_{kk}} \|C_{:,k}\|_2$	$\frac{\lvert\hat{u}_k - \hat{y}_k\rvert}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2$

MCVOPT:

$$\mathcal{A}(k) = \underbrace{|\hat{u}_k - \hat{y}_k|}_{\text{"uncertainty"}} \underbrace{\frac{1}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2^2}_{\text{"kernel info"}}$$

	Laplace Learning (HF)	Gaussian Regression
C	$L_{\mathcal{U},\mathcal{U}}^{-1}$	$\left(L + \frac{1}{\gamma^2} P^T P\right)^{-1}$
VOpt	$\frac{1}{C_{kk}} \|C_{:,k}\|_2^2$	$\frac{1}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2^2$
Uncertainty	$\lvert\hat{u}_k - \hat{y}_k\rvert$	$\lvert\hat{u}_k - \hat{y}_k\rvert$
Model Change (MC)	$\frac{\lvert\hat{u}_k - \hat{y}_k\rvert}{C_{kk}} \|C_{:,k}\|_2$	$\frac{\lvert\hat{u}_k - \hat{y}_k\rvert}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2$

MCVOPT:
$$\mathcal{A}(k) = \underbrace{|\hat{u}_k - \hat{y}_k|}_{\text{“uncertainty”}} \underbrace{\frac{1}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2^2}_{\text{“kernel info”}}$$

Exploitation + Exploration

UCLA REUCAM 2021 Project – joint work with Dr. Jeffrey Calder (UMN)

- NGA NURI Grant #HM04762110003, (Dr. Andrea Bertozzi, PI)
- Undergraduates: Xoaquim Baca (Harvey Mudd), Jack Mauro (LMU), Jason Setiadi (UMN), Zhan Shi (UCLA)

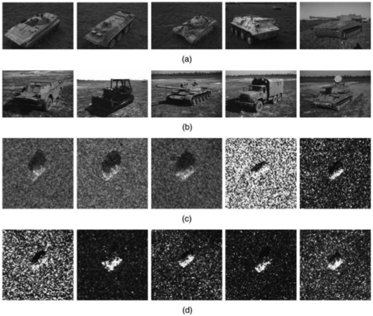


Fig. 2 MSTAR database. (a) and (b) Visible light images for BMP2, BTR70, T72, BTR60, 2S1, BRDM2, D7, T62, ZIL131, and ZSU23/4. (c) and (d) Corresponding SAR images for 10 targets measured at azimuth angle of 45 deg.

Figure 5: image credit: Perumal, Vasuki (2013)

MSTAR Dataset

- Synthetic Aperture Radar (SAR)
- Automatic Target Recognition (ATR)
- 6,784 images of size 88×88

