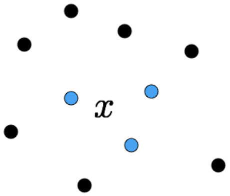
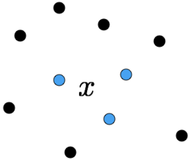
Best Practice and Statistical tradeoffs?



Samory Kpotufe

Statistics, Columbia University

Vanilla NN prediction:



Reduces to regression: let $f_k(x) = \text{avg}\,(Y_i)$ of k-NN(x)

... then: $h_k(x) \equiv \mathbb{1}\{f_k(x) \geq 1/2\}$.

Prediction Time: at least order k,

Irrespective of fast search method.

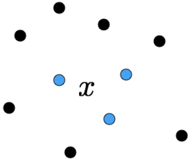
Unfortunately, optimal accuracy requires large $k = \Omega($root of$(n))$ ...

Vanilla NN prediction:

Regression:

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \mathbb{R}$.

Learn: $f_k(x) =$ average (Y_i) of k-NN(x).



Reduces to regression: let $f_k(x) =$ avg (Y_i) of k-NN(x)

... then: $h_k(x) \equiv \mathbb{1}\{f_k(x) \geq 1/2\}$.

Prediction Time: at least order k,

Irrespective of fast search method.

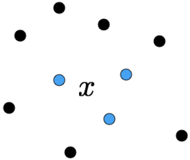
Unfortunately, optimal accuracy requires large $k = \Omega($**root of**$(n))$ **...**

Vanilla NN prediction:

Classification:

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $h_k(x) =$ majority (Y_i) of k-NN(x).



Reduces to regression: let $f_k(x) =$ avg (Y_i) of k-NN(x)

... then: $h_k(x) \equiv \mathbb{1}\{f_k(x) \geq 1/2\}$.

Prediction Time: at least order k,

Irrespective of fast search method.

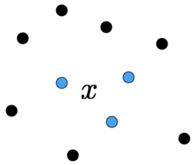
Unfortunately, optimal accuracy requires large $k = \Omega($root of$(n))$ ...

Vanilla NN prediction:

Classification:

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $h_k(x) =$ majority (Y_i) of k-NN(x).



Reduces to regression: let $f_k(x) =$ avg (Y_i) of k-NN(x)

... then: $h_k(x) \equiv \mathbb{1}\{f_k(x) \geq 1/2\}$.

Prediction Time: at least order k,

Irrespective of fast search method.

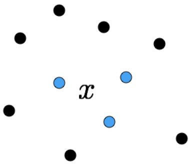
Unfortunately, optimal accuracy requires large $k = \Omega($root of$(n))$ ...

Vanilla NN prediction:

Classification:

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $h_k(x) =$ majority (Y_i) of k-NN(x).



Reduces to regression: let $f_k(x) =$ avg (Y_i) of k-NN(x)

... then: $h_k(x) \equiv \mathbb{1}\{f_k(x) \geq 1/2\}$.

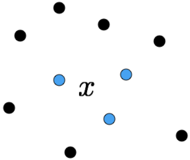
Prediction Time: at least order k,

Irrespective of fast search method.

Unfortunately, optimal accuracy requires large $k = \Omega($root of$(n))$ ...

Vanilla NN prediction:

Classification:



Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $h_k(x) =$ majority (Y_i) of k-NN(x).

Reduces to regression: let $f_k(x) =$ avg (Y_i) of k-NN(x)

... then: $h_k(x) \equiv \mathbb{1}\{f_k(x) \geq 1/2\}$.

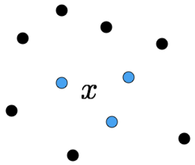
Prediction Time: at least order k,

Irrespective of fast search method.

Unfortunately, optimal accuracy requires large $k = \Omega($root of$(n))$ ...

Vanilla NN prediction:

Classification:



Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $h_k(x) =$ majority (Y_i) of k-NN(x).

Reduces to regression: let $f_k(x) =$ avg (Y_i) of k-NN(x)

... then: $h_k(x) \equiv \mathbb{1}\{f_k(x) \geq 1/2\}$.

Prediction Time: at least order k,

Irrespective of fast search method.

Unfortunately, optimal accuracy requires large $k = \Omega($**root of**$(n))$ **...**

Statistical performance of k-NN:

Consider regression: $Y = f(X) + \text{noise}$, $\dim(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d} \text{ minimized at } k \propto n^{2/(2+d)}$$

Same story for classification ...

So for optimal accuracy, prediction time = $\Omega(n^{2/(2+d)})$
(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Statistical performance of k-NN:

Consider regression: $Y = f(X) + \text{noise}$, $\dim(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d} \quad \text{minimized at } k \propto n^{2/(2+d)}$$

Same story for classification ...

So for optimal accuracy, prediction time $= \Omega(n^{2/(2+d)})$
(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Statistical performance of k-NN:

Consider regression: $Y = f(X) + \text{noise}$, $\dim(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d}$$ minimized at $k \propto n^{2/(2+d)}$

Same story for classification ...

So for optimal accuracy, prediction time $= \Omega(n^{2/(2+d)})$
(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Statistical performance of k-NN:

Consider regression: $Y = f(X) + \text{noise}$, $\dim(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d}$$ minimized at $k \propto n^{2/(2+d)}$

Same story for classification ...

So for optimal accuracy, prediction time $= \Omega(n^{2/(2+d)})$
(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Statistical performance of k-NN:

Consider regression: $Y = f(X) + \text{noise}$, $\dim(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d} \text{ minimized at } k \propto n^{2/(2+d)}$$

Same story for classification ...

So for optimal accuracy, prediction time = $\Omega(n^{2/(2+d)})$

(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Statistical performance of k-NN:

Consider regression: $Y = f(X) + \text{noise}$, $\dim(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d}$$ minimized at $k \propto n^{2/(2+d)}$

Same story for classification ...

So for optimal accuracy, prediction time $= \Omega(n^{2/(2+d)})$

(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Statistical performance of k-NN:

Consider regression: $Y = f(X) +$ noise, dim$(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d}$$ minimized at $k \propto n^{2/(2+d)}$

Same story for classification ...

