Practical applications of metric space magnitude and weighting vectors

Definition

A **metric space** is an ordered pair (M, d) where M is a set and d is a metric on M, i.e., a function

$$d\colon M \times M \to \mathbb{R}$$

such that for any $x, y, z \in M$, the following holds:[2]

1. $d(x, y) = 0 \Leftrightarrow x = y$ identity of indiscernibles
2. $d(x, y) = d(y, x)$ symmetry
3. $d(x, z) \leq d(x, y) + d(y, z)$ subadditivity or triangle inequality

Definition

Definition 1. Let X be a finite metric space with metric d. Denote the number of points in X by $|X|$. The *similarity matrix* of X is defined to be $\zeta_X(i,j) := \exp(-d(x_i, x_j))$ for $1 \leqslant i, j \leqslant |X|$. Whenever the inverse of ζ_X exists, we define the *weighting vector* of X to be

$$w_X := \zeta_X^{-1} \mathbb{1},$$

where $\mathbb{1}$ is the $|X| \times 1$ column vector of all ones. The *magnitude* of X is defined to be the quantity

$$\mathrm{Mag}(X) := \mathbb{1}^T w_X = \mathbb{1}^T \zeta_X^{-1} \mathbb{1}.$$

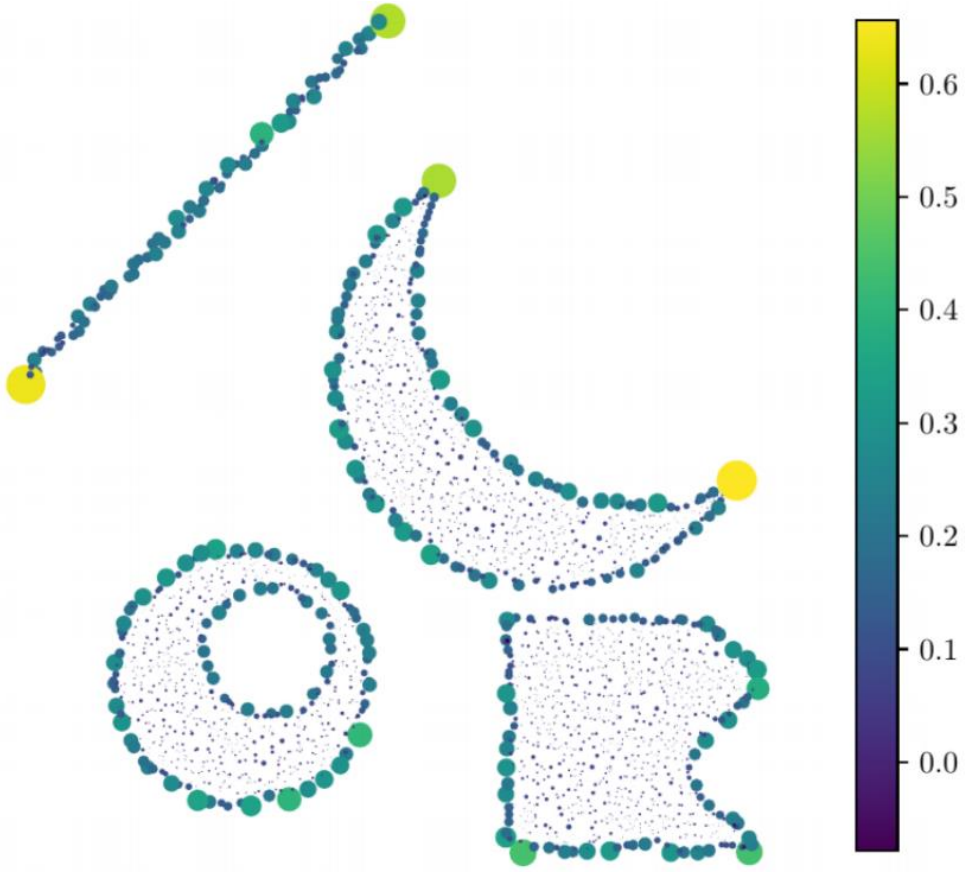
That is, $\mathrm{Mag}(X)$ is the sum of all the entries of the weighting vector w_X.