SAR Data Pipeline

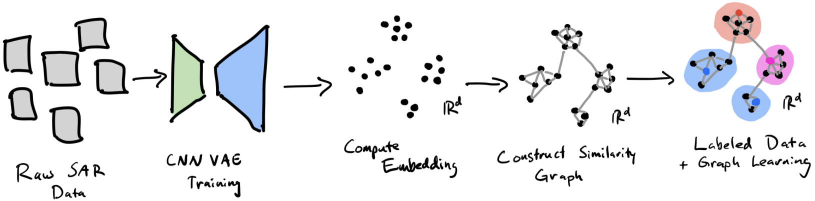


Figure 6: Unsupervised CNNVAE Representation Learning

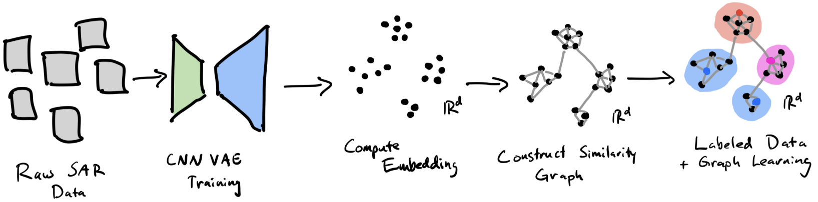


Figure 6: Unsupervised CNNVAE Representation Learning

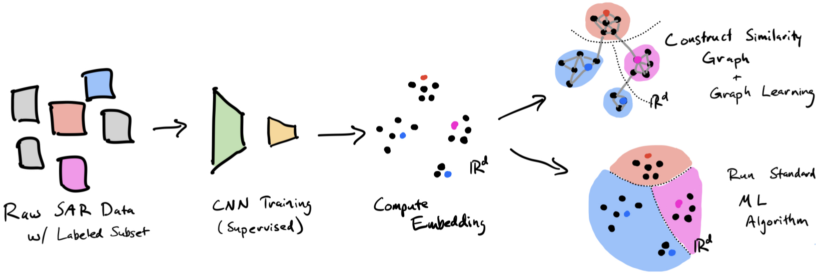


Figure 7: *Supervised* CNN Representation Learning

- **CNN**: 5%, 10%, 15%, ... training data, test various ML algorithms
  - "Upper bound" for capability of unsupervised representations?
- **CNN-VAE** : all training data, but **no label information**

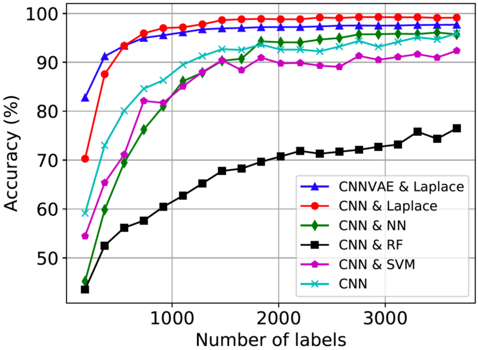


Figure 8: Performance of CNN vs CNNVAE representations with various ML algorithms

SSL Model: Laplace Learning

SSL Model: Laplace Learning

Active Learning Model for LL:

- Kernel: $K := L^{-1} \in \mathbb{R}^{N \times N}$
- Covariance: $C = L^{-1}_{\mathcal{U},\mathcal{U}}$

SSL Model: Laplace Learning

Active Learning Model for LL:

- Kernel: $K := L^{-1} \in \mathbb{R}^{N \times N}$
- Covariance: $C = L^{-1}_{\mathcal{U},\mathcal{U}} = K_{\mathcal{U},\mathcal{U}} - K_{\mathcal{U},\mathcal{L}} K^{-1}_{\mathcal{L},\mathcal{L}} K_{\mathcal{L},\mathcal{U}}$

SSL Model: Laplace Learning

Active Learning Model for LL:

- Kernel: $K := L^{-1} \in \mathbb{R}^{N \times N}$
- Covariance: $C = L^{-1}_{\mathcal{U},\mathcal{U}} = K_{\mathcal{U},\mathcal{U}} - K_{\mathcal{U},\mathcal{L}} K^{-1}_{\mathcal{L},\mathcal{L}} K_{\mathcal{L},\mathcal{U}}$

$K^{-1}_{\mathcal{L},\mathcal{L}}$ **not always invertible when using spectral truncation... unstable**

SSL Model: Laplace Learning

Active Learning Model for LL:

- Kernel: $K := L^{-1} \in \mathbb{R}^{N \times N}$
- Covariance: $C = L^{-1}_{\mathcal{U},\mathcal{U}} = K_{\mathcal{U},\mathcal{U}} - K_{\mathcal{U},\mathcal{L}} K^{-1}_{\mathcal{L},\mathcal{L}} K_{\mathcal{L},\mathcal{U}}$

$K^{-1}_{\mathcal{L},\mathcal{L}}$ **not always invertible when using spectral truncation... unstable**

Using GR model's covariance solves this instability issue!

$$C_{GR} = \left(L + \frac{1}{\gamma^2} P^T P\right)^{-1} = K - K_{:,\mathcal{L}} \underbrace{\left(K_{\mathcal{L},\mathcal{L}} + \gamma^2 I_{|\mathcal{L}|}\right)^{-1}}_{\text{invertible, even in sp. trunc.}} K_{\mathcal{L},:}$$

(Note Laplace Learning is $\gamma \to 0^+$ limit of GR)

With graph built from CNNVAE representations and *1 initially labeled point per class*, select 500 active learning query points sequentially.

