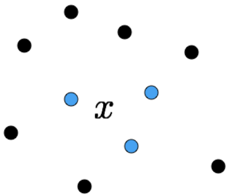
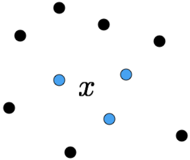
# Fast NN prediction with no Statistical tradeoff




**Samory Kpotufe**

ORFE, Princeton University   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.

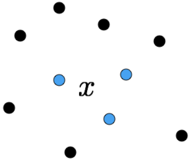
**Unfortunately, optimal accuracy requires large $k = \Omega($root of$(n))$ ...**

# 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.

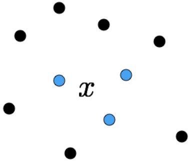
**Unfortunately, optimal accuracy requires large** $k = \Omega($**root of**$(n))$ **…**

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

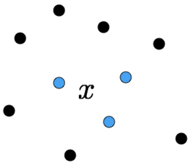
**Prediction Time:** at least order $k$,

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

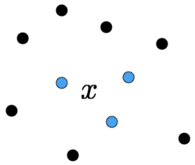
**Prediction Time:** at least order $k$,

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.

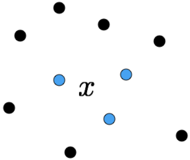
**Unfortunately, optimal accuracy requires large** $k = \Omega($**root of**$(n))$ ...

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

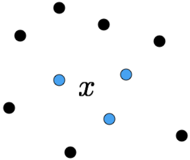
**Prediction Time:** at least order $k$,

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:

$$\mathbb{E}\left(f_k(X) - f(X)\right)^2 \approx \frac{1}{k} + \left(\frac{k}{n}\right)^{2/d}$$ 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:** $Y = f(X) +$ 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:

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

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:** $Y = f(X) +$ 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:

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

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:*

$\epsilon$-**NN:** use all samples $\epsilon$-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:*

$\epsilon$-**NN:** use all samples $\epsilon$-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:*

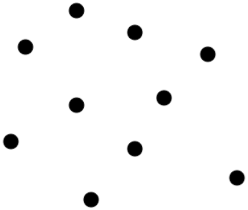
$\epsilon$-**NN:** use all samples $\epsilon$-close to $x$

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

**Outline:**

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

# **Quantization:** *reduce the data*



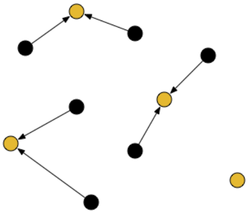
$\{X_i\}_{i=1}^{n}$

**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*



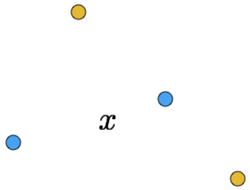
Assign $\{X_i\}$ to representatives $\mathbf{Q} \equiv \{q\}$

**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*



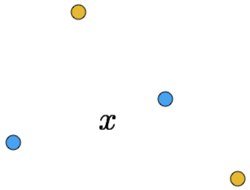

Pick $q$'s in $\mathbf{Q}$ close to $x$

**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*



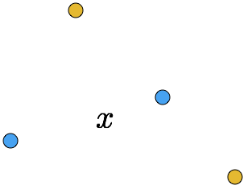

Pick $q$'s in $\mathbf{Q}$ close to $x$

**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 $B(x, \epsilon)$.

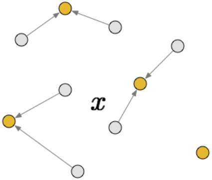
**Main issues:**

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

# $\epsilon$-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 $\{X_i \to q\}$



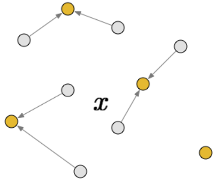

**We'll make a few changes for the guarantees we want ...**

## $\epsilon$-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 $\{X_i \to q\}$



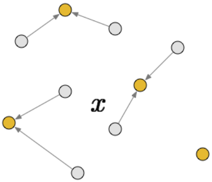

We'll make a few changes for the guarantees we want ...

# $\epsilon$-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 $\{X_i \to q\}$



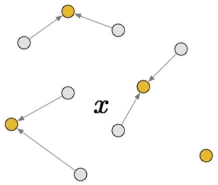

We'll make a few changes for the guarantees we want ...

# $\epsilon$-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 ...

# $\epsilon$-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

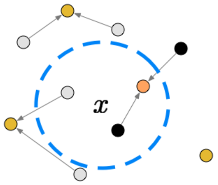
Pick $\mathbf{Q}$ to **(1)** have small size, and **(2)** be close to $\{X_i\}$ ...

**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 ..

# $\epsilon$-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

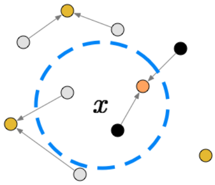
Pick $\mathbf{Q}$ to **(1)** have small size, and **(2)** be close to $\{X_i\}$ ...