So for optimal accuracy, prediction time = $\Omega(n^{2/(2+d)})$

(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Statistical performance of k-NN:

Consider regression: $Y = f(X) + \text{noise}$, $\dim(X) = d$

Suppose $f(x) \doteq \mathbb{E}[Y|x]$ is Lipschitz:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d} \text{ minimized at } k \propto n^{2/(2+d)}$$

Same story for classification ...

So for optimal accuracy, prediction time = $\Omega(n^{2/(2+d)})$

(Irrespective of fast proximity search)

Our goal: optimal accuracy with prediction time $= O(\log n)$

Fast prediction with no tradeoff:

How to achieve this:

Data quantization or Sub-sampling + (simple Variance correction)

We'll consider common NN approaches:

ϵ-**NN:** use all samples ϵ-close to x

k-**NN:** use the k closest samples to x

Fast prediction with no tradeoff:

How to achieve this:

Data quantization or Sub-sampling + (simple Variance correction)

We'll consider common NN approaches:

ϵ-**NN:** use all samples ϵ-close to x

k-**NN:** use the k closest samples to x

Fast prediction with no tradeoff:

How to achieve this:

Data quantization or Sub-sampling + (simple Variance correction)

We'll consider common NN approaches:

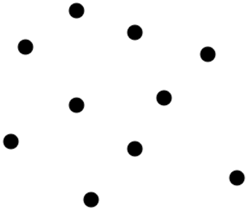
ϵ-**NN:** use all samples ϵ-close to x

k-**NN:** use the k closest samples to x

Outline:

- NN and Data Quantization
- NN and Subsampling
- Overview and Open Questions

Quantization: *reduce the data*



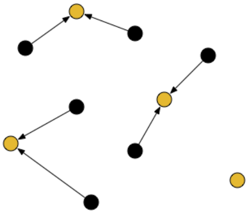
$\{X_i\}_{i=1}^n$

Two options: Pick k closest q's to x or Pick all q's in $B(x, \epsilon)$.

Main issues:

Size of Q ... How to choose Q ... How to use Q

Quantization: *reduce the data*



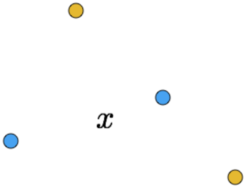
Assign $\{X_i\}$ to representatives $\mathbf{Q} \equiv \{q\}$

Two options: Pick k closest q's to x or Pick all q's in $B(x, \epsilon)$.

Main issues:

Size of $\mathbf{Q}$ ... How to choose $\mathbf{Q}$ ... How to use $\mathbf{Q}$

Quantization: *reduce the data*



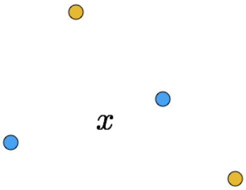
Pick q's in $\mathbf{Q}$ close to x

Two options: Pick k closest q's to x or Pick all q's in $B(x, \epsilon)$.

Main issues:

Size of $\mathbf{Q}$ ... How to choose $\mathbf{Q}$ ... How to use $\mathbf{Q}$

Quantization: *reduce the data*



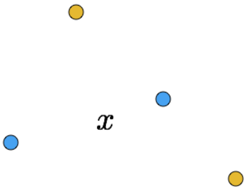
Pick q's in $\mathbf{Q}$ close to x

Two options: Pick k closest q's to x or Pick all q's in $B(x, \epsilon)$.

Main issues:

Size of $\mathbf{Q}$ ... How to choose $\mathbf{Q}$ ... How to use $\mathbf{Q}$

Quantization: *reduce the data*



Pick q's in $\mathbf{Q}$ close to x

Two options: Pick k closest q's to x or Pick all q's in $B(x, \epsilon)$.

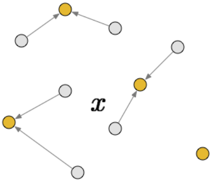
Main issues:

Size of $\mathbf{Q}$ ... How to choose $\mathbf{Q}$ ... How to use $\mathbf{Q}$

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

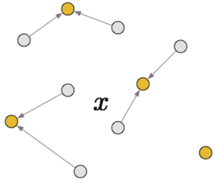
Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$



We'll make a few changes for the guarantees we want ...

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

Pick $\mathbf{Q}$ to **(1)** have small size, and **(2)** be close to $\{X_i\}$ ...



Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$

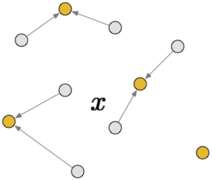
We'll make a few changes for the guarantees we want ...

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

Pick $\mathbf{Q}$ to **(1)** have small size, and **(2)** be close to $\{X_i\}$ ...

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$



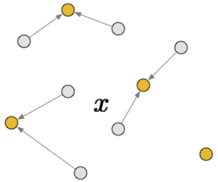
We'll make a few changes for the guarantees we want ...

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

Pick $\mathbf{Q}$ to **(1)** have small size, and **(2)** be close to $\{X_i\}$ ...

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$



We'll make a few changes for the guarantees we want ...

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

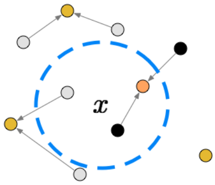
Pick $\mathbf{Q}$ to **(1)** have small size, and **(2)** be close to $\{X_i\}$ ...

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$

$f_{\mathbf{Q}}(x) =$ avg (Y_q) of q's in $B(x, \epsilon)$

$h_{\mathbf{Q}}(x) = \mathbb{1}\{f_{\mathbf{Q}}(x) \geq 1/2\}$.



We'll make a few changes for the guarantees we want ..

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

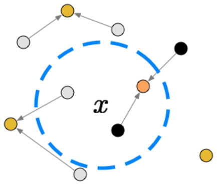
Pick $\mathbf{Q}$ to **(1)** have small size, and **(2)** be close to $\{X_i\}$ ...

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$

$f_{\mathbf{Q}}(x) =$ avg (Y_q) of q's in $B(x, \epsilon)$

$h_{\mathbf{Q}}(x) = \mathbb{1}\{f_{\mathbf{Q}}(x) \geq 1/2\}$.



We'll make a few changes for the guarantees we want ..

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

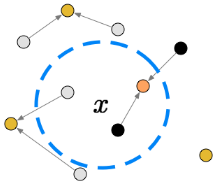
Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$

$f_{\mathbf{Q}}(x) =$ avg (Y_q) of q's in $B(x, \epsilon)$

$h_{\mathbf{Q}}(x) = \mathbb{1}\{f_{\mathbf{Q}}(x) \geq 1/2\}$.



