Optimal Probability Estimation and Classification

with

Jayadev Acharya, Ashkan Jafarpour, Ananda Theertha Suresh

UC San Diego

Probability Estimation

Probability Estimation

- Domains

Probability Estimation

- Domains
  - Large alphabets

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction
  - Closeness testing

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction
  - Closeness testing
  - Identity testing

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction
  - Closeness testing
  - Identity testing
  - Classification

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction
  - Closeness testing
  - Identity testing
  - Classification
- Methodologies

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction
  - Closeness testing
  - Identity testing
  - Classification
- Methodologies
  - Define doable problem

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction
  - Closeness testing
  - Identity testing
  - Classification
- Methodologies
  - Define doable problem
  - Approach limits

Probability Estimation

- Domains
  - Large alphabets
  - Mixture models
  - Continuous distributions
- Applications
  - Compression
  - Prediction
  - Closeness testing
  - Identity testing
  - Classification
- Methodologies
  - Define doable problem
  - Approach limits
  - Approach the best possible

Overview

Overview

- Probability estimation

Overview

- Probability estimation
  - Motivation

Overview

- Probability estimation
  - Motivation
  - Combined probability

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator
  - Proof sketch

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator
  - Proof sketch
- Classification

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator
  - Proof sketch
- Classification
  - Motivation

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator
  - Proof sketch
- Classification
  - Motivation
  - Label-invariant classifier

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator
  - Proof sketch
- Classification
  - Motivation
  - Label-invariant classifier
  - Nearly-optimal estimators

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator
  - Proof sketch
- Classification
  - Motivation
  - Label-invariant classifier
  - Nearly-optimal estimators
- Prediction

Overview

- Probability estimation
  - Motivation
  - Combined probability
  - Previous results
  - Optimal estimator
  - Proof sketch
- Classification
  - Motivation
  - Label-invariant classifier
  - Nearly-optimal estimators
- Prediction
- Conclusion

Motivation

Original Scene

Original Scene

- Coin: (p_h, p_t) $\qquad p_h + p_t = 1$

Original Scene

- Coin: $(p_h, p_t) \qquad p_h + p_t = 1$
- Flip n times, estimate p_h

Original Scene

- Coin: $(p_h, p_t) \qquad p_h + p_t = 1$
- Flip n times, estimate p_h
- Empirical frequency estimate: h appear n_h times, $E_h \stackrel{\text{def}}{=} \frac{n_h}{n}$

Original Scene

- Coin: $(p_h, p_t) \qquad p_h + p_t = 1$
- Flip n times, estimate p_h
- Empirical frequency estimate: h appear n_h times, $E_h \stackrel{\text{def}}{=} \frac{n_h}{n}$
- Law of large numbers: $E_h \xrightarrow[n \to \infty]{} p_h$

Original Scene

- Coin: $(p_h, p_t) \qquad p_h + p_t = 1$
- Flip n times, estimate p_h
- Empirical frequency estimate: h appear n_h times, $E_h \stackrel{\text{def}}{=} \frac{n_h}{n}$
- Law of large numbers: $E_h \xrightarrow[n \to \infty]{} p_h$
- Expectation: $\mathbb{E}(E_h) = p_h$

Original Scene

- Coin: $(p_h, p_t) \qquad p_h + p_t = 1$
- Flip n times, estimate p_h
- Empirical frequency estimate: h appear n_h times, $E_h \stackrel{\text{def}}{=} \frac{n_h}{n}$
- Law of large numbers: $E_h \xrightarrow[n\to\infty]{} p_h$
- Expectation: $\mathbb{E}(E_h) = p_h$
- Standard deviation: $\sqrt{\frac{pq}{n}} \leq \frac{1}{2\sqrt{n}}$

Original Scene

- Coin: $(p_h, p_t) \qquad p_h + p_t = 1$
- Flip n times, estimate p_h
- Empirical frequency estimate: h appear n_h times, $E_h \stackrel{\text{def}}{=} \frac{n_h}{n}$
- Law of large numbers: $E_h \xrightarrow[n\to\infty]{} p_h$
- Expectation: $\mathbb{E}(E_h) = p_h$
- Standard deviation: $\sqrt{\frac{pq}{n}} \leq \frac{1}{2\sqrt{n}}$
- For $|E_h - p_h| < \delta$ need $n = \Theta\left(\frac{1}{\delta^2}\right)$

Original Scene

- Coin: (p_h, p_t) $\qquad p_h + p_t = 1$
- Flip n times, estimate p_h
- Empirical frequency estimate: h appear n_h times, $E_h \stackrel{\text{def}}{=} \frac{n_h}{n}$
- Law of large numbers: $E_h \xrightarrow[n\to\infty]{} p_h$
- Expectation: $\mathbb{E}(E_h) = p_h$
- Standard deviation: $\sqrt{\frac{pq}{n}} \le \frac{1}{2\sqrt{n}}$
- For $|E_h - p_h| < \delta$ need $n = \Theta\left(\frac{1}{\delta^2}\right)$
- For any given difference, need constant samples

Large Alphabets

Large Alphabets

- Text processing ($\approx 500,000$ words)

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]
  - Topic modeling [Blei Ng Jordan '03]

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]
  - Topic modeling [Blei Ng Jordan '03]
- Biology

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]
  - Topic modeling [Blei Ng Jordan '03]
- Biology
  - Species estimation

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]
  - Topic modeling [Blei Ng Jordan '03]
- Biology
  - Species estimation
  - Genetics

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]
  - Topic modeling [Blei Ng Jordan '03]
- Biology
  - Species estimation
  - Genetics
- Online marketing

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]
  - Topic modeling [Blei Ng Jordan '03]
- Biology
  - Species estimation
  - Genetics
- Online marketing
  - Ad click-through

Large Alphabets

- Text processing ($\approx 500,000$ words)
  - Speech recognition
  - Machine translation
- Natural language processing (bag of words)
  - Text classification [McCallum Nigam '98]
  - Topic modeling [Blei Ng Jordan '03]
- Biology
  - Species estimation
  - Genetics
- Online marketing
  - Ad click-through
  - Movies

Sample Complexity

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$
- For some distributions, $n = \Omega(k)$ [Paninski '04]

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$
- For some distributions, $n = \Omega(k)$ [Paninski '04]
  - Take arbitrary $k/2$-element subset of $\{1, , \ldots ,, k\}$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$
- For some distributions, $n = \Omega(k)$ [Paninski '04]
  - Take arbitrary $k/2$-element subset of $\{1, , \ldots ,, k\}$
  - If $n < k/4$, observe $\leq k/4$ values, uniform over remaining $3k/4$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$
- For some distributions, $n = \Omega(k)$ [Paninski '04]
  - Take arbitrary $k/2$-element subset of $\{1, , \ldots ,, k\}$
  - If $n < k/4$, observe $\leq k/4$ values, uniform over remaining $3k/4$
  - $||E - p||_1 \geq \frac{1}{3}$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$
- For some distributions, $n = \Omega(k)$ [Paninski '04]
  - Take arbitrary $k/2$-element subset of $\{1, , \ldots ,, k\}$
  - If $n < k/4$, observe $\leq k/4$ values, uniform over remaining $3k/4$
  - $||E - p||_1 \geq \frac{1}{3}$
- $n = \Theta(k/\delta^2)$

Sample Complexity

- $P = (p_1, p_2, \dots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$
- For some distributions, $n = \Omega(k)$ [Paninski '04]
  - Take arbitrary $k/2$-element subset of $\{1, , \dots ,, k\}$
  - If $n < k/4$, observe $\leq k/4$ values, uniform over remaining $3k/4$
  - $||E - p||_1 \geq \frac{1}{3}$
- $n = \Theta(k/\delta^2)$
- $k = 500,000$, $\delta = 0.01 \quad \rightarrow \quad n = 50\text{B}$