**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 ..**

# $\epsilon$-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

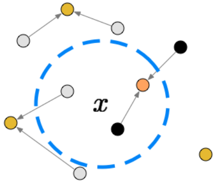
Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

**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 ..**

# $\epsilon$-NN Heuristics: [Atkeson et al 97] [Carrier et al. 88] [Lee, Gray 08]

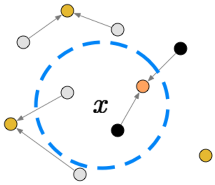
Pick $\mathbf{Q}$ as **(1)** $(\alpha \cdot \epsilon)$-packing, and **(2)** an $(\alpha \cdot \epsilon)$-cover of $\{X_i\}$.

**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 $d$

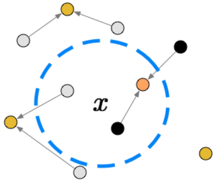
**Relate $f_{\mathbf{Q}}$ to $\epsilon$-NN $f_\epsilon$ (on $n$ samples) ...**

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

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

**Relate $f_{\mathbf{Q}}$ to $\epsilon$-NN $f_\epsilon$ (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 $O(1/\min n_q)$

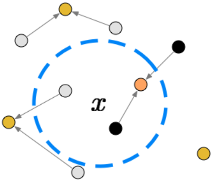
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon)|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

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

**Relate $f_{\mathbf{Q}}$ to $\epsilon$-NN $f_\epsilon$ (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 $O(1/\min n_q)$

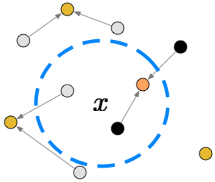
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon)|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

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

**Relate $f_{\mathbf{Q}}$ to $\epsilon$-NN $f_\epsilon$ (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 $2\epsilon$-close to $x$ ($\approx$ bias of $f_\epsilon$)




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

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

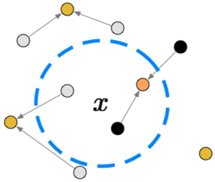
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

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

**Relate $f_{\mathbf{Q}}$ to $\epsilon$-NN $f_\epsilon$ (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 $2\epsilon$-close to $x$ ($\approx$ bias of $f_\epsilon$)




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

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

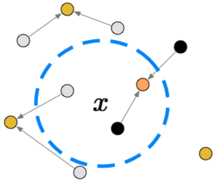
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

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

**Relate $f_{\mathbf{Q}}$ to $\epsilon$-NN $f_\epsilon$ (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 $2\epsilon$-close to $x$ ($\approx$ bias of $f_\epsilon$)




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

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

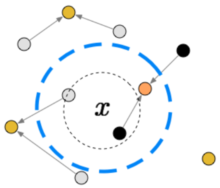
Argue that $\sum n_q > |\{X_i\} \cap B(x, (1-\alpha)\epsilon))|$ ($\approx$ Var of $f_{(1-\alpha)\epsilon}$)

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

**Relate $f_{\mathbf{Q}}$ to $\epsilon$-NN $f_\epsilon$ (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 $2\epsilon$-close to $x$ ($\approx$ bias of $f_\epsilon$)




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

- 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 $\epsilon$:

- $f_Q$ (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 $\epsilon$:

- $f_Q$ (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 $\epsilon$:

- $f_Q$ (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 $\epsilon$:

- $f_Q$ (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 $\epsilon$-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 $\epsilon$-NN ...

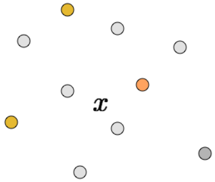
**Outline:**

- 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]:

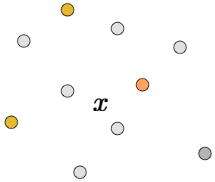
- Large $N \implies$ reduce variance.
- **Tradeoff on** $m$**:** small $m \implies$ richer $\{S_t\}$, but more variance.

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]:

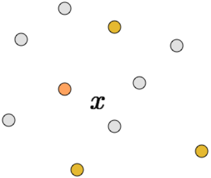
- Large $N \implies$ reduce variance.
- **Tradeoff on** $m$**:** small $m \implies$ richer $\{S_t\}$, but more variance.

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]*:*

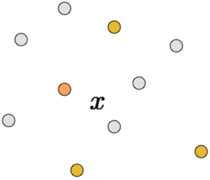
- Large $N \implies$ reduce variance.
- **Tradeoff on** $m$**:** small $m \implies$ richer $\{S_t\}$, but more variance.

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]:

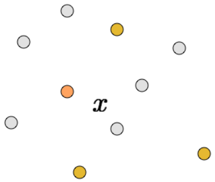
- Large $N \implies$ reduce variance.
- **Tradeoff on** $m$**:** small $m \implies$ richer $\{S_t\}$, but more variance.

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]:

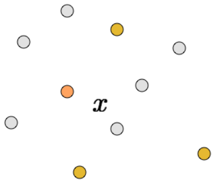
- Large $N \implies$ reduce variance.
- **Tradeoff on** $m$**:** small $m \implies$ richer $\{S_t\}$, but more variance.

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.

