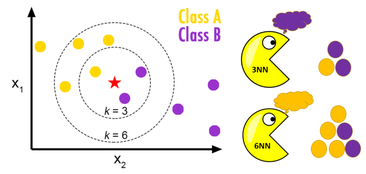
Rates of Convergence for Large-scale Nearest Neighbor Classification

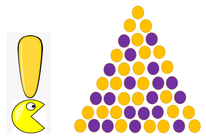
Xingye Qiao[1], *Jiexin Duan*[2], *Guang Cheng*[2]
Binghamton University[1], Purdue University[2]

Motivation

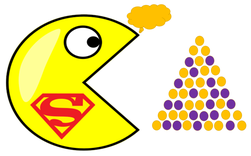
- k Nearest Neighbor Classifier (kNN)



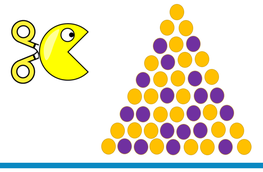
- Challenge of kNN for big data



- If we have a super computer (oracle kNN)
  - Large computational/space complexity
  - Expensive cost
  - Communication, privacy or ownership limitations

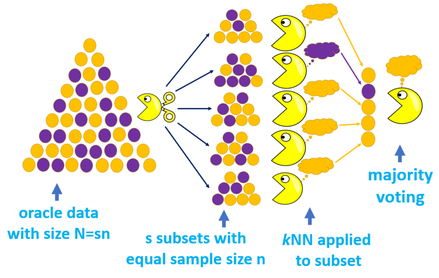


- Without a super computer, split the data!

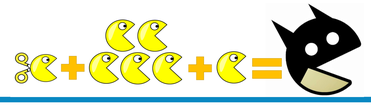


bigNN Classifier

- Divide & Conquer Framework



- Construction of Pac-**Bat**man (**B**ig D**at**a)



Statistical Guarantees of bigNN

- Primary criterion: classification accuracy
  - Regret=Expected Risk−Bayes Risk= $\mathbb{E}_{\mathcal{D}}\left[R(\widehat{\phi}_n)\right] - R(\phi^{\text{Bayes}})$
  - A small value of regret is preferred
- Secondary criterion: classification stability

Definition: Classification Instability (CIS)

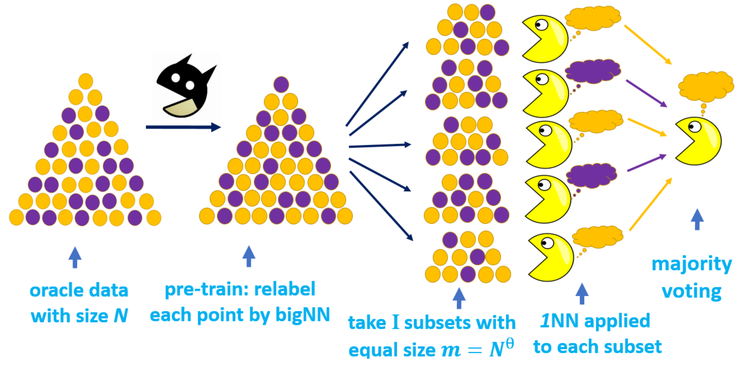
Define classification instability of a classification procedure Ψ as

$$\text{CIS}(\Psi) = \mathbb{E}_{\mathcal{D}_1,\mathcal{D}_2}\left[\mathbb{P}_X\left(\widehat{\phi}_{n1}(X) \neq \widehat{\phi}_{n2}(X)\right)\right],$$

where $\widehat{\phi}_{n1}$ and $\widehat{\phi}_{n2}$ are the classifiers trained from the same classification procedure Ψ based on $\mathcal{D}_1$ and $\mathcal{D}_2$ with the same distribution.