Sample Complexity

- $P = (p_1, p_2, \ldots p_k)$
- n samples, $X^n \sim p$
- E: estimator
- ℓ_1 distance: $||E - p||_1 \stackrel{\text{def}}{=} \sum_{i=1}^{k} |E(i) - p_i|$
- $\ell_1 \leq .01$ with probability $\geq .99$
- Empirical: $n = \mathcal{O}(k)$
- For some distributions, $n = \Omega(k)$ [Paninski '04]
  - Take arbitrary $k/2$-element subset of $\{1, , \ldots ,, k\}$
  - If $n < k/4$, observe $\leq k/4$ values, uniform over remaining $3k/4$
  - $||E - p||_1 \geq \frac{1}{3}$
- $n = \Theta(k/\delta^2)$
- $k = 500,000$, $\delta = 0.01 \quad \rightarrow \quad n = 50\text{B}$
- KL divergence: similar, $n = \Theta(k)$

Previous Approaches

- Properties of p

Previous Approaches

- Properties of p
  - $\forall i \; p_i > 1/k$, estimate support size, entropy

Previous Approaches

- Properties of p
  - $\forall i \; p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]

Previous Approaches

- Properties of p
  - $\forall i \; p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement

Previous Approaches

- Properties of p
  - $\forall i \ p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement
- Assumptions on p

Previous Approaches

- Properties of p
  - $\forall i\ p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement
- Assumptions on p
  - p is monotone (or m-modal) over $[k]$

Previous Approaches

- Properties of p
  - $\forall i \ p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement
- Assumptions on p
  - p is monotone (or m-modal) over $[k]$
  - $n = \mathrm{polylog}(k)$ [Daskalakis, Diaconikolas, Servedio '12]

Previous Approaches

- Properties of p
  - $\forall i \; p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement
- Assumptions on p
  - p is monotone (or m-modal) over $[k]$
  - $n = \mathrm{polylog}(k)$ [Daskalakis, Diaconikolas, Servedio '12]
- Our approach

Previous Approaches

- Properties of p
  - $\forall i \; p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement
- Assumptions on p
  - p is monotone (or m-modal) over $[k]$
  - $n = \mathrm{polylog}(k)$ [Daskalakis, Diaconikolas, Servedio '12]
- Our approach
  - General distributions

Previous Approaches

- Properties of p
  - $\forall i \ p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement
- Assumptions on p
  - p is monotone (or m-modal) over $[k]$
  - $n = \mathrm{polylog}(k)$ [Daskalakis, Diaconikolas, Servedio '12]
- Our approach
  - General distributions
  - Best anyone can do

Previous Approaches

- Properties of p
  - $\forall i \; p_i > 1/k$, estimate support size, entropy
  - $n = \Theta\left(\frac{k}{\log k}\right)$ [Valiant Valiant '11]
  - Factor of $\log k$ improvement
- Assumptions on p
  - p is monotone (or m-modal) over $[k]$
  - $n = \text{polylog}(k)$ [Daskalakis, Diaconikolas, Servedio '12]
- Our approach
  - General distributions
  - Best anyone can do
  - Natural estimators

Combined-Probability Estimation

Natural Estimators

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$
- $x^5 = a\, b\, b\, a\, c$

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$
- $x^5 = a\, b\, b\, a\, c$
- p_a, p_b?

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$
- $x^5 = a\, b\, b\, a\, c$
- p_a, p_b?
  - Both appeared twice

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$
- $x^5 = a\, b\, b\, a\, c$
- p_a, p_b?
    - Both appeared twice
- Without prior knowledge, for every natural estimator $\hat{p}$

$$\hat{p}_a = \hat{p}_b$$

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$
- $x^5 = a\,b\,b\,a\,c$
- p_a, p_b?
  - Both appeared twice
- Without prior knowledge, for every natural estimator $\hat{p}$

$$\hat{p}_a = \hat{p}_b$$

- If symbols have appeared same # of times

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$
- $x^5 = a\, b\, b\, a\, c$
- p_a, p_b?
  - Both appeared twice
- Without prior knowledge, for every natural estimator $\hat{p}$

$$\hat{p}_a = \hat{p}_b$$

- If symbols have appeared same # of times
  - Assign same probability

Natural Estimators

- Distribution over $\{a, b, c, d, e, f\}$
- $x^5 = a\, b\, b\, a\, c$
- p_a, p_b?
  - Both appeared twice
- Without prior knowledge, for every natural estimator $\hat{p}$

$$\hat{p}_a = \hat{p}_b$$

- If symbols have appeared same # of times
  - Assign same probability
  - Similarly for unseen symbols

Definitions

Definitions

- μ: multiplicity, number of occurrences of a symbol

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\mathrm{def}}{=} \#$ symbols with multiplicity μ

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\ d\ c\ d$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\ d\ c\ d$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\ d\ c\ d$

$$N_0 = 1 \ (\text{b})$$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\, d\, c\, d$

$$N_0 = 1 \text{ (b)} \quad N_1 = 2 \text{ (a,c)}$$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\, d\, c\, d$

$$N_0 = 1 \text{ (b)} \quad N_1 = 2 \text{ (a,c)} \quad N_2 = 1 \text{ (d)}$$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\ d\ c\ d$

$$N_0 = 1\ (\mathrm{b}) \quad N_1 = 2\ (\mathrm{a,c}) \quad N_2 = 1\ (\mathrm{d})$$
$$S_0 = p_b$$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\ d\ c\ d$

$$N_0 = 1\ (\text{b}) \quad N_1 = 2\ (\text{a,c}) \quad N_2 = 1\ (\text{d})$$
$$S_0 = p_b \quad S_1 = p_a + p_c$$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\ d\ c\ d$

$$N_0 = 1\ (\text{b}) \quad N_1 = 2\ (\text{a,c}) \quad N_2 = 1\ (\text{d})$$
$$S_0 = p_b \quad S_1 = p_a + p_c \quad S_2 = p_d$$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\ d\ c\ d$

$$\begin{array}{lll} N_0 = 1 \text{ (b)} & N_1 = 2 \text{ (a,c)} & N_2 = 1 \text{ (d)} \\ S_0 = p_b & S_1 = p_a + p_c & S_2 = p_d \end{array}$$

- If symbol x appeared $\mu \geq 1$ times, $q_x = \frac{S_\mu}{N_\mu}$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\,d\,c\,d$

$$\begin{array}{lll} N_0 = 1\ (\mathrm{b}) & N_1 = 2\ (\mathrm{a,c}) & N_2 = 1\ (\mathrm{d}) \\ S_0 = p_b & S_1 = p_a + p_c & S_2 = p_d \end{array}$$

- If symbol x appeared $\mu \geq 1$ times, $q_x = \frac{S_\mu}{N_\mu}$
  - $q_a = q_c = \frac{S_1}{N_1} = \frac{S_1}{2}$

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\, d\, c\, d$

$$N_0 = 1 \text{ (b)} \quad N_1 = 2 \text{ (a,c)} \quad N_2 = 1 \text{ (d)}$$
$$S_0 = p_b \quad S_1 = p_a + p_c \quad S_2 = p_d$$

- If symbol x appeared $\mu \geq 1$ times, $q_x = \frac{S_\mu}{N_\mu}$
  - $q_a = q_c = \frac{S_1}{N_1} = \frac{S_1}{2}$
- Unseen probability: S_0

Definitions

- μ: multiplicity, number of occurrences of a symbol
- $N_\mu \stackrel{\text{def}}{=} \#$ symbols with multiplicity μ
- $S_\mu \stackrel{\text{def}}{=}$ sum of probabilities of symbols with multiplicity μ
- Example: distribution over $\{a, b, c, d\}$
  - $x^4 = a\, d\, c\, d$

$$\begin{array}{lll} N_0 = 1 \text{ (b)} & N_1 = 2 \text{ (a,c)} & N_2 = 1 \text{ (d)} \\ S_0 = p_b & S_1 = p_a + p_c & S_2 = p_d \end{array}$$

- If symbol x appeared $\mu \geq 1$ times, $q_x = \frac{S_\mu}{N_\mu}$
  - $q_a = q_c = \frac{S_1}{N_1} = \frac{S_1}{2}$
- Unseen probability: S_0
- Combined-probability estimation: estimate $S_0, S_1, \ldots, S_n$

Distance Measures

Distance Measures

- $\widehat{S} = (\widehat{S}_0, \widehat{S}_1, \ldots, \widehat{S}_n)$ estimate of $S = (S_0, S_1, \ldots, S_n)$

Distance Measures

- $\widehat{S} = (\widehat{S}_0, \widehat{S}_1, \ldots, \widehat{S}_n)$ estimate of $S = (S_0, S_1, \ldots, S_n)$
- Optimality criteria?