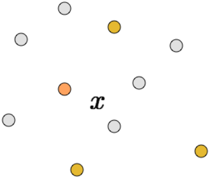
- **Tradeoff on** $m$: small $m \implies$ richer $\{S_t\}$, but more variance.

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 $m/n \xrightarrow{n\to\infty} 0$.

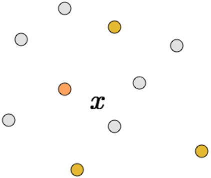
## **Subsampling:** *reduce data and parallelize*

**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]:

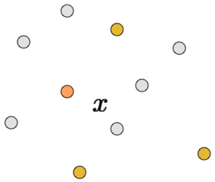
- Large $N \implies$ reduce variance.
- **Tradeoff on** $m$**:** small $m \implies$ richer $\{S_t\}$, but more variance.

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 $m \ll n$



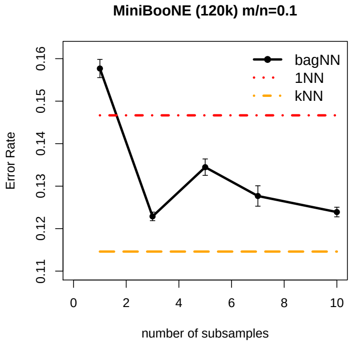

$Y_t(x) \leftarrow Y$-value of 1-NN$(x)$ in $S_t$

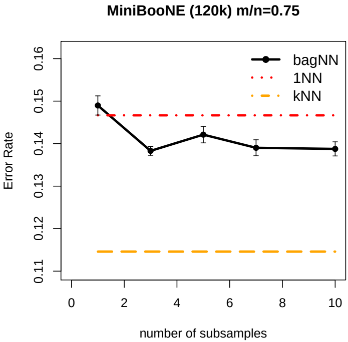
$h_{N,m}(x) =$ majority $(Y_t)$ over $\{S_t\}$

**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.

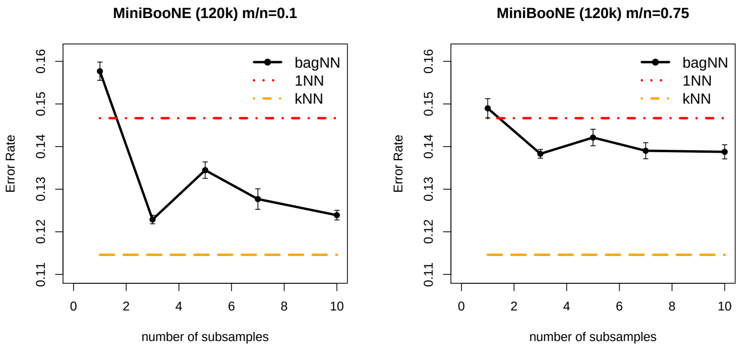
Optimal choice: $m = \Omega(n^{d/(2+d)}) \implies$ ratio $m/n \xrightarrow{n \to \infty} 0$.




**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?

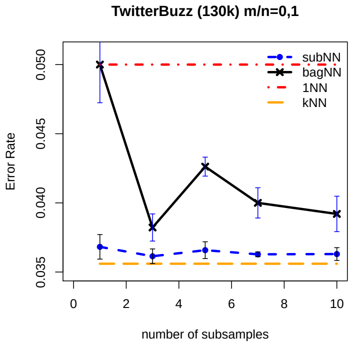
[Biau et al. 2010] [Samworth 2010]: Yes, as $N \to \infty$

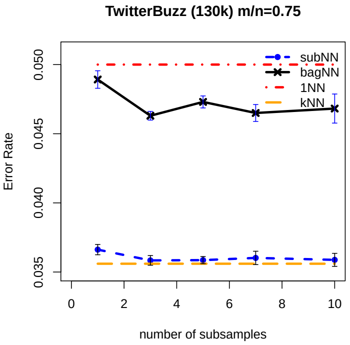
**We want high accuracy for small $N$:**

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 $\mathrm{OPT}_k(n) + m^{-1/d}$

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 $S$,

$h_{\text{sub}}(x) \leftarrow h_k(S(x))$ 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 $\mathrm{OPT}_k(n) + m^{-1/d}$

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 $S$,

$h_{\text{sub}}(x) \leftarrow h_k(S(x))$ 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 $\mathrm{OPT}_k(n) + m^{-1/d}$

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 $S$,

$$h_{\mathsf{sub}}(x) \leftarrow h_k(S(x))$$ 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 $\mathrm{OPT}_k(n) + m^{-1/d}$

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 $S$,

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

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

**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
- Maintaining accuracy of related methods (e.g. SVMs)

...

# Thanks

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
- Maintaining accuracy of related methods (e.g. SVMs)

...

**Thanks**

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

- Maintaining accuracy of related methods (e.g. SVMs)

...

**Thanks**

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
- Maintaining accuracy of related methods (e.g. SVMs)

...

**Thanks**

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
- Maintaining accuracy of related methods (e.g. SVMs)

...

**Thanks**

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
- Maintaining accuracy of related methods (e.g. SVMs)

...

# Thanks