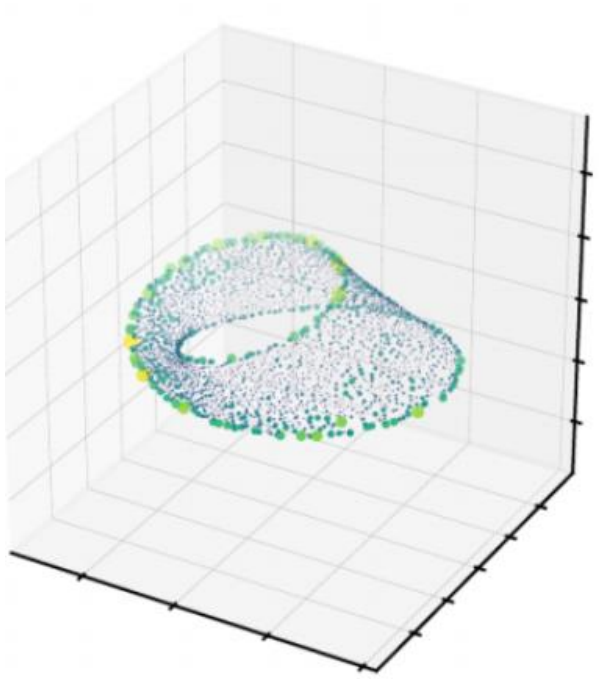
Example

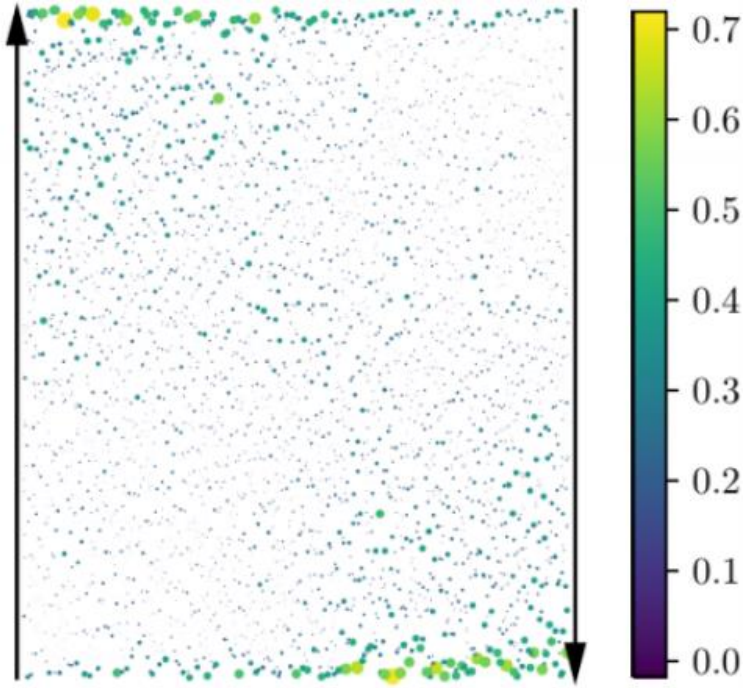
	-4	-1	0	1	9
-4	1	0.0498	0.0183	0.0067	2.26E-06
-1	0.0498	1	0.3679	0.1353	4.54E-05
0	0.0183	0.3679	1	0.3679	0.0001
1	0.0067	0.1353	0.3679	1	0.0003
4	2.26E-06	4.54E-05	0.0001	0.0003	1