We'll make a few changes for the guarantees we want ..

ϵ-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

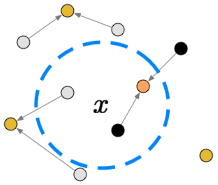
Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: $Y_q \equiv$ avg (Y_i) of $\{X_i \to q\}$

$f_{\mathbf{Q}}(x) =$ weighted avg (Y_q) of q's in $B(x, \epsilon)$

$h_{\mathbf{Q}}(x) = \mathbb{1}\{f_{\mathbf{Q}}(x) \geq 1/2\}$.



We'll make a few changes for the guarantees we want ..

Intuition: Suppose $(\mathcal{X}, \rho)$ has doubling dimension d

Relate $f_{\mathbf{Q}}$ to ϵ-NN f_ϵ (on n samples) ...

Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

$$f_{\mathbf{Q}}(x) = \frac{1}{\sum n_q} \sum_{q \in B(x,\epsilon)} n_q Y_q$$

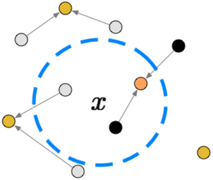
- Has variance $O(1/\sum n_q)$ rather than $O(1/\min n_q)$

Argue that $\sum n_q \geq |\{X_i\} \cap B(x, (1-\alpha)\epsilon)|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

Intuition: Suppose $(\mathcal{X}, \rho)$ has doubling dimension d

Relate $f_\mathbf{Q}$ to ϵ-NN f_ϵ (on n samples) ...

Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.



$$f_\mathbf{Q}(x) = \frac{1}{\sum n_q} \sum_{q \in B(x,\epsilon)} n_q Y_q$$

- Has variance $O(1/\sum n_q)$ rather than $O(1/\min n_q)$

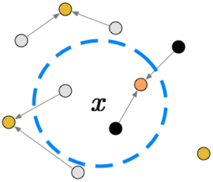
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

Intuition: Suppose $(\mathcal{X}, \rho)$ has doubling dimension d

Relate $f_{\mathbf{Q}}$ to ϵ-NN f_ϵ (on n samples) ...

Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

- $\mathbf{Q} \cap B(x, \epsilon)$ is small (of size $O(\alpha^{-d})$)



$$f_{\mathbf{Q}}(x) = \frac{1}{\sum n_q} \sum_{q \in B(x,\epsilon)} n_q Y_q$$

- Has variance $O(1/\sum n_q)$ rather than $O(1/\min n_q)$

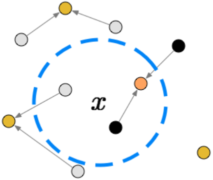
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

Intuition: Suppose $(\mathcal{X}, \rho)$ has doubling dimension d

Relate $f_{\mathbf{Q}}$ to ϵ-NN f_ϵ (on n samples) ...

Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

- $\mathbf{Q} \cap B(x, \epsilon)$ is small (of size $O(\alpha^{-d})$)
- Relevant X_i's are 2ϵ-close to x ($\approx$ bias of f_ϵ)



$$f_{\mathbf{Q}}(x) = \frac{1}{\sum n_q} \sum_{q \in B(x,\epsilon)} n_q Y_q$$

- Has variance $O(1/\sum n_q)$ rather than $O(1/\min n_q)$

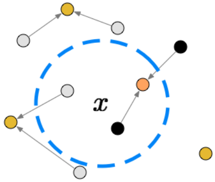
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

Intuition: Suppose $(\mathcal{X}, \rho)$ has doubling dimension d

Relate $f_{\mathbf{Q}}$ to ϵ-NN f_ϵ (on n samples) ...

Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

- $\mathbf{Q} \cap B(x, \epsilon)$ is small (of size $O(\alpha^{-d})$)
- Relevant X_i's are 2ϵ-close to x ($\approx$ bias of f_ϵ)



$$f_{\mathbf{Q}}(x) = \frac{1}{\sum n_q} \sum_{q \in B(x,\epsilon)} n_q Y_q$$

- Has variance $O(1/\sum n_q)$ rather than $O(1/\min n_q)$

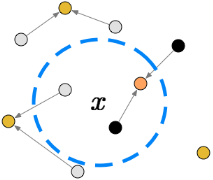
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

Intuition: Suppose $(\mathcal{X}, \rho)$ has doubling dimension d

Relate $f_{\mathbf{Q}}$ to ϵ-NN f_ϵ (on n samples) ...

Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

- $\mathbf{Q} \cap B(x, \epsilon)$ is small (of size $O(\alpha^{-d})$)
- Relevant X_i's are 2ϵ-close to x ($\approx$ bias of f_ϵ)



$$f_{\mathbf{Q}}(x) = \frac{1}{\sum n_q} \sum_{q \in B(x,\epsilon)} n_q Y_q$$

- Has variance $O(1/\sum n_q)$ rather than $O(1/\min n_q)$

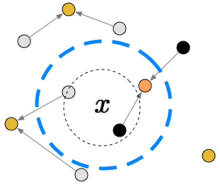
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

Intuition: Suppose $(\mathcal{X}, \rho)$ has doubling dimension d

Relate $f_{\mathbf{Q}}$ to ϵ-NN f_ϵ (on n samples) ...

Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

- $\mathbf{Q} \cap B(x, \epsilon)$ is small (of size $O(\alpha^{-d})$)
- Relevant X_i's are 2ϵ-close to x ($\approx$ bias of f_ϵ)



$$f_{\mathbf{Q}}(x) = \frac{1}{\sum n_q} \sum_{q \in B(x,\epsilon)} n_q Y_q$$

- Has variance $O(1/\sum n_q)$ rather than $O(1/\min n_q)$

Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

Guarantees: [Kpo., Verma, 17]

Assume a fast-range search procedure for $\mathbf{Q} \cap B(x, \epsilon)$ ...

Theorem. For appropriate choice of ϵ:

- f_Q (or h_Q) can be computed in time $O(\log(n) + \alpha^{-d})$.
- The excess risk of f_Q (or h_Q) is of optimal order $n^{-1/(2+d)}$.