With graph built from CNNVAE representations and *1 initially labeled point per class*, select 500 active learning query points sequentially.

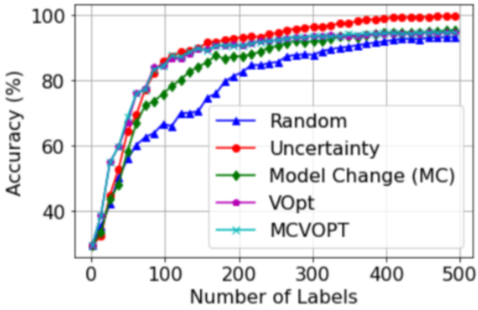


Figure 9: MSTAR Active Learning Results

Results:

Achieve 99.7% accuracy within 400 queries!

- *Best*: Uncertainty

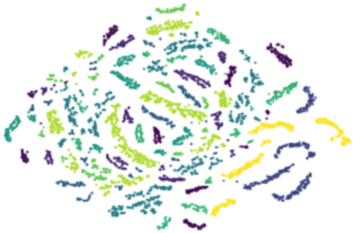
Previous slide max'd out at 97.7% after 3K labeled points

Uncertainty usually characterized as exploitative, suboptimal.. *Why did it perform so well?*

Uncertainty usually characterized as exploitative, suboptimal.. *Why did it perform so well?*

t-SNE Embedding Visualization

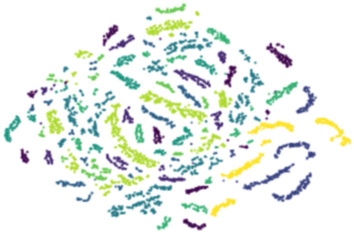
- Colored according to ground-truth classes
- Suggest natural clustering structure



Uncertainty usually characterized as exploitative, suboptimal.. *Why did it perform so well?*

t-SNE Embedding Visualization

- Colored according to ground-truth classes
- Suggest natural clustering structure



Laplace Learning Degeneracy

- "Spiky" behavior in low-label rates (Calder et al 2020)
- "Not confident" in unexplored clusters
  - Encourages exploration!

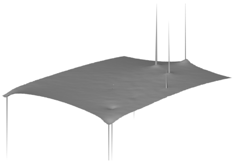
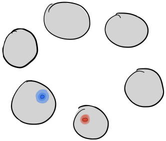


photo credit: Calder et al, 2020

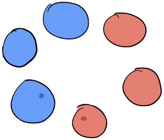
Some Takeaways

Need to Balance:



Conservative

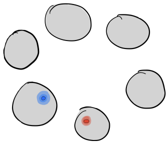
Model “Confidence”



Aggressive

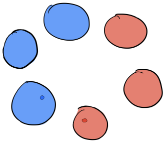
Some Takeaways

Need to Balance:



Conservative

Model "Confidence"



Aggressive

Acquisition Function Design

$$\frac{1}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2^2$$

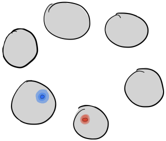
Exploration

$$|\hat{u}_k - \hat{y}_k|$$

Exploitation

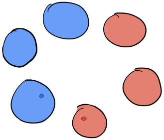
Some Takeaways

Need to Balance:



Conservative

Model "Confidence"



Aggressive

Acquisition Function Design

$$\frac{1}{\gamma^2 + C_{kk}} \|C_{:,k}\|_2^2$$

Exploration

$$|\hat{u}_k - \hat{y}_k|$$

Exploitation

Problem: *How to balance to get proper exploration vs exploitation tradeoff?*

Future Directions

- Exploration vs Exploitation
  - Mathematical definition for exploration?
  - When to “flip switch”?
  - Acquisition functions that naturally switch? (provably?)
  - Ad-hoc combinations

Future Directions

- Exploration vs Exploitation
  - Mathematical definition for exploration?
  - When to "flip switch"?
  - Acquisition functions that naturally switch? (provably?)
  - Ad-hoc combinations
- Accuracy curves are the **wrong metric** for comparison, I believe
  - Dataset-dependent quantity that captures exploration behavior?