	-4	-1	0	1	9
-4	1.0025	-0.0499	5.30E-18	-5.65E-19	-1.65E-22
-1	-0.0499	1.1590	-0.4254	-1.07E-17	-7.77E-23
0	3.23E-18	-0.4254	1.3130	-0.4255	1.73E-20
1	1.99E-26	-1.08E-24	-0.4255	1.1565	-0.0003
9	-5.94E-23	3.21E-21	1.77E-20	-0.0003	1
w	**0.952574**	**0.683633**	**0.462117**	**0.730723**	**0.999665**

Example







Classification

Algorithm 1 Classification via weighting vector

input Data set X, $L = \{L_1, L_2, ..., L_k\}$ labels, function DECIDE : $\mathbb{R}^k \rightarrow \{1, 2, ..., k\}$, function SCALE_i : $(\mathbb{R}, \mathbb{R}^{|X_i|}) \rightarrow \mathbb{R}$ for each $i \in \{1, 2, ..., k\}$

input unlabeled point x'

$p = []$

for $i \in \{1, 2, ..., k\}$ **do**

$Y = \{x'\} \cup X_i$

$w'_i = w_Y(x')$

$w = \text{SCALE}_i(w'_i, W_{X_i})$

$p.\text{append}(w)$

end for

let $j = \text{DECIDE}(p)$

output L_j

Classification

Table 1.

dataset	K-Neighbors	Logistic Reg.	Rand. Forest	SVM	Weight
2-d checkerboard	0.92 ± 0.02	0.51 ± 0.04	$\mathbf{0.94 \pm 0.01}$	0.62 ± 0.04	0.92 ± 0.01
clevedata.mat	0.82 ± 0.04	$\mathbf{0.85 \pm 0.02}$	0.82 ± 0.03	0.84 ± 0.03	0.84 ± 0.03
dimdata.mat	0.94 ± 0.01	0.95 ± 0.01	0.95 ± 0.00	$\mathbf{0.96 \pm 0.00}$	0.93 ± 0.01
housingdata.mat	0.87 ± 0.02	0.87 ± 0.03	0.87 ± 0.02	0.87 ± 0.03	0.87 ± 0.02
ionodata.mat	0.84 ± 0.05	0.89 ± 0.02	0.94 ± 0.02	$\mathbf{0.95 \pm 0.02}$	0.81 ± 0.08
iris	0.94 ± 0.04	0.87 ± 0.05	0.94 ± 0.04	$\mathbf{0.96 \pm 0.03}$	0.85 ± 0.13
sklearn digits	0.97 ± 0.01	0.96 ± 0.01	0.95 ± 0.01	$\mathbf{0.98 \pm 0.01}$	0.97 ± 0.00
ticdata.mat	0.85 ± 0.02	0.69 ± 0.03	$\mathbf{0.93 \pm 0.02}$	0.88 ± 0.02	0.78 ± 0.03

Active learning

Algorithm 2 Active learning via weighting vector

input Data set X,

$\mathcal{L} = \emptyset; \mathcal{U} = X$

initialize $\mathcal{L}$; $\mathcal{U} = X - \mathcal{L}$; with it's corresponding $\mathcal{Y}_{\mathcal{L}}$

$f = \textbf{train\_classifier}(\mathcal{L}, \mathcal{Y}_{\mathcal{L}})$

while (not converged) **or** (labeling budget not reached) **do**

$\tilde{X}_i = \{x \in X \mid f(x) = i\}$ for $i = 0, 1$.

calculate weighting vectors $w_{\tilde{X}_i}$

$Q_{\min,i} = \arg\min_{\mathcal{U}} \left| w_{\tilde{X}_i} \right|$ for $i = 0, 1$

$Q_{\max,i} = \arg\max_{\mathcal{U}} \left| w_{\tilde{X}_i} \right|$ for $i = 0, 1$

$\mathcal{Y}_{\mathcal{Q}}$=query_labels($Q_{\min,0}, Q_{\max,0}, Q_{\min,1}, Q_{\max,1}$)