Distance Measures

- $\widehat{S} = (\widehat{S}_0, \widehat{S}_1, \dots, \widehat{S}_n)$ estimate of $S = (S_0, S_1, \dots, S_n)$
- Optimality criteria?
  - ℓ_1 distance: consistency, classification

$$||S - \widehat{S}||_1 \stackrel{\text{def}}{=} \sum_{\mu=0}^{n} |S_\mu - \widehat{S}_\mu|$$

Distance Measures

- $\widehat{S} = (\widehat{S}_0, \widehat{S}_1, \ldots, \widehat{S}_n)$ estimate of $S = (S_0, S_1, \ldots, S_n)$
- Optimality criteria?
  - ℓ_1 distance: consistency, classification

$$||S - \widehat{S}||_1 \stackrel{\text{def}}{=} \sum_{\mu=0}^{n} |S_\mu - \widehat{S}_\mu|$$

  - KL divergence: universal compression, prediction with log-loss

$$D(S||\widehat{S}) \stackrel{\text{def}}{=} \sum_{\mu=0}^{n} S_\mu \log \frac{S_\mu}{\widehat{S}_\mu}$$

Empirical Frequency

Empirical Frequency

- $x^9 = a\,b\,e\,b\,c\,a\,d\,c\,c$

Empirical Frequency

- $x^9 = a\, b\, e\, b\, c\, a\, d\, c\, c$
- $S_2 = p_a + p_b$

Empirical Frequency

- $x^9 = a\ b\ e\ b\ c\ a\ d\ c\ c$
- $S_2 = p_a + p_b$
- Empirical frequency: $E_2 = 2/9 + 2/9 = 4/9$

Empirical Frequency

- $x^9 = a\, b\, e\, b\, c\, a\, d\, c\, c$
- $S_2 = p_a + p_b$
- Empirical frequency: $E_2 = 2/9 + 2/9 = 4/9$
  - Recall: $N_\mu = \#$ of symbols appearing μ times

$$E_\mu = N_\mu \frac{\mu}{n}$$

Empirical Frequency

- $x^9 = a\ b\ e\ b\ c\ a\ d\ c\ c$
- $S_2 = p_a + p_b$
- Empirical frequency: $E_2 = 2/9 + 2/9 = 4/9$
  - Recall: $N_\mu = \#$ of symbols appearing μ times

$$E_\mu = N_\mu \frac{\mu}{n}$$

- $\#$ of samples for $\ell_1 \leq 0.01$ with probability ≥ 0.99

Empirical Frequency

- $x^9 = a\, b\, e\, b\, c\, a\, d\, c\, c$
- $S_2 = p_a + p_b$
- Empirical frequency: $E_2 = 2/9 + 2/9 = 4/9$
  - Recall: $N_\mu = \#$ of symbols appearing μ times

$$E_\mu = N_\mu \frac{\mu}{n}$$

- $\#$ of samples for $\ell_1 \leq 0.01$ with probability ≥ 0.99
  - $E_0 = 0$ always

Empirical Frequency

- $x^9 = a\ b\ e\ b\ c\ a\ d\ c\ c$
- $S_2 = p_a + p_b$
- Empirical frequency: $E_2 = 2/9 + 2/9 = 4/9$
  - Recall: $N_\mu = \#$ of symbols appearing μ times

$$E_\mu = N_\mu \frac{\mu}{n}$$

- $\#$ of samples for $\ell_1 \le 0.01$ with probability ≥ 0.99
  - $E_0 = 0$ always
  - $U\{1, \ldots, k\}$, $n = 0.98k$

$$S_0 > 0.02$$

Empirical Frequency

- $x^9 = a\, b\, e\, b\, c\, a\, d\, c\, c$
- $S_2 = p_a + p_b$
- Empirical frequency: $E_2 = 2/9 + 2/9 = 4/9$
  - Recall: $N_\mu = \#$ of symbols appearing μ times

$$E_\mu = N_\mu \frac{\mu}{n}$$

- $\#$ of samples for $\ell_1 \leq 0.01$ with probability ≥ 0.99
  - $E_0 = 0$ always
  - $U\{1, \ldots, k\}$, $n = 0.98k$

$$S_0 > 0.02$$

  - $n > 0.98k$

Good Turing

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times
- For $\mu = 0, 1, \ldots$

$$G_\mu = N_{\mu+1} \frac{\mu + 1}{n}$$

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times
- For $\mu = 0, 1, \ldots$

$$G_\mu = N_{\mu+1} \frac{\mu + 1}{n}$$

- Unbiased

$$\mathbb{E}[G_\mu] = \mathbb{E}[S_\mu]$$

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times
- For $\mu = 0, 1, \ldots$

$$G_\mu = N_{\mu+1} \frac{\mu + 1}{n}$$

- Unbiased

$$\mathbb{E}[G_\mu] = \mathbb{E}[S_\mu]$$

- Probability of unseen mass

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times
- For $\mu = 0, 1, \ldots$
$$G_\mu = N_{\mu+1} \frac{\mu + 1}{n}$$
- Unbiased
$$\mathbb{E}[G_\mu] = \mathbb{E}[S_\mu]$$
- Probability of unseen mass
  - $E_0 = 0$

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times
- For $\mu = 0, 1, \ldots$
$$G_\mu = N_{\mu+1} \frac{\mu + 1}{n}$$
- Unbiased
$$\mathbb{E}[G_\mu] = \mathbb{E}[S_\mu]$$
- Probability of unseen mass
  - $E_0 = 0$
  - $G_0 = \frac{N_1}{n}$

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times
- For $\mu = 0, 1, \ldots$
$$G_\mu = N_{\mu+1} \frac{\mu + 1}{n}$$
- Unbiased
$$\mathbb{E}[G_\mu] = \mathbb{E}[S_\mu]$$
- Probability of unseen mass
  - $E_0 = 0$
  - $G_0 = \frac{N_1}{n}$
- Basic tool in NLP [Church Gale '81]

Good Turing

- $N_{\mu+1}$: # symbols appearing $\mu + 1$ times
- For $\mu = 0, 1, \ldots$
$$G_\mu = N_{\mu+1} \frac{\mu + 1}{n}$$
- Unbiased
$$\mathbb{E}[G_\mu] = \mathbb{E}[S_\mu]$$
- Probability of unseen mass
  - $E_0 = 0$
  - $G_0 = \frac{N_1}{n}$
- Basic tool in NLP [Church Gale '81]
- Performance guarantee?

Previous results

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors
- [McAllester Schapire '00]

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors
- [McAllester Schapire '00]
  - w.h.p., $\forall \mu$

$$|S_\mu - G_\mu| = \widetilde{\mathcal{O}}\left(\frac{\mu + 1}{\sqrt{n}}\right)$$

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors
- [McAllester Schapire '00]
  - w.h.p., $\forall \mu$

$$|S_\mu - G_\mu| = \widetilde{\mathcal{O}}\left(\frac{\mu + 1}{\sqrt{n}}\right)$$

- Holds for all distributions, regardless of support size!

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors
- [McAllester Schapire '00]
  - w.h.p., $\forall \mu$

$$|S_\mu - G_\mu| = \widetilde{\mathcal{O}}\left(\frac{\mu + 1}{\sqrt{n}}\right)$$

- Holds for all distributions, regardless of support size!
- Good if μ is small

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors
- [McAllester Schapire '00]
  - w.h.p., $\forall \mu$
$$|S_\mu - G_\mu| = \widetilde{\mathcal{O}}\left(\frac{\mu + 1}{\sqrt{n}}\right)$$
- Holds for all distributions, regardless of support size!
- Good if μ is small
- $||S - G||_1 \to 0$?

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors
- [McAllester Schapire '00]
  - w.h.p., $\forall \mu$
$$|S_\mu - G_\mu| = \widetilde{\mathcal{O}}\left(\frac{\mu + 1}{\sqrt{n}}\right)$$
- Holds for all distributions, regardless of support size!
- Good if μ is small
- $||S - G||_1 \to 0$?
  - No, $\mu > \sqrt{n}$?