Guarantees: [Kpo., Verma, 17]

Assume a fast-range search procedure for $\mathbf{Q} \cap B(x, \epsilon)$ ...

Theorem. For appropriate choice of ϵ:

- f_Q (or h_Q) can be computed in time $O(\log(n) + \alpha^{-d})$.
- The excess risk of f_Q (or h_Q) is of optimal order $n^{-1/(2+d)}$.

Guarantees: [Kpo., Verma, 17]

Assume a fast-range search procedure for $\mathbf{Q} \cap B(x, \epsilon)$ ...

Theorem. For appropriate choice of ϵ:

- f_Q (or h_Q) can be computed in time $O(\log(n) + \alpha^{-d})$.
- The excess risk of f_Q (or h_Q) is of optimal order $n^{-1/(2+d)}$.

Guarantees: [Kpo., Verma, 17]

Assume a fast-range search procedure for $\mathbf{Q} \cap B(x, \epsilon)$ ...

Theorem. For appropriate choice of ϵ:

- f_Q (or h_Q) can be computed in time $O(\log(n) + \alpha^{-d})$.
- The excess risk of f_Q (or h_Q) is of optimal order $n^{-1/(2+d)}$.

Table: $\frac{\epsilon\text{-NN Error}}{\text{Quantization Error}}$ vs $\frac{\epsilon\text{-NN Time}}{\text{Quantization Time}}$

Datasets	SARCOS (42k)	CT Slices (51k)	MiniBooNE (128k)
$\alpha = 1/6$	0.99 - 2.03	0.93 - 1.29	0.99 - 1.17
$\alpha = 2/6$	**0.99 - 4.10**	0.92 - 2.04	0.99 - 1.65
$\alpha = 3/6$	**0.98 - 6.31**	**0.91 - 3.17**	**0.99 - 4.05**
$\alpha = 4/6$	**0.96 - 7.70**	**0.91 - 5.40**	**0.98 - 6.42**
$\alpha = 5/6$	0.89 - 9.26	0.85 - 11.94	**0.94 - 8.83**
$\alpha = 6/6$	0.77 - 10.14	0.43 - 15.33	0.88 - 10.22

As $\alpha \nearrow$, Error of $f_{\mathbf{Q}} \nearrow$, but Prediction Time $\searrow$

Main downside of Quantization:

Computing $\mathbf{Q}$ can be $O(n^2)$.

Also, it's unclear how to choose $\mathbf{Q}$ for k-NN rather than ϵ-NN ...

Main downside of Quantization:

Computing $\mathbf{Q}$ can be $O(n^2)$.

Also, it's unclear how to choose $\mathbf{Q}$ for k-NN rather than ϵ-NN ...

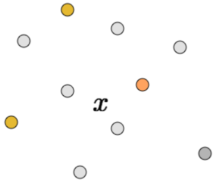
Outline:

- NN and Data Quantization
- NN and Subsampling
- Overview and Open Questions

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]:

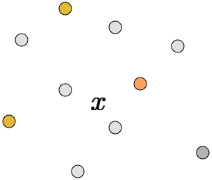
- Large $N \implies$ reduce variance.
- **Tradeoff on** m: small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]:

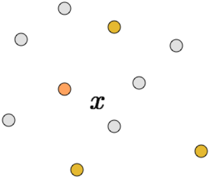
- Large $N \implies$ reduce variance.
- **Tradeoff on** m**:** small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]*:*

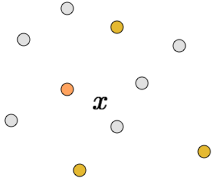
- Large $N \implies$ reduce variance.
- **Tradeoff on** m: small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]:

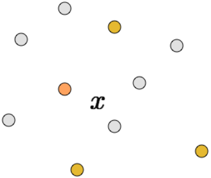
- Large $N \implies$ reduce variance.
- **Tradeoff on** m**:** small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0,1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]:

- Large $N \implies$ reduce variance.
- **Tradeoff on** m**:** small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.

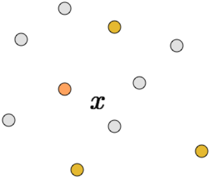
Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$

$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$



Desired N, m [Biau et al. 2010] [Samworth 2010]:

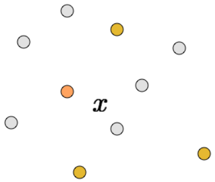
- Large $N \implies$ reduce variance.
- **Tradeoff on** m: small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]*:*

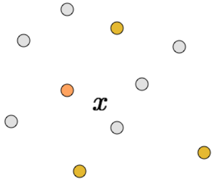
- Large $N \implies$ reduce variance.
- **Tradeoff on** m**:** small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]*:*

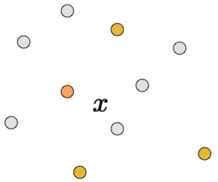
- Large $N \implies$ reduce variance.
- **Tradeoff on** m**:** small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n \to \infty} 0$.

Subsampling: *reduce data and parallelize*

Data: $\{(X_i, Y_i)\}_{i=1}^n$, $Y \in \{0, 1\}$.

Learn: N subsamples $\{S_t\}$ of size $m \ll n$



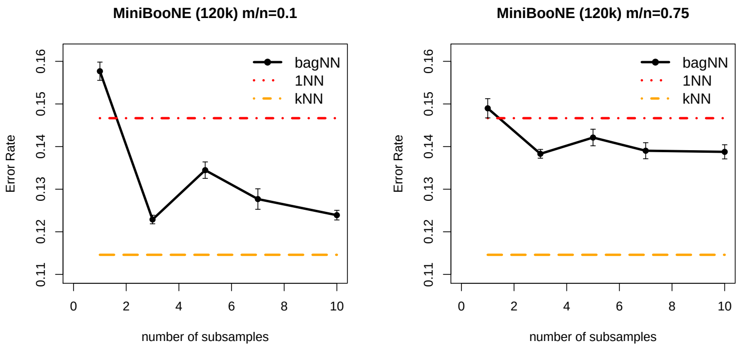
$Y_t(x) \leftarrow Y$-value of 1-NN(x) in S_t

$h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}$

Desired N, m [Biau et al. 2010] [Samworth 2010]*:*

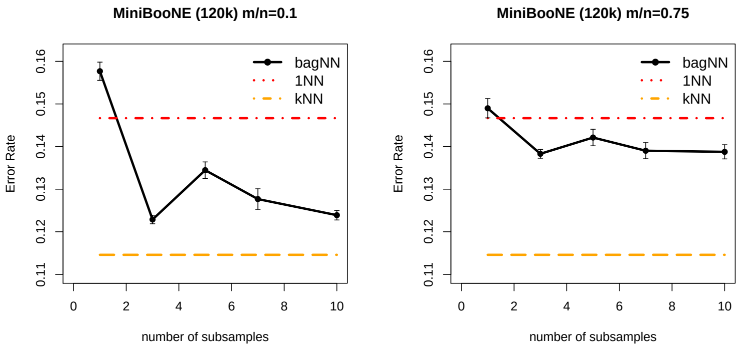
- Large $N \implies$ reduce variance.
- **Tradeoff on** m**:** small $m \implies$ richer $\{S_t\}$, but more variance.

Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n\to\infty} 0$.



Rule of Thumb: Pick $(m/n) \approx 10\%$ (often most accurate).

2 to 8 times speedup over k-NN prediction time



Rule of Thumb: Pick $(m/n) \approx 10\%$ (often most accurate).

2 to 8 times speedup over k-NN prediction time

But can we get accuracy $\approx$ that of k-NN?

[Biau et al. 2010] [Samworth 2010]: Yes, as $N \to \infty$

We want high accuracy for small N:

Correct the variance in each subsample ...

Variant (subNN): replace all Y_i by $h_k(X_i)$

[Xue, Kpo., 17]

But can we get accuracy $\approx$ that of k-NN?

[Biau et al. 2010] [Samworth 2010]: Yes, as $N \to \infty$

We want high accuracy for small N:

Correct the variance in each subsample ...

Variant (subNN): replace all Y_i by $h_k(X_i)$

[Xue, Kpo., 17]

But can we get accuracy $\approx$ that of k-NN?

[Biau et al. 2010] [Samworth 2010]: Yes, as $N \to \infty$

We want high accuracy for small N:

Correct the variance in each subsample ...

Variant (subNN): replace all Y_i by $h_k(X_i)$
[Xue, Kpo., 17]

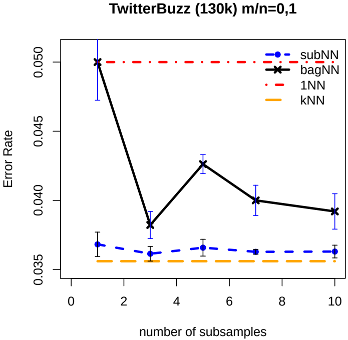
But can we get accuracy $\approx$ that of k-NN?

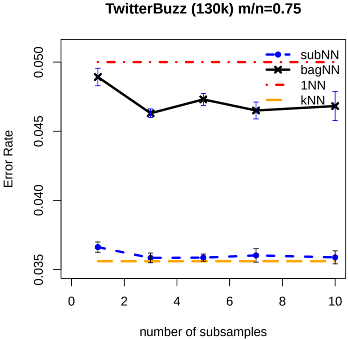
[Biau et al. 2010] [Samworth 2010]: Yes, as $N \to \infty$

We want high accuracy for small N:

Correct the variance in each subsample ...

Variant (subNN): replace all Y_i by $h_k(X_i)$
[Xue, Kpo., 17]





Error is now close to that of k-NN while maintaining 2-8 times speedup.

Guarantees for subNN:

Suppose P_X is doubling (i.e., $P_X(B(x,r)) \gtrsim r^d$), and $E[Y|x]$ is Lipschitz

Theorem. For a good choice of $k = k(n)$,

- Parallel computation time is no more than that of (fast) 1-NN

- The Excess Error is at most $\text{OPT}_k(n) + m^{-1/d}$

OPT $m = \text{root}(n)$ and we can let $m/n \to 0$.

Intuition: let $N = 1$, and $S(x) \doteq \text{NN}(x)$ in subsample S,

$$h_{\text{sub}}(x) \leftarrow h_k(S(x)) \text{ now}$$

$$h_k(S(x)) \approx h^*(S(x)) \approx h^*(x) + \text{distance}(x, S(x))$$

Guarantees for subNN:

Suppose P_X is doubling (i.e., $P_X(B(x,r)) \gtrsim r^d$), and $E[Y|x]$ is Lipschitz

Theorem. For a good choice of $k = k(n)$,

- Parallel computation time is no more than that of (fast) 1-NN
- The Excess Error is at most $\text{OPT}_k(n) + m^{-1/d}$

OPT $m = \text{root}(n)$ and we can let $m/n \to 0$.

Intuition: let $N = 1$, and $S(x) \doteq \text{NN}(x)$ in subsample S,

$h_{\text{sub}}(x) \leftarrow h_k(S(x))$ now

$h_k(S(x)) \approx h^*(S(x)) \approx h^*(x) + \text{distance}(x, S(x))$

Guarantees for subNN:

Suppose P_X is doubling (i.e., $P_X(B(x,r)) \gtrsim r^d$), and $E[Y|x]$ is Lipschitz

Theorem. For a good choice of $k = k(n)$,

- Parallel computation time is no more than that of (fast) 1-NN
- The Excess Error is at most $\text{OPT}_k(n) + m^{-1/d}$

OPT $m = \text{root}(n)$ and we can let $m/n \to 0$.

Intuition: let $N = 1$, and $S(x) \doteq \text{NN}(x)$ in subsample S,

$h_{\text{sub}}(x) \leftarrow h_k(S(x))$ now

$h_k(S(x)) \approx h^*(S(x)) \approx h^*(x) + \text{distance}(x, S(x))$

Guarantees for subNN:

Suppose P_X is doubling (i.e., $P_X(B(x,r)) \gtrsim r^d$), and $E[Y|x]$ is Lipschitz

Theorem. For a good choice of $k = k(n)$,

- Parallel computation time is no more than that of (fast) 1-NN
- The Excess Error is at most $\text{OPT}_k(n) + m^{-1/d}$

OPT $m = \text{root}(n)$ and we can let $m/n \to 0$.