  - A small value of CIS is preferred
- Asymptotic comparison between bigNN and oracle kNN
  - **Theorem 1**: bigNN and oracle kNN have the same convergence rate of excess risk (regret)
  - **Theorem 2**: bigNN and oracle kNN have the same convergence rate of classification instability (CIS)

Denoised bigNN Classifier



- **Theorem 3**: Regret of denoised bigNN equals Regret of bigNN plus a relatively small additional error

Real Data Examples for bigNN

Data	size	dim	γ	R.BigNN	R.kNN	R.OWNN	C.BigNN	C.kNN	C.OWNN	Speedup
htru2	17898	8	0.1	2.0385	2.1105	2.1188	0.3670	0.6152	0.5528	2.72
htru2	17898	8	0.2	2.0929	2.1105	2.1188	0.6323	0.6152	0.5528	7.65
htru2	17898	8	0.3	2.1971	2.1105	2.1188	0.5003	0.6152	0.5528	21.65
gisette	6000	5000	0.2	3.9344	3.5020	3.4749	4.4261	4.4752	4.3317	5.13
musk1	476	166	0.1	14.7619	14.9767	14.9757	24.2362	23.0664	23.2707	1.79
musk2	6598	166	0.2	3.8250	3.4400	3.2841	4.7575	5.1925	4.1615	5.73
occup	20560	6	0.1	0.6207	0.6205	0.6037	0.3790	0.4431	0.5795	2.93
occup	20560	6	0.2	0.6119	0.6205	0.6037	0.3717	0.4431	0.5795	6.97
occup	20560	6	0.3	0.6548	0.6205	0.6037	0.3081	0.4431	0.5795	19.19
credit	30000	24	0.1	18.8300	18.8681	18.8414	2.7940	3.5292	3.4392	3.36
credit	30000	24	0.2	18.8467	18.8681	18.8414	4.3917	3.5292	3.4392	7.86
credit	30000	24	0.3	18.9250	18.8681	18.8414	4.2496	3.5292	3.4392	23.22
SUSY	$5000K$	18	0.1	19.3103	21.0381	20.7752	7.7034	7.4011	7.5921	4.59
SUSY	$5000K$	18	0.2	21.6149	21.0381	20.7752	7.9073	7.4011	7.5921	16.76
SUSY	$5000K$	18	0.3	22.3197	21.0381	20.7752	4.6716	7.4011	7.5921	88.22

Notes: Number of subsets $s = N^{\gamma}$. The prefix 'R.' means risk, and 'C.' means CIS (both in %). Speedup is defined as the computing time for oracle kNN divided by that for bigNN. OWNN is the oracle optimal weighted NN classifier.

Simulations

bigNN V.S. oracle kNN

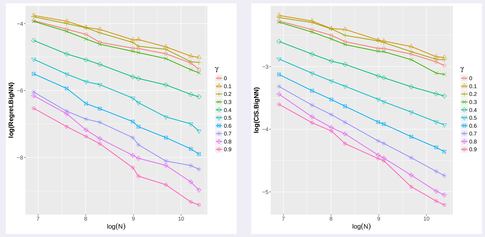


Figure 1: Regret and CIS for bigNN and oracle kNN ($\gamma = 0$). Different curves show different γ.

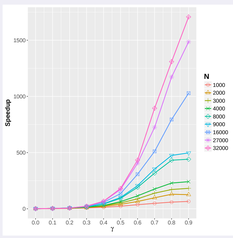


Figure 2: Speedup of bigNN, $\gamma = 0.0, 0.1 \ldots 0.9$. $\gamma = 0$ corresponds to the oracle kNN.

denoised bigNN V.S. bigNN

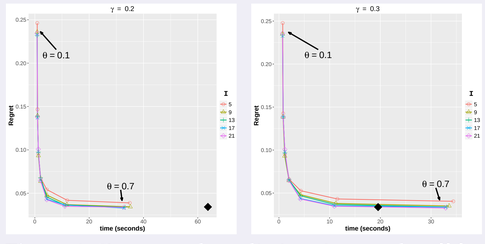


Figure 3: Regret and prediction time trade-off for denoised bigNN and bigNN (black diamonds). $\gamma = 0.2, 0.3$. $\theta = 0.1, 0.2, \ldots, 0.7$. Different curves show different I.

Future Work

- Explore the optimal splitting schemes for bigNN.
- Quantify the relative performance of two NN classifiers attaining the same rate (such as bigNN and oracle NN).
- Prove the sharp upper bound on γ.