Previous results

- w.h.p.: with probability $\geq 1 - 1/poly(n)$
- $\widetilde{\mathcal{O}}$: up-to polylogarithmic factors
- [McAllester Schapire '00]
  - w.h.p., $\forall \mu$

$$|S_\mu - G_\mu| = \widetilde{\mathcal{O}}\left(\frac{\mu + 1}{\sqrt{n}}\right)$$

- Holds for all distributions, regardless of support size!
- Good if μ is small
- $||S - G||_1 \to 0$?
  - No, $\mu > \sqrt{n}$?
- Fix?

[Drukh Mansour '05]

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ
  - If $\mu > n^{0.4}$ use empirical estimator

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ
  - If $\mu > n^{0.4}$ use empirical estimator
  - If $\mu \leq n^{0.4}$ use Good-Turing estimator

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ
  - If $\mu > n^{0.4}$ use empirical estimator
  - If $\mu \leq n^{0.4}$ use Good-Turing estimator

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ
  - If $\mu > n^{0.4}$ use empirical estimator
  - If $\mu \le n^{0.4}$ use Good-Turing estimator

$$||S - C||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/6}}\right) \text{ and } D(S||C) = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/3}}\right)$$

- Independent of k!

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ
  - If $\mu > n^{0.4}$ use empirical estimator
  - If $\mu \le n^{0.4}$ use Good-Turing estimator

$$||S - C||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/6}}\right) \text{ and } D(S||C) = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/3}}\right)$$

- Independent of k!
- # of samples for $\ell_1 \le 0.1$ with probability ≥ 0.99

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ
  - If $\mu > n^{0.4}$ use empirical estimator
  - If $\mu \leq n^{0.4}$ use Good-Turing estimator

$$||S - C||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/6}}\right) \text{ and } D(S||C) = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/3}}\right)$$

- Independent of k!
- # of samples for $\ell_1 \leq 0.1$ with probability ≥ 0.99
  - $n \approx 1M$

[Drukh Mansour '05]

- Combined Good-Turing and empirical estimator: C_μ
  - If $\mu > n^{0.4}$ use empirical estimator
  - If $\mu \leq n^{0.4}$ use Good-Turing estimator

$$||S - C||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/6}}\right) \text{ and } D(S||C) = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/3}}\right)$$

- Independent of k!
- # of samples for $\ell_1 \leq 0.1$ with probability ≥ 0.99
  - $n \approx 1M$
- Optimal?

New Results

- Improve Good-Turing/empirical combination bounds?

New Results

- Improve Good-Turing/empirical combination bounds?
- No: $\exists\, p$ such that w.h.p.

$$||S - C||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/6}}\right)$$

New Results

- Improve Good-Turing/empirical combination bounds?
- No: $\exists\, p$ such that w.h.p.

$$||S - C||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/6}}\right)$$

- Estimator with better performance?

New Results

- Improve Good-Turing/empirical combination bounds?
- No: $\exists\, p$ such that w.h.p.

$$||S - C||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/6}}\right)$$

- Estimator with better performance?
- Yes: new estimator F such that w.h.p.

$$||S - F||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/4}}\right)$$

New Results

- Improve Good-Turing/empirical combination bounds?
- No: $\exists\, p$ such that w.h.p.

$$||S - C||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/6}}\right)$$

- Estimator with better performance?
- Yes: new estimator F such that w.h.p.

$$||S - F||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/4}}\right)$$

- Optimal?

New Results

- Improve Good-Turing/empirical combination bounds?
- No: $\exists\, p$ such that w.h.p.

$$||S - C||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/6}}\right)$$

- Estimator with better performance?
- Yes: new estimator F such that w.h.p.

$$||S - F||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/4}}\right)$$

- Optimal?
- Yes: For any $\widehat{S}$, $\exists\, p$ such that w.h.p.

$$||S - \widehat{S}||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/4}}\right)$$

New Results

- Improve Good-Turing/empirical combination bounds?
- No: $\exists\, p$ such that w.h.p.

$$||S - C||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/6}}\right) \text{ and } D(S||C) = \widetilde{\Omega}\left(\frac{1}{n^{1/3}}\right)$$

- Estimator with better performance?
- Yes: new estimator F such that w.h.p.

$$||S - F||_1 = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/4}}\right) \text{ and } D(S||F) = \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/2}}\right)$$

- Optimal?
- Yes: For any $\widehat{S}$, $\exists\, p$ such that w.h.p.

$$||S - \widehat{S}||_1 = \widetilde{\Omega}\left(\frac{1}{n^{1/4}}\right) \text{ and } D(S||\widehat{S}) = \widetilde{\Omega}\left(\frac{1}{n^{1/2}}\right)$$

Observations

Observations

- Hold for any k

Observations

- Hold for any k
- Optimal

Observations

- Hold for any k
- Optimal
- Difference between $\Theta(n^{1/3})$ and $\Theta(n^{1/2})$?

Observations

- Hold for any k
- Optimal
- Difference between $\Theta(n^{1/3})$ and $\Theta(n^{1/2})$?
- Suppose constants are of the same order : NOT shown

Observations

- Hold for any k
- Optimal
- Difference between $\Theta(n^{1/3})$ and $\Theta(n^{1/2})$?
- Suppose constants are of the same order : NOT shown
- Estimate within KL divergence $\delta \approx (0.01)$

Observations

- Hold for any k
- Optimal
- Difference between $\Theta(n^{1/3})$ and $\Theta(n^{1/2})$?
- Suppose constants are of the same order : NOT shown
- Estimate within KL divergence $\delta \approx (0.01)$
  - Good-Turing and empirical: $\delta^{-3} \approx (1M)$

Observations

- Hold for any k
- Optimal
- Difference between $\Theta(n^{1/3})$ and $\Theta(n^{1/2})$?
- Suppose constants are of the same order : NOT shown
- Estimate within KL divergence $\delta \approx (0.01)$
  - Good-Turing and empirical: $\delta^{-3} \approx (1M)$
  - Our approach: $\delta^{-2} \approx (10,000)$
- Computationally efficient: linear time complexity

Observations

- Hold for any k
- Optimal
- Difference between $\Theta(n^{1/3})$ and $\Theta(n^{1/2})$?
- Suppose constants are of the same order : NOT shown
- Estimate within KL divergence $\delta \approx (0.01)$
  - Good-Turing and empirical: $\delta^{-3} \approx (1M)$
  - Our approach: $\delta^{-2} \approx (10,000)$
- Computationally efficient: linear time complexity
- Applications?

Classification

Classification

Classification

- Unknown discrete distributions: p, q

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$$p \to x^3 \quad q \to y^3 \quad z \qquad \text{class.}$$

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$$p \to x^3 \quad q \to y^3 \quad z \quad \text{class.}$$

$$a\, a\, b$$

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$$p \rightarrow x^3 \quad q \rightarrow y^3 \quad z \quad \text{class.}$$

$$a\,a\,b \qquad b\,c\,b$$

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)
		c	

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	$x^3\ (p)$
		b	$y^3\ (q)$
		c	$y^3\ (q)$

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \rightarrow x^3$	$q \rightarrow y^3$	z	class.
$a\,a\,b$	$b\,c\,b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\,a\,b$	$b\,c\,b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications
  - Spam filtering

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications
  - Spam filtering
  - Movie selection

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\,a\,b$	$b\,c\,b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications
  - Spam filtering
  - Movie selection
  - Medical diagnosis

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications
  - Spam filtering
  - Movie selection
  - Medical diagnosis
  - Stock recommendation

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\,a\,b$	$b\,c\,b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications
  - Spam filtering
  - Movie selection
  - Medical diagnosis
  - Stock recommendation
  - ...

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications
  - Spam filtering
  - Movie selection
  - Medical diagnosis
  - Stock recommendation
  - ...
  - Life

Classification

- Unknown discrete distributions: p, q
- Training: $X^n \sim p$ and $Y^n \sim q$
- Test Z: $\sim p$ or q
- For simplicity p, q equally likely

$p \to x^3$	$q \to y^3$	z	class.
$a\ a\ b$	$b\ c\ b$	a	x^3 (p)
		b	y^3 (q)
		c	y^3 (q)
		d	either

- Applications
  - Spam filtering
  - Movie selection
  - Medical diagnosis
  - Stock recommendation
  - ...
  - Life: everything based on experience

Competitive Classification

Competitive Classification

- Optimal classifier?