$\mathcal{L} = \mathcal{L} \cup \{Q_{\min,0}, Q_{\max,0}, Q_{\min,1}, Q_{\max,1}\}$

$\mathcal{Y}_{\mathcal{L}} = \mathcal{Y}_{\mathcal{L}} \cup \mathcal{Y}_{\mathcal{Q}}$

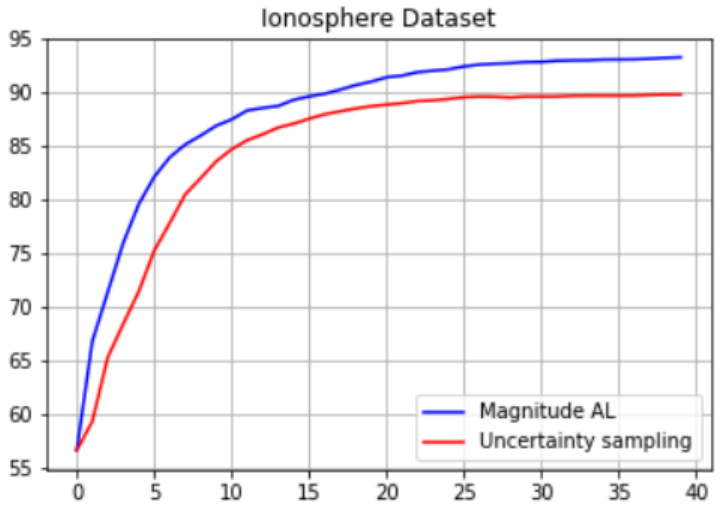
$\mathcal{U} = X - \mathcal{L}$;

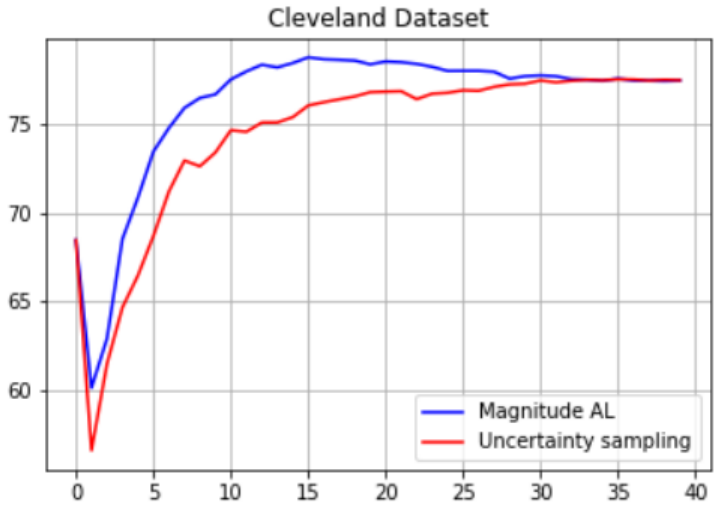
$f = \textbf{train\_classifier}(\mathcal{L}, \mathcal{Y}_{\mathcal{L}})$