Intuition: let $N = 1$, and $S(x) \doteq \text{NN}(x)$ in subsample S,

$$h_{\text{sub}}(x) \leftarrow h_k(S(x)) \text{ now}$$

$$h_k(S(x)) \approx h^*(S(x)) \approx h^*(x) + \text{distance}(x, S(x))$$

Guarantees for subNN:

Suppose P_X is doubling (i.e., $P_X(B(x,r)) \gtrsim r^d$), and $E[Y|x]$ is Lipschitz

Theorem. For a good choice of $k = k(n)$,

- Parallel computation time is no more than that of (fast) 1-NN
- The Excess Error is at most $\mathrm{OPT}_k(n) + m^{-1/d}$

OPT $m = \mathrm{root}(n)$ and we can let $m/n \to 0$.

Intuition: let $N = 1$, and $S(x) \doteq \mathrm{NN}(x)$ in subsample S,

$$h_{\mathrm{sub}}(x) \leftarrow h_k(S(x)) \text{ now}$$

$$h_k(S(x)) \approx h^*(S(x)) \approx h^*(x) + \mathrm{distance}(x, S(x))$$

Guarantees for subNN:

Suppose P_X is doubling (i.e., $P_X(B(x,r)) \gtrsim r^d$), and $E[Y|x]$ is Lipschitz

Theorem. For a good choice of $k = k(n)$,

- Parallel computation time is no more than that of (fast) 1-NN
- The Excess Error is at most $\text{OPT}_k(n) + m^{-1/d}$

OPT $m = \text{root}(n)$ and we can let $m/n \to 0$.

Intuition: let $N = 1$, and $S(x) \doteq \text{NN}(x)$ in subsample S,

$$h_{\text{sub}}(x) \leftarrow h_k(S(x)) \text{ now}$$

$$h_k(S(x)) \approx h^*(S(x)) \approx h^*(x) + \text{distance}(x, S(x))$$

Guarantees for subNN:

Suppose P_X is doubling (i.e., $P_X(B(x,r)) \gtrsim r^d$), and $E[Y|x]$ is Lipschitz

Theorem. For a good choice of $k = k(n)$,

- Parallel computation time is no more than that of (fast) 1-NN
- The Excess Error is at most $\text{OPT}_k(n) + m^{-1/d}$

$$\text{OPT } m = \text{root}(n) \text{ and we can let } m/n \to 0.$$

Intuition: let $N = 1$, and $S(x) \doteq \text{NN}(x)$ in subsample S,

$$h_{\text{sub}}(x) \leftarrow h_k(S(x)) \text{ now}$$

$$h_k(S(x)) \approx h^*(S(x)) \approx h^*(x) + \text{distance}(x, S(x))$$

Outline:

- NN and Data Quantization
- NN and Subsampling
- Overview and Open Questions

So it's possible to get accuracy $\approx$ OPT-NN, in the time of 1-NN

Various open questions:

- Integrating all the data structures
- Taking Y into account in Quantization or Subsampling distribution

So it's possible to get accuracy $\approx$ OPT-NN, in the time of 1-NN

Various open questions:

- Integrating all the data structures
- Taking Y into account in Quantization or Subsampling distribution

So it's possible to get accuracy $\approx$ OPT-NN, in the time of 1-NN

Various open questions:

- Integrating all the data structures

- Taking Y into account in Quantization or Subsampling distribution

So it's possible to get accuracy $\approx$ OPT-NN, in the time of 1-NN

Various open questions:

- Integrating all the data structures
- Taking Y into account in Quantization or Subsampling distribution

A nice open question:

Is there a better subsampling distribution?

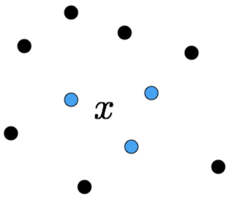
Maybe ...

A nice open question:

Is there a better subsampling distribution?

Maybe ...

Subsampling is weighted k-NN

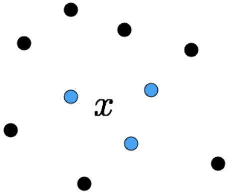


Weighted k-NN: give more weight to closest neighbors

Associate $w_1 \geq w_2 \geq \ldots \geq w_k$ to k-NN$(x) = \{X_{(1)}, \ldots, X_{(k)}\}$

- $h_{k,w}(x) =$ weighted majority (Y_i) of k-NN(x).

Subsampling is weighted k-NN

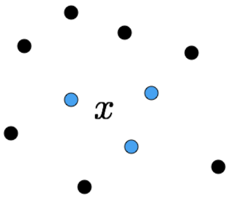


Weighted k-NN: give more weight to closest neighbors

Associate $w_1 \geq w_2 \geq \ldots \geq w_k$ to k-NN$(x) = \{X_{(1)}, \ldots, X_{(k)}\}$

- $h_{k,w}(x) =$ weighted majority (Y_i) of k-NN(x).

Subsampling is weighted k-NN

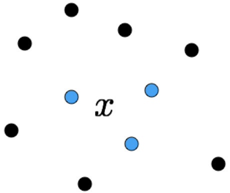


Weighted k-NN: give more weight to closest neighbors

Associate $w_1 \geq w_2 \geq \ldots \geq w_k$ to k-NN$(x) = \{X_{(1)}, \ldots, X_{(k)}\}$

- $h_{k,w}(x) =$ weighted majority (Y_i) of k-NN(x).

Subsampling is weighted k-NN



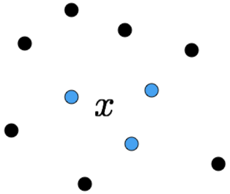
Recall Subsampling: $h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}_{t=1}^{N}$

Intuition: *suppose* $N \to \infty$

Each $X_{(i)} \in k$-NN(x) will appear often as 1-NN(x) in some S_t

Say $X_{(i)}$ appears n_i times, then it contributes $w_i \propto n_i$ to majority.

Subsampling is weighted k-NN



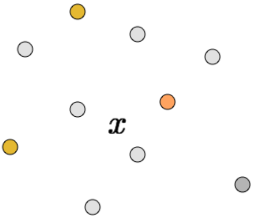
Recall Subsampling: $h_{N,m}(x)$ = majority (Y_t) over $\{S_t\}_{t=1}^N$

Intuition: *suppose* $N \to \infty$

Each $X_{(i)} \in k$-NN(x) will appear often as 1-NN(x) in some S_t

Say $X_{(i)}$ appears n_i times, then it contributes $w_i \propto n_i$ to majority.