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"

Competitive Classification

- Optimal classifier?
- Unfortunately, no “entropy”
- Competitive classifier — (nearly) as well as best

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance (ignores training X^n, Y^n)

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p, q) \leq P_E^*(p, q) + \epsilon \quad \forall p, q$

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p,q)$ - lowest error of any classifier for (p,q)
- Requires knowing p,q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p,q) \le P_E^*(p,q) + \epsilon \quad \forall p,q$
  - Typically $\epsilon = \epsilon_{n,k}$

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p, q) \leq P_E^*(p, q) + \epsilon \quad \forall p, q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p, q) \leq P_E^*(p, q) + \epsilon \quad \forall p, q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p, q) \leq P_E^*(p, q) + \epsilon \quad \forall p, q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p, q) \leq P_E^*(p, q) + \epsilon \quad \forall p, q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p, q: uniform over disjoint $k/2$ element subsets of $\{1, \ldots, k\}$

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p,q)$ - lowest error of any classifier for (p,q)
- Requires knowing p,q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p,q) \le P_E^*(p,q) + \epsilon \quad \forall p,q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p,q: uniform over disjoint $k/2$ element subsets of $\{1,\ldots,k\}$
    - e.g. $p = U[1,\ldots,k/2], \quad q = U[k/2+1,\ldots,k]$

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p, q)$ - lowest error of any classifier for (p, q)
- Requires knowing p, q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p, q) \leq P_E^*(p, q) + \epsilon \quad \forall p, q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p, q: uniform over disjoint $k/2$ element subsets of $\{1, \ldots, k\}$
    - e.g. $p = U[1, \ldots, k/2], \quad q = U[k/2 + 1, \ldots, k]$
  - $n = k/4 \quad \rightarrow \quad \Pr(z \text{ does not appear in } x^n \text{ or } y^n) \geq 1/2$

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p,q)$ - lowest error of any classifier for (p,q)
- Requires knowing p,q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p,q) \le P_E^*(p,q) + \epsilon \quad \forall p,q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p,q: uniform over disjoint $k/2$ element subsets of $\{1,\ldots,k\}$
    - e.g. $p = U[1,\ldots,k/2], \quad q = U[k/2+1,\ldots,k]$
  - $n = k/4 \quad \to \quad \Pr(z \text{ does not appear in } x^n \text{ or } y^n) \ge 1/2$
  - $P_E^A \ge 1/4$ for any A

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p,q)$ - lowest error of any classifier for (p,q)
- Requires knowing p,q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p,q) \le P_E^*(p,q) + \epsilon \quad \forall p,q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p,q: uniform over disjoint $k/2$ element subsets of $\{1,\ldots,k\}$
    - e.g. $p = U[1,\ldots,k/2], \quad q = U[k/2+1,\ldots,k]$
  - $n = k/4 \quad \to \quad \Pr(z \text{ does not appear in } x^n \text{ or } y^n) \ge 1/2$
  - $P_E^A \ge 1/4$ for any A
  - $P_E^* =$

Competitive Classification

- Optimal classifier?
- Unfortunately, no “entropy”
- Competitive classifier — (nearly) as well as best
- $P_E^*(p,q)$ - lowest error of any classifier for (p,q)
- Requires knowing p,q in advance (ignores training $X^n,\ Y^n$)
- Classifier A is ϵ-competitive if $P_E^A(p,q) \le P_E^*(p,q) + \epsilon \quad \forall p,q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p,q: uniform over disjoint $k/2$ element subsets of $\{1,\ldots,k\}$
    - e.g. $p = U[1,\ldots,k/2], \quad q = U[k/2+1,\ldots,k]$
  - $n = k/4 \quad \to \quad \Pr(z \text{ does not appear in } x^n \text{ or } y^n) \ge 1/2$
  - $P_E^A \ge 1/4$ for any A
  - $P_E^* = 0$

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p,q)$ - lowest error of any classifier for (p,q)
- Requires knowing p,q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p,q) \le P_E^*(p,q) + \epsilon \quad \forall p,q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p,q: uniform over disjoint $k/2$ element subsets of $\{1,\ldots,k\}$
    - e.g. $p = U[1,\ldots,k/2], \quad q = U[k/2+1,\ldots,k]$
  - $n = k/4 \quad \to \quad \Pr(z \text{ does not appear in } x^n \text{ or } y^n) \ge 1/2$
  - $P_E^A \ge 1/4$ for any A
  - $P_E^* = 0 \qquad$ (for any n)

Competitive Classification

- Optimal classifier?
- Unfortunately, no "entropy"
- Competitive classifier — (nearly) as well as best
- $P_E^*(p,q)$ - lowest error of any classifier for (p,q)
- Requires knowing p,q in advance (ignores training X^n, Y^n)
- Classifier A is ϵ-competitive if $P_E^A(p,q) \leq P_E^*(p,q) + \epsilon \quad \forall p,q$
  - Typically $\epsilon = \epsilon_{n,k}$
- A is *uniformly competitive* if $\epsilon_n \to 0$, regardless of k
- Are there uniformly-competitive classifiers?
  - Given any n (however large), take $k = 4n$
  - p,q: uniform over disjoint $k/2$ element subsets of $\{1,\ldots,k\}$
    - e.g. $p = U[1,\ldots,k/2], \quad q = U[k/2+1,\ldots,k]$
  - $n = k/4 \quad \to \quad \Pr(z \text{ does not appear in } x^n \text{ or } y^n) \geq 1/2$
  - $P_E^A \geq 1/4$ for any A
  - $P_E^* = 0 \qquad$ (for any n)
- No uniformly-competitive classifiers!

Label-Invariant Classification

Label-Invariant Classification

- P_E^* requires knowing p and q in advance

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power,

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

$$x^3 \qquad y^3 \qquad z$$

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

$$x^3 \qquad y^3 \qquad z$$

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

$$\begin{array}{ccc} x^3 & y^3 & z \\ a\,a\,b & c\,b\,a & a \end{array}$$

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

x^3	y^3	z
$a\,a\,b$	$c\,b\,a$	a
$u\,u\,v$	$w\,v\,u$	u

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

$$\begin{array}{ccc} x^3 & y^3 & z \\ a\,a\,b & c\,b\,a & a \\ u\,u\,v & w\,v\,u & u \end{array}$$

- Output in both cases?

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

$$\begin{array}{ccc} x^3 & y^3 & z \\ a\,a\,b & c\,b\,a & a \\ u\,u\,v & w\,v\,u & u \end{array}$$

- Output in both cases? Same!

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

x^3	y^3	z
$a\ a\ b$	$c\ b\ a$	a
$u\ u\ v$	$w\ v\ u$	u

- Output in both cases? Same!
- *Label-invariant*, *canonical*, classifiers

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

$$\begin{array}{ccc} x^3 & y^3 & z \\ a\,a\,b & c\,b\,a & a \\ u\,u\,v & w\,v\,u & u \end{array}$$

- Output in both cases? Same!
- *Label-invariant*, *canonical*, classifiers
- We assume no prior knowledge, all natural classifiers canonical

Label-Invariant Classification

- P^*_E requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

x^3	y^3	z
$a\,a\,b$	$c\,b\,a$	a
$u\,u\,v$	$w\,v\,u$	u

- Output in both cases? Same!
- *Label-invariant*, *canonical*, classifiers
- We assume no prior knowledge, all natural classifiers canonical
- $P^{**}_E(p, q)$ — best error of any label-invariant classifier

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

x^3	y^3	z
$a\,a\,b$	$c\,b\,a$	a
$u\,u\,v$	$w\,v\,u$	u

- Output in both cases? Same!
- *Label-invariant*, *canonical*, classifiers
- We assume no prior knowledge, all natural classifiers canonical
- $P_E^{**}(p, q)$ — best error of any label-invariant classifier
- Also requires knowing p, q in advance

Label-Invariant Classification

- P_E^* requires knowing p and q in advance
- Too much power, no real classifier knows that much!
- Limit to more real classifiers
- Every real classifier is *label invariant* (*canonical*)

x^3	y^3	z
$a\,a\,b$	$c\,b\,a$	a
$u\,u\,v$	$w\,v\,u$	u

- Output in both cases? Same!
- *Label-invariant*, *canonical*, classifiers
- We assume no prior knowledge, all natural classifiers canonical
- $P_E^{**}(p, q)$ — best error of any label-invariant classifier
- Also requires knowing p, q in advance
- Can we find a uniformly-competitive canonical estimator?