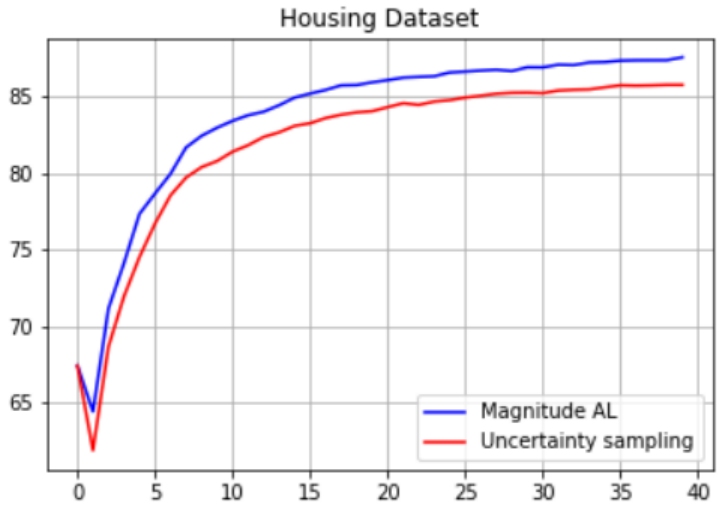
end while

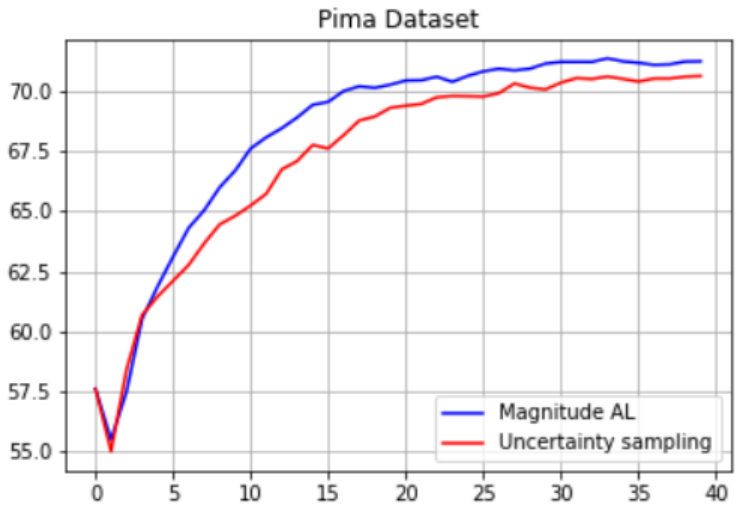
output f

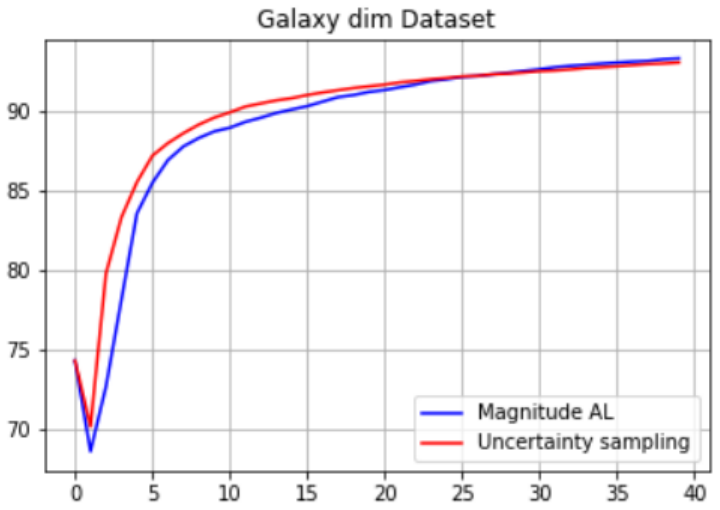
Active learning

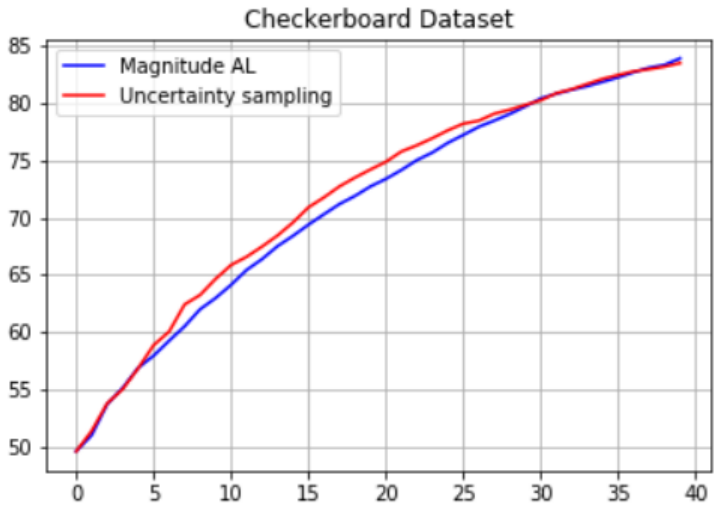












Outlier detection

Algorithm 3 Outlier detection via magnitude

input dataset X, threshold τ