Subsampling is weighted k-NN



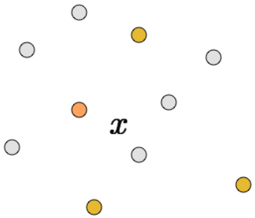
Recall Subsampling: $h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}_{t=1}^{N}$

Intuition: *suppose* $N \to \infty$

Each $X_{(i)} \in k$-NN(x) will appear often as 1-NN(x) in some S_t

Say $X_{(i)}$ appears n_i times, then it contributes $w_i \propto n_i$ to majority.

Subsampling is weighted k-NN



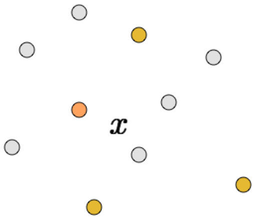
Recall Subsampling: $h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}_{t=1}^{N}$

Intuition: *suppose* $N \to \infty$

Each $X_{(i)} \in k$-NN(x) will appear often as 1-NN(x) in some S_t

Say $X_{(i)}$ appears n_i times, then it contributes $w_i \propto n_i$ to majority.

Subsampling is weighted k-NN



Recall Subsampling: $h_{N,m}(x) =$ majority (Y_t) over $\{S_t\}_{t=1}^{N}$

Intuition: *suppose* $N \to \infty$

Each $X_{(i)} \in k$-NN(x) will appear often as 1-NN(x) in some S_t

Say $X_{(i)}$ appears n_i times, then it contributes $w_i \propto n_i$ to majority.

$$n_i \approx N \cdot \mathbb{P}_{S_t}\left(X_{(i)} \text{ } is \text{ } 1\text{-}NN(x) \text{ } in \text{ } S_t\right) = N \cdot \mathbb{P}_i$$

Sampling S_t $\approx$ pick each point w.p. $p = (m/n)$

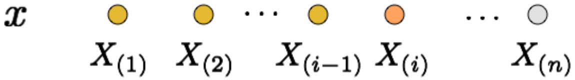
$$\therefore \quad \mathbb{P}_i \approx (1-p)^{i-1} \cdot p$$

So $h_{N,m} \approx h_{k,w}$ with $w_i \propto n_i \propto \mathbb{P}_i$

[Biau et al. 2010] [Samworth 2010]:

err($h_{N,m}$) $\to$ err($h_{k,w}$) typically less than err(h_k)

$$n_i \approx N \cdot \mathbb{P}_{S_t}\left(X_{(i)} \textit{ is } 1\text{-}NN(x) \textit{ in } S_t\right) = N \cdot \mathbb{P}_i$$



Sampling S_t $\approx$ pick each point w.p. $p = (m/n)$

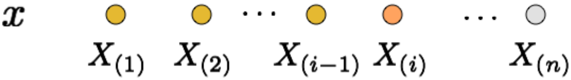
$$\therefore \quad \mathbb{P}_i \approx (1-p)^{i-1} \cdot p$$

So $h_{N,m} \approx h_{k,w}$ with $w_i \propto n_i \propto \mathbb{P}_i$

[Biau et al. 2010] [Samworth 2010]:

err$(h_{N,m}) \to$ err$(h_{k,w})$ typically less than err(h_k)

$$n_i \approx N \cdot \mathbb{P}_{S_t}\left(X_{(i)} \text{ is 1-}NN(x) \text{ in } S_t\right) = N \cdot \mathbb{P}_i$$



Sampling S_t $\approx$ pick each point w.p. $p = (m/n)$

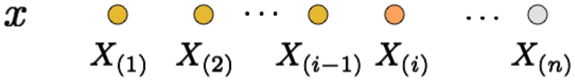
$$\therefore \quad \mathbb{P}_i \approx (1-p)^{i-1} \cdot p$$

So $h_{N,m} \approx h_{k,w}$ with $w_i \propto n_i \propto \mathbb{P}_i$

[Biau et al. 2010] [Samworth 2010]:

err$(h_{N,m}) \to$ err$(h_{k,w})$ typically less than err(h_k)

$$n_i \approx N \cdot \mathbb{P}_{S_t}\left(X_{(i)} \text{ is 1-NN}(x) \text{ in } S_t\right) = N \cdot \mathbb{P}_i$$



Sampling S_t $\approx$ pick each point w.p. $p = (m/n)$

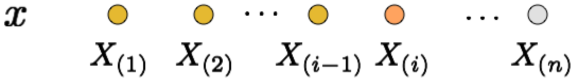
$$\therefore \quad \mathbb{P}_i \approx (1-p)^{i-1} \cdot p$$

So $h_{N,m} \approx h_{k,w}$ with $w_i \propto n_i \propto \mathbb{P}_i$

[Biau et al. 2010] [Samworth 2010]:

err$(h_{N,m}) \to$ err$(h_{k,w})$ typically less than err(h_k)

$$n_i \approx N \cdot \mathbb{P}_{S_t}\left(X_{(i)} \ \textit{is} \ 1\text{-}NN(x) \ \textit{in} \ S_t\right) = N \cdot \mathbb{P}_i$$



Sampling S_t $\approx$ pick each point w.p. $p = (m/n)$

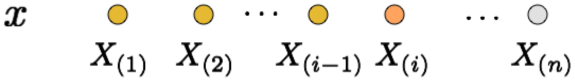
$$\therefore \quad \mathbb{P}_i \approx (1-p)^{i-1} \cdot p$$

So $h_{N,m} \approx h_{k,w}$ with $w_i \propto n_i \propto \mathbb{P}_i$

[Biau et al. 2010] [Samworth 2010]:

err$(h_{N,m}) \to$ err$(h_{k,w})$ typically less than err(h_k)

$$n_i \approx N \cdot \mathbb{P}_{S_t}\left(X_{(i)} \textit{ is } 1\text{-}NN(x) \textit{ in } S_t\right) = N \cdot \mathbb{P}_i$$



Sampling S_t $\approx$ pick each point w.p. $p = (m/n)$

$$\therefore \quad \mathbb{P}_i \approx (1-p)^{i-1} \cdot p$$

So $h_{N,m} \approx h_{k,w}$ with $w_i \propto n_i \propto \mathbb{P}_i$

[Biau et al. 2010] [Samworth 2010]:

err$(h_{N,m}) \to$ err$(h_{k,w})$ typically less than err(h_k)