Empirical Classifier

Empirical Classifier

- Previous example: $x^3 = a\,a\,b$ $\qquad y^3 = c\,b\,a$ $\qquad z = a$

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?

Empirical Classifier

- Previous example: $x^3 = a\, a\, b \qquad y^3 = c\, b\, a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$

Empirical Classifier

- Previous example: $x^3 = a\ a\ b \qquad y^3 = c\ b\ a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$
  - Recall $X^n \sim p$, $Y^n \sim q$

Empirical Classifier

- Previous example: $x^3 = a\, a\, b \qquad y^3 = c\, b\, a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$
  - Recall $X^n \sim p$, $Y^n \sim q$
  - $n_z(x^n) \geq 1 \to p(z) = 1/n \qquad n_z(y^n) \geq 1 \to q(z) = 1/(2n)$

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$
  - Recall $X^n \sim p$, $Y^n \sim q$
  - $n_z(x^n) \geq 1 \to p(z) = 1/n \qquad n_z(y^n) \geq 1 \to q(z) = 1/(2n)$
  - $n_z(x^n), n_z(y^n) \geq 1$, optimal: $z \sim p$

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$
  - Recall $X^n \sim p$, $Y^n \sim q$
  - $n_z(x^n) \geq 1 \to p(z) = 1/n \qquad n_z(y^n) \geq 1 \to q(z) = 1/(2n)$
  - $n_z(x^n), n_z(y^n) \geq 1$, optimal: $z \sim p$
  - Label invariant

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$
  - Recall $X^n \sim p$, $Y^n \sim q$
  - $n_z(x^n) \geq 1 \to p(z) = 1/n \qquad n_z(y^n) \geq 1 \to q(z) = 1/(2n)$
  - $n_z(x^n), n_z(y^n) \geq 1$, optimal: $z \sim p$
  - Label invariant
  - $\Pr\left(1 \leq n_z(\overline{x}) < n_z(\overline{y})\right) > 0.03$

Empirical Classifier

- Previous example: $x^3 = a\,a\,b \qquad y^3 = c\,b\,a \qquad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$
  - Recall $X^n \sim p$, $Y^n \sim q$
  - $n_z(x^n) \geq 1 \to p(z) = 1/n \qquad n_z(y^n) \geq 1 \to q(z) = 1/(2n)$
  - $n_z(x^n), n_z(y^n) \geq 1$, optimal: $z \sim p$
  - Label invariant
  - $\Pr\left(1 \leq n_z(\overline{x}) < n_z(\overline{y})\right) > 0.03$
  - $P_E^{\text{empirical}}(p, q) > P_E^{**}(p, q) + 0.01$

Empirical Classifier

- Previous example: $x^3 = a\,a\,b$ $\quad y^3 = c\,b\,a$ $\quad z = a$
  - $z \sim x^3 \quad (p)$
- Empirical classifier: assign to training where z appeared more
  - Proxy for distribution with highest probability
  - Label invariant
- Competitive?
  - For arbitrary n, let $p = U[n]$ and $q = U[2n]$
  - Optimal classifier: $z \to p$ if $p(z) > q(z)$, otherwise $z \to q$
  - Recall $X^n \sim p$, $Y^n \sim q$
  - $n_z(x^n) \geq 1 \to p(z) = 1/n \qquad n_z(y^n) \geq 1 \to q(z) = 1/(2n)$
  - $n_z(x^n), n_z(y^n) \geq 1$, optimal: $z \sim p$
  - Label invariant
  - $\Pr(1 \leq n_z(\overline{x}) < n_z(\overline{y})) > 0.03$
  - $P_E^{\text{empirical}}(p,q) > P_E^{**}(p,q) + 0.01$
- Empirical classifier not competitive with label-invariant class.

Competitive Label-Invariant Classifier

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs
- Modify new estimator for sequence-pairs

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs
- Modify new estimator for sequence-pairs
- Label-invariant classifier A such that $\forall p, q$,

$$P_E^A(p,q) \leq P_E^{**}(p,q) + \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/5}}\right)$$

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs
- Modify new estimator for sequence-pairs
- Label-invariant classifier A such that $\forall p, q$,

$$P_E^A(p,q) \leq P_E^{**}(p,q) + \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/5}}\right)$$

- Independent of k!

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs
- Modify new estimator for sequence-pairs
- Label-invariant classifier A such that $\forall p, q$,

$$P_E^A(p,q) \le P_E^{**}(p,q) + \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/5}}\right)$$

- Independent of k!
- Runs in linear time

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs
- Modify new estimator for sequence-pairs
- Label-invariant classifier A such that $\forall p, q$,

$$P_E^A(p,q) \leq P_E^{**}(p,q) + \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/5}}\right)$$

- Independent of k!
- Runs in linear time
- First uniformly-optimal classifier

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs
- Modify new estimator for sequence-pairs
- Label-invariant classifier A such that $\forall p, q$,

$$P_E^A(p,q) \le P_E^{**}(p,q) + \tilde{\mathcal{O}}\left(\frac{1}{n^{1/5}}\right)$$

- Independent of k!
- Runs in linear time
- First uniformly-optimal classifier
- Omniscient oracle too powerful, compare to more realistic one

Competitive Label-Invariant Classifier

- Are there uniformly competitive label-invariant classifiers??
- Relate classification to estimation over sequence-pairs
- Modify new estimator for sequence-pairs
- Label-invariant classifier A such that $\forall p, q$,

$$P_E^A(p,q) \leq P_E^{**}(p,q) + \widetilde{\mathcal{O}}\left(\frac{1}{n^{1/5}}\right)$$

- Independent of k!
- Runs in linear time
- First uniformly-optimal classifier
- Omniscient oracle too powerful, compare to more realistic one
- Lower bound: For any classifier C, $\exists p, q$ such that

$$P_E^C(p,q) \geq P_E^{**}(p,q) + \widetilde{\Omega}(\frac{1}{n^{1/3}})$$

Experiments

Experiments

- Netflix challenge: 10% $\rightarrow$ \$1M

Experiments

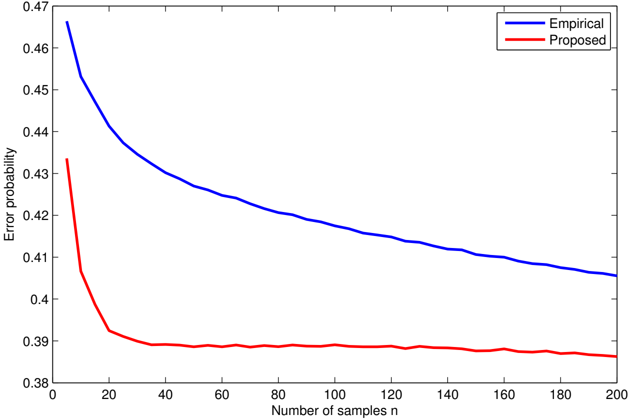
- Netflix challenge: 10% $\rightarrow$ \$1M
- Zipf distributions $p_i \propto i^{-s}$, $s = 1$ and $s = 1.5$, $k = 100$

Experiments

- Netflix challenge: 10% $\rightarrow$ \$1M
- Zipf distributions $p_i \propto i^{-s}$, $s = 1$ and $s = 1.5$, $k = 100$

Experiments

- Netflix challenge: 10% $\rightarrow$ \$1M
- Zipf distributions $p_i \propto i^{-s}$, $s = 1$ and $s = 1.5$, $k = 100$



Prediction / Universal Compression

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern
- $x^5 = a\,b\,b\,a\,c$

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern
- $x^5 = a\, b\, b\, a\, c$
- Dict: $a \to 1, b \to 2, c \to 3$ and pattern: $1\,2\,2\,1\,3$

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern
- $x^5 = a\,b\,b\,a\,c$
- Dict: $a \to 1, b \to 2, c \to 3$ and pattern: $1\,2\,2\,1\,3$
- Redundancy of patterns?