Future Directions

- Exploration vs Exploitation
  - Mathematical definition for exploration?
  - When to "flip switch"?
  - Acquisition functions that naturally switch? (provably?)
  - Ad-hoc combinations
- Accuracy curves are the **wrong metric** for comparison, I believe
  - Dataset-dependent quantity that captures exploration behavior?
- Batch Learning – Is there a way that is efficient to select multiple query points at a time?
  - Coresets... but these lack human-in-the-loop
  - Submodular functions (VOPT)

Misc Image References

- https://hocview.com/fitness-tracker-that-does-not-require-a-smartphone-or-computer/
- https://www.kenhub.com/en/library/anatomy/normal-chest-x-ray
- https://edu.gcfglobal.org/en/gmail/introduction-to-gmail/1/
- https://www.cs.toronto.edu/~kriz/cifar.html

References I

Cai, Wenbin, Ya Zhang, and Jun Zhou. "Maximizing Expected Model Change for Active Learning in Regression". In: *2013 IEEE 13th International Conference on Data Mining*. ISSN: 2374-8486. Dec. 2013, pp. 51–60. DOI: 10.1109/ICDM.2013.104.

Calder, Jeff, Brendan Cook, et al. "Poisson Learning: Graph Based Semi-Supervised Learning At Very Low Label Rates". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1306–1316. URL: http://proceedings.mlr.press/v119/calder20a.html.

Calder, Jeff, Dejan Slepčev, and Matthew Thorpe. *Rates of Convergence for Laplacian Semi-Supervised Learning with Low Labeling Rates*. 2020. arXiv: 2006.02765 [math.ST].

Christophe, Emmanuel, Corinne Mailhes, and P Duhamel. "Hyperspectral image compression: Adapting SPIHT and EZW to anisotropic 3-D wavelet coding". In: *IEEE transactions on image processing* 17 (2009). a publication of the IEEE Signal Processing Society, pp. 2334–46.

Karzand, Mina and Robert D. Nowak. "MaxiMin Active Learning in Overparameterized Model Classes". In: *IEEE Journal on Selected Areas in Information Theory* 1.1 (May 2020). Conference Name: IEEE Journal on Selected Areas in Information Theory, pp. 167–177. ISSN: 2641-8770. DOI: 10.1109/JSAIT.2020.2991518.

Maaten, Laurens van der and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Miller, Kevin, Hao Li, and Andrea L Bertozzi. "Efficient Graph-Based Active Learning with Probit Likelihood via Gaussian Approximations". en. In: *ICML* Workshop on Real-World Experiment Design and Active Learning (2020).

Perumal, Vasuki. "Automatic target classification of manmade objects in synthetic aperture radar images using Gabor wavelet and neural network". In: *Journal of Applied Remote Sensing* 7 (2013).

Rasmussen, Carl Edward and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 2006. ISBN: 978-0-262-18253-9.

Settles, Burr. "Active Learning". en. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (June 2012), pp. 1–114. ISSN: 1939-4608, 1939-4616. DOI: 10.2200/S00429ED1V01Y201207AIM018. URL: http://www.morganclaypool.com/doi/abs/10.2200/S00429ED1V01Y201207AIM018 (visited on 06/11/2020).

Zhu, Xiaojin, Zoubin Ghahramani, and John Lafferty. "Semi-supervised learning using Gaussian fields and harmonic functions". In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. Washington, DC, USA: AAAI Press, Aug. 2003, pp. 912–919. ISBN: 978-1-57735-189-4. (Visited on 06/11/2020).

Zhu, Xiaojin, John Lafferty, and Zoubin Ghahramani. "Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions". In: *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*. 2003, pp. 58–65.