$h_{k,w}$ is often more accurate than h_k

	1-NN	Majority voting	inverse distance	Dudani	Shepard
PP-attach	80.1	83.4 (13)	83.7 (13)	**84.2** (30)	84.0 (35)
Glass	76.4	**77.3** (2)	–	**77.3** (5)	76.8 (3)
Wine	96.7	**97.8** (3)	**97.8** (3)	**97.8** (7)	**97.8** (3)
Sonar	82.5	–	83.1 (7)	**85.0** (9)	–
Letter	95.6	–	–	**96.0** (5)	–
Isolet	88.6	91.9 (13)	92.4 (13)	**92.9** (15)	92.4 (13)
Vowel	52.6	–	55.6 (7)	**55.8** (15)	55.0 (7)
Segmentation	**90.9**	–	–	–	–
Ionosphere	90.0	–	–	**90.6** (5)	–
Diabetes	66.1	70.1 (3)	69.7 (80)	**70.3** (100)	70.1 (3)
Cancer predicition	69.0	79.5 (11)	**81.0** (9)	79.5 (21)	79.5 (11)
Cancer diagnosis	89.1	**93.3** (5)	93.2 (5)	92.8 (30)	**93.3** (5)
Heart disease	57.0	**62.7** (2)	58.7 (11)	58.7 (9)	59.3 (100)

Experiments on UCI datasets [Zavrel 97]

Dudani scheme: $w_i \propto k - i + 1$ independent of $\text{dist}(x, X_{(i)})$

Theory seems to point to the same ...

$h_{k,w}$ is often more accurate than h_k

	1-NN	Majority voting	inverse distance	Dudani	Shepard
PP-attach	80.1	83.4 (13)	83.7 (13)	**84.2** (30)	84.0 (35)
Glass	76.4	**77.3** (2)	–	**77.3** (5)	76.8 (3)
Wine	96.7	**97.8** (3)	**97.8** (3)	**97.8** (7)	**97.8** (3)
Sonar	82.5	–	83.1 (7)	**85.0** (9)	–
Letter	95.6	–	–	**96.0** (5)	–
Isolet	88.6	91.9 (13)	92.4 (13)	**92.9** (15)	92.4 (13)
Vowel	52.6	–	55.6 (7)	**55.8** (15)	55.0 (7)
Segmentation	**90.9**	–	–	–	–
Ionosphere	90.0	–	–	**90.6** (5)	–
Diabetes	66.1	70.1 (3)	69.7 (80)	**70.3** (100)	70.1 (3)
Cancer predicition	69.0	79.5 (11)	**81.0** (9)	79.5 (21)	79.5 (11)
Cancer diagnosis	89.1	**93.3** (5)	93.2 (5)	92.8 (30)	**93.3** (5)
Heart disease	57.0	**62.7** (2)	58.7 (11)	58.7 (9)	59.3 (100)

Experiments on UCI datasets [Zavrel 97]

Dudani scheme: $w_i \propto k - i + 1$ independent of $\text{dist}(x, X_{(i)})$

Theory seems to point to the same ...

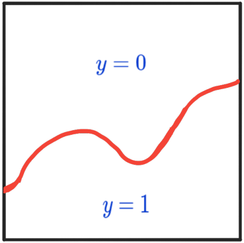
$h_{k,w}$ is often more accurate than h_k

	1-NN	Majority voting	inverse distance	Dudani	Shepard
PP-attach	80.1	83.4 (13)	83.7 (13)	**84.2** (30)	84.0 (35)
Glass	76.4	**77.3** (2)	–	**77.3** (5)	76.8 (3)
Wine	96.7	**97.8** (3)	**97.8** (3)	**97.8** (7)	**97.8** (3)
Sonar	82.5	–	83.1 (7)	**85.0** (9)	–
Letter	95.6	–	–	**96.0** (5)	–
Isolet	88.6	91.9 (13)	92.4 (13)	**92.9** (15)	92.4 (13)
Vowel	52.6	–	55.6 (7)	**55.8** (15)	55.0 (7)
Segmentation	**90.9**	–	–	–	–
Ionosphere	90.0	–	–	**90.6** (5)	–
Diabetes	66.1	70.1 (3)	69.7 (80)	**70.3** (100)	70.1 (3)
Cancer predicition	69.0	79.5 (11)	**81.0** (9)	79.5 (21)	79.5 (11)
Cancer diagnosis	89.1	**93.3** (5)	93.2 (5)	92.8 (30)	**93.3** (5)
Heart disease	57.0	**62.7** (2)	58.7 (11)	58.7 (9)	59.3 (100)

Experiments on UCI datasets [Zavrel 97]

Dudani scheme: $w_i \propto k - i + 1$ independent of dist$(x, X_{(i)})$

Theory seems to point to the same ...

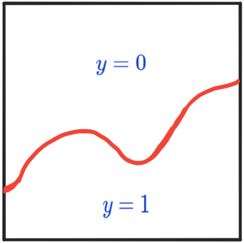


Theory: [Samworth 2010] under minor technical conditions ...

$\exists \{w_i^*\}$, independent of distance, s.t.

$$\frac{\mathbb{E}\,\text{err}(h_{k,w^*})}{\mathbb{E}\,\text{err}(h_k)} \xrightarrow{n \to \infty} C < 1.$$

w^*, C depend on changes in P_X and $\mathbb{E}[Y|X]$ near class-boundary.



Theory: [Samworth 2010] under minor technical conditions ...

$\exists\{w_i^*\}$, independent of distance, s.t.

$$\frac{\mathbb{E}\,\mathsf{err}(h_{k,w^*})}{\mathbb{E}\,\mathsf{err}(h_k)} \xrightarrow{n\to\infty} C < 1.$$

w^*, C depend on changes in P_X and $\mathbb{E}[Y|X]$ near class-boundary.

Open Questions:

Best $h_{k,w^*} \equiv$ Best subsampling distribution?

How do we even infer best w^* from data?

Open Questions:

Best h_{k,w^*} $\equiv$ Best subsampling distribution?

How do we even infer best w^* from data?

Open Questions:

Best h_{k,w^*} $\equiv$ Best subsampling distribution?

How do we even infer best w^* from data?

. . .