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern
- $x^5 = a\, b\, b\, a\, c$
- Dict: $a \to 1, b \to 2, c \to 3$ and pattern: 1 2 2 1 3
- Redundancy of patterns?
  - (ADO '12): $\widetilde{\mathcal{O}}(n^{1/3})$

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern
- $x^5 = a\,b\,b\,a\,c$
- Dict: $a \to 1, b \to 2, c \to 3$ and pattern: 1 2 2 1 3
- Redundancy of patterns?
  - (ADO '12): $\widetilde{\mathcal{O}}(n^{1/3})$
- Computationally efficient sequential algorithms?

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern
- $x^5 = a\,b\,b\,a\,c$
- Dict: $a \to 1, b \to 2, c \to 3$ and pattern: 1 2 2 1 3
- Redundancy of patterns?
  - (ADO '12): $\widetilde{\mathcal{O}}(n^{1/3})$
- Computationally efficient sequential algorithms?
  - (OSZ '03): $\mathcal{O}(n^{2/3})$

Universal Compression

- X^n: generated by unknown *i.i.d.* distribution
- Code designed for distribution q
- Redundancy

$$R = \min_q \max_p \mathbb{E}\left[\log \frac{p}{q}\right]$$

- Compress sequences: compress dictionary + pattern
- $x^5 = a\,b\,b\,a\,c$
- Dict: $a \to 1, b \to 2, c \to 3$ and pattern: 1 2 2 1 3
- Redundancy of patterns?
  - (ADO '12): $\widetilde{\mathcal{O}}(n^{1/3})$
- Computationally efficient sequential algorithms?
  - (OSZ '03): $\mathcal{O}(n^{2/3})$
- New bound: $\widetilde{\mathcal{O}}(n^{1/2})$

Proof Sketch

Motivation

Motivation

- N_μ: # of symbols appearing μ times

Motivation

- N_μ: # of symbols appearing μ times
- Empirical
$$E_\mu = N_\mu \frac{\mu}{n}$$

Motivation

- N_μ: # of symbols appearing μ times
- Empirical

$$E_\mu = N_\mu \frac{\mu}{n}$$

- Multiply by a correction term c_μ to improve the estimate

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

Motivation

- N_μ: # of symbols appearing μ times
- Empirical

$$E_\mu = N_\mu \frac{\mu}{n}$$

- Multiply by a correction term c_μ to improve the estimate

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

  - c_μ: a function of x^n

Motivation

- N_μ: # of symbols appearing μ times
- Empirical
$$E_\mu = N_\mu \frac{\mu}{n}$$
- Multiply by a correction term c_μ to improve the estimate
$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$
  - c_μ: a function of x^n
- Ignoring constants:
$$|S_\mu - \widehat{S}_\mu| \approx \text{ bias} + \sqrt{\text{variance}}$$

New estimator

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical			

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1		

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing			

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$		

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$	0	$\mathbb{E}[N_\mu]\frac{(\mu+1)^2}{n^2}$

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$	0	$\mathbb{E}[N_\mu]\frac{(\mu+1)^2}{n^2}$
New			

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$	0	$\mathbb{E}[N_\mu]\frac{(\mu+1)^2}{n^2}$
New	$\frac{\mu+1}{\mu}\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$		

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$	0	$\mathbb{E}[N_\mu]\frac{(\mu+1)^2}{n^2}$
New	$\frac{\mu+1}{\mu}\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$	0	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$	0	$\mathbb{E}[N_\mu]\frac{(\mu+1)^2}{n^2}$
New	$\frac{\mu+1}{\mu}\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$	0	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$

- Best of both estimators!

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$	0	$\mathbb{E}[N_\mu]\frac{(\mu+1)^2}{n^2}$
New	$\frac{\mu+1}{\mu}\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$	0	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$

- Best of both estimators!
- Idealized as we don't know the expectations

New estimator

$$\widehat{S}_\mu = N_\mu \frac{\mu}{n} c_\mu$$

- c_μ: a function of x^n

Estimator	c_μ	Bias	Variance
Empirical	1	$\mathbb{E}[N_\mu]\frac{\sqrt{\mu}}{n}$	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$
Good-Turing	$\frac{\mu+1}{\mu}\frac{N_{\mu+1}}{N_\mu}$	0	$\mathbb{E}[N_\mu]\frac{(\mu+1)^2}{n^2}$
New	$\frac{\mu+1}{\mu}\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$	0	$\mathbb{E}[N_\mu]\frac{\mu}{n^2}$

- Best of both estimators!
- Idealized as we don't know the expectations
- How to estimate $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$?

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_{\mu}]}$

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$
  - Expected # of symbols appearing μ times

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$
  - Expected # of symbols appearing μ times
- Good-Turing: $\mathbb{E}[N_\mu] \sim N_\mu$, high variance

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$
  - Expected # of symbols appearing μ times
- Good-Turing: $\mathbb{E}[N_\mu] \sim N_\mu$, high variance
- Better estimators for $\mathbb{E}[N_\mu]$

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$
  - Expected # of symbols appearing μ times
- Good-Turing: $\mathbb{E}[N_\mu] \sim N_\mu$, high variance
- Better estimators for $\mathbb{E}[N_\mu]$
- Given: X^n or $N_0, N_1, \ldots N_n$

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$
  - Expected # of symbols appearing μ times
- Good-Turing: $\mathbb{E}[N_\mu] \sim N_\mu$, high variance
- Better estimators for $\mathbb{E}[N_\mu]$
- Given: X^n or $N_0, N_1, \ldots N_n$
- Linear?

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$
  - Expected # of symbols appearing μ times
- Good-Turing: $\mathbb{E}[N_\mu] \sim N_\mu$, high variance
- Better estimators for $\mathbb{E}[N_\mu]$
- Given: X^n or $N_0, N_1, \ldots N_n$
- Linear?
  - $\sum_\mu h_\mu N_\mu$

Estimating $\frac{\mathbb{E}[N_{\mu+1}]}{\mathbb{E}[N_\mu]}$

- Given: sequence X^n, estimate $\mathbb{E}[N_\mu]$
  - Expected # of symbols appearing μ times
- Good-Turing: $\mathbb{E}[N_\mu] \sim N_\mu$, high variance
- Better estimators for $\mathbb{E}[N_\mu]$
- Given: X^n or $N_0, N_1, \ldots N_n$
- Linear?
  - $\sum_\mu h_\mu N_\mu$
- Why should it work?

Linear estimator

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias $= 0$ variance $= \mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias $= 0$ variance $= \mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias = 0 variance = $\mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Expected # symbols appearing 100 and 101 times are close

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias = 0 variance = $\mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Expected # symbols appearing 100 and 101 times are close
- Momentarily assume: $N_{\mu-1}, N_\mu, N_{\mu+1}$ independent

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias = 0 variance = $\mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \le \epsilon$
  - Expected # symbols appearing 100 and 101 times are close
- Momentarily assume: $N_{\mu-1}, N_\mu,\ N_{\mu+1}$ independent
- New Estimator

$$\left(N_{\mu-1} + N_\mu + N_{\mu+1}\right)/3$$

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias = 0 variance = $\mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \le \epsilon$
  - Expected # symbols appearing 100 and 101 times are close
- Momentarily assume: $N_{\mu-1}, N_\mu, N_{\mu+1}$ independent
- New Estimator

$$\left(N_{\mu-1} + N_\mu + N_{\mu+1}\right)/3$$

- Bias $\le 2\epsilon/3 \le \epsilon$

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias $= 0$ variance $= \mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Expected # symbols appearing 100 and 101 times are close
- Momentarily assume: $N_{\mu-1}, N_\mu,\ N_{\mu+1}$ independent
- New Estimator

$$\left(N_{\mu-1} + N_\mu + N_{\mu+1}\right)/3$$

- Bias $\leq 2\epsilon/3 \leq \epsilon$
- Variance of sum $=$ sum of variances

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias $= 0$ variance $= \mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \le \epsilon$
  - Expected # symbols appearing 100 and 101 times are close
- Momentarily assume: $N_{\mu-1}, N_\mu, N_{\mu+1}$ independent
- New Estimator