$X_{in} = \{x \in X \mid \text{abs}(w_X(x)) < \text{median}(w_X) + 1.5\text{std}(w_X)\}$

$X_{out} = X \backslash X_{in}$

for $x \in X_{out}$ **do**

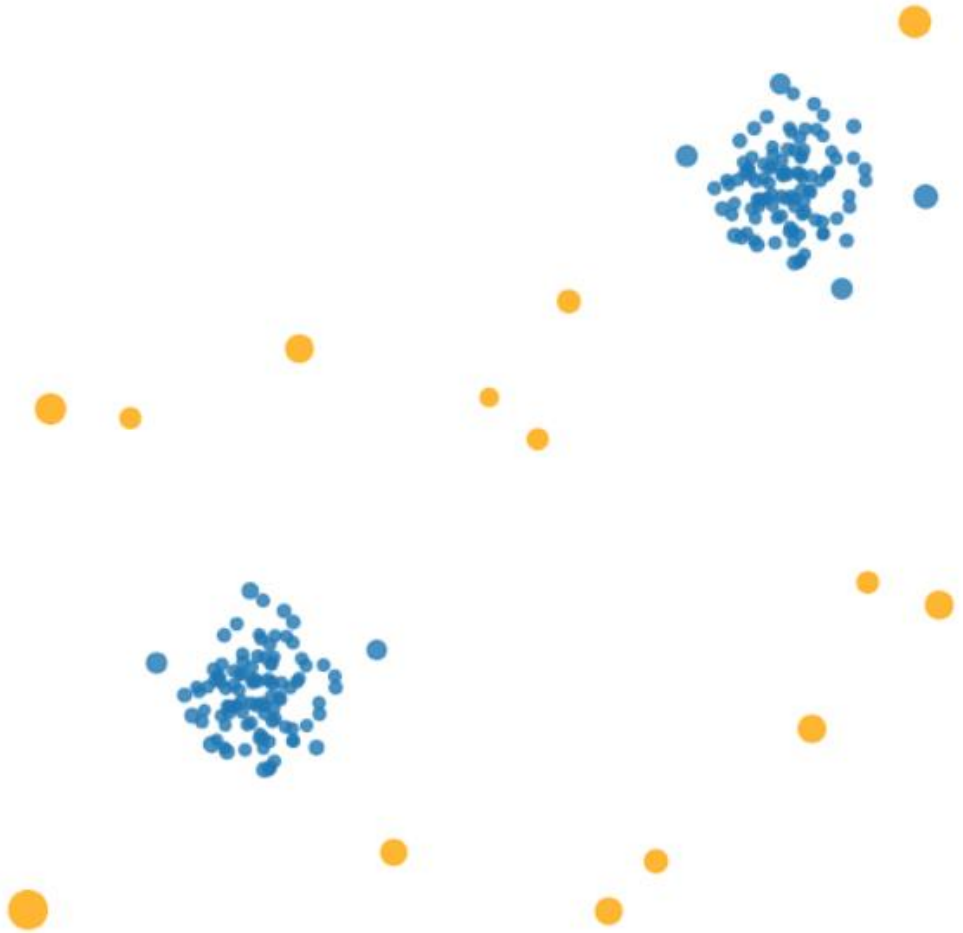
  if $\gamma_{Xx} < \tau$ **then**

    $X_{in} \leftarrow x$

  end if

end for

Outlier detection



Thanks