$$\left(N_{\mu-1} + N_\mu + N_{\mu+1}\right)/3$$

- Bias $\le 2\epsilon/3 \le \epsilon$
- Variance of sum $=$ sum of variances
- $\sigma' = \sigma/\sqrt{3}$

$$\text{error} \le \frac{1}{\sqrt{3}}\sqrt{\mathbb{E}[N_\mu]} + \epsilon$$

Linear estimator

- Simple estimator for $\mathbb{E}[N_\mu]$: N_μ
- Bias $=0$ variance $=\mathbb{E}[N_\mu]$

$$|N_\mu - \mathbb{E}[N_\mu]| = \sqrt{\mathbb{E}[N_\mu]}$$

- $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \le \epsilon$
  - Expected # symbols appearing 100 and 101 times are close
- Momentarily assume: $N_{\mu-1}, N_\mu, N_{\mu+1}$ independent
- New Estimator

$$\left(N_{\mu-1} + N_\mu + N_{\mu+1}\right)/3$$

- Bias $\le 2\epsilon/3 \le \epsilon$
- Variance of sum $=$ sum of variances
- $\sigma' = \sigma/\sqrt{3}$

$$\text{error} \le \frac{1}{\sqrt{3}}\sqrt{\mathbb{E}[N_\mu]} + \epsilon$$

- Improvement

Technical Details

Technical Details

- $N_{\mu-1}$, N_{μ}, $N_{\mu+1}$ are not independent

Technical Details

- $N_{\mu-1}$, N_{μ}, $N_{\mu+1}$ are not independent
- Need to show

Technical Details

- $N_{\mu-1}$, N_{μ}, $N_{\mu+1}$ are not independent
- Need to show
  - $|\mathbb{E}[N_{\mu}] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_{\mu}] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$

Technical Details

- $N_{\mu-1}$, N_μ, $N_{\mu+1}$ are not independent
- Need to show
  - $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Bounds on bias, variance are enough for concentration

Technical Details

- $N_{\mu-1}$, N_μ, $N_{\mu+1}$ are not independent
- Need to show
  - $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Bounds on bias, variance are enough for concentration
- Simple averaging does not yield optimal estimator

Technical Details

- $N_{\mu-1}$, N_μ, $N_{\mu+1}$ are not independent
- Need to show
  - $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Bounds on bias, variance are enough for concentration
- Simple averaging does not yield optimal estimator
- Explicit estimator such that bias and variance is optimized

Technical Details

- $N_{\mu-1}$, N_μ, $N_{\mu+1}$ are not independent
- Need to show
  - $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Bounds on bias, variance are enough for concentration
- Simple averaging does not yield optimal estimator
- Explicit estimator such that bias and variance is optimized
  - Properties of Poisson functions, distribution approximations

Technical Details

- $N_{\mu-1}$, N_{μ}, $N_{\mu+1}$ are not independent
- Need to show
  - $|\mathbb{E}[N_{\mu}] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_{\mu}] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Bounds on bias, variance are enough for concentration
- Simple averaging does not yield optimal estimator
- Explicit estimator such that bias and variance is optimized
  - Properties of Poisson functions, distribution approximations
  - Adaptively choose the # of non-zero coefficients based on X^n

Technical Details

- $N_{\mu-1}$, N_{μ}, $N_{\mu+1}$ are not independent
- Need to show
  - $|\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu+1}]|, |\mathbb{E}[N_\mu] - \mathbb{E}[N_{\mu-1}]| \leq \epsilon$
  - Bounds on bias, variance are enough for concentration
- Simple averaging does not yield optimal estimator
- Explicit estimator such that bias and variance is optimized
  - Properties of Poisson functions, distribution approximations
  - Adaptively choose the # of non-zero coefficients based on X^n
- Converse: show that estimation is hard for some distributions

Estimator Properties

Estimator Properties

- Linear estimator for $\mathbb{E}[N_\mu]$: $\sum_{|i|\leq r} h_i N_{\mu+i}$

Estimator Properties

- Linear estimator for $\mathbb{E}[N_\mu]$: $\sum_{|i|\le r} h_i N_{\mu+i}$
- Bias: $\mathbb{E}[N_\mu - \sum_{|i|\le r} h_i N_{\mu+i}]$

$$\mathbb{E}[N_\mu] = \sum_x \binom{n}{\mu} p_x^\mu (1-p_x)^{n-\mu}$$

Estimator Properties

- Linear estimator for $\mathbb{E}[N_\mu]$: $\sum_{|i|\leq r} h_i N_{\mu+i}$
- Bias: $\mathbb{E}[N_\mu - \sum_{|i|\leq r} h_i N_{\mu+i}]$

$$\mathbb{E}[N_\mu] = \sum_x \binom{n}{\mu} p_x^\mu (1-p_x)^{n-\mu}$$

- Problem

Estimator Properties

- Linear estimator for $\mathbb{E}[N_\mu]$: $\sum_{|i|\le r} h_i N_{\mu+i}$
- Bias: $\mathbb{E}[N_\mu - \sum_{|i|\le r} h_i N_{\mu+i}]$

$$\mathbb{E}[N_\mu] = \sum_x \binom{n}{\mu} p_x^\mu (1-p_x)^{n-\mu}$$

- Problem
  - After rescaling, contribution of symbol with probability p

$$\binom{n}{\mu} p^\mu (1-p)^{n-\mu} \sum_{|i|\le r} h_i' \left(\frac{np}{\mu}\right)^i$$

Estimator Properties

- Linear estimator for $\mathbb{E}[N_\mu]$: $\sum_{|i|\le r} h_i N_{\mu+i}$
- Bias: $\mathbb{E}[N_\mu - \sum_{|i|\le r} h_i N_{\mu+i}]$

$$\mathbb{E}[N_\mu] = \sum_x \binom{n}{\mu} p_x^\mu (1-p_x)^{n-\mu}$$

- Problem
  - After rescaling, contribution of symbol with probability p

$$\binom{n}{\mu} p^\mu (1-p)^{n-\mu} \sum_{|i|\le r} h'_i \left(\frac{np}{\mu}\right)^i$$

    - h'_i: scaled version of h_is

Estimator Properties

- Linear estimator for $\mathbb{E}[N_\mu]$: $\sum_{|i|\leq r} h_i N_{\mu+i}$
- Bias: $\mathbb{E}[N_\mu - \sum_{|i|\leq r} h_i N_{\mu+i}]$

$$\mathbb{E}[N_\mu] = \sum_x \binom{n}{\mu} p_x^\mu (1-p_x)^{n-\mu}$$

- Problem
  - After rescaling, contribution of symbol with probability p

$$\binom{n}{\mu} p^\mu (1-p)^{n-\mu} \sum_{|i|\leq r} h_i' \left(\frac{np}{\mu}\right)^i$$

    - h_i': scaled version of h_is
  - Variance $\propto \max_i h_i^2$

Estimator Properties

- Linear estimator for $\mathbb{E}[N_\mu]$: $\sum_{|i|\leq r} h_i N_{\mu+i}$
- Bias: $\mathbb{E}[N_\mu - \sum_{|i|\leq r} h_i N_{\mu+i}]$

$$\mathbb{E}[N_\mu] = \sum_x \binom{n}{\mu} p_x^\mu (1-p_x)^{n-\mu}$$

- Problem
  - After rescaling, contribution of symbol with probability p

$$\binom{n}{\mu} p^\mu (1-p)^{n-\mu} \sum_{|i|\leq r} h'_i \left(\frac{np}{\mu}\right)^i$$

    - h'_i: scaled version of h_is
  - Variance $\propto \max_i h_i^2$
  - Set term close to $\binom{n}{\mu} p^\mu (1-p)^{n-\mu}$ s.t. $\max_i h(i)$ is bounded

A polynomial problem

A polynomial problem

- Approximating a polynomial with bounded co-efficients

A polynomial problem

- Approximating a polynomial with bounded co-efficients
- Let $x = \frac{np}{\mu} \approx 1$

A polynomial problem

- Approximating a polynomial with bounded co-efficients
- Let $x = \frac{np}{\mu} \approx 1$
- Minimize

$$\delta = \max_{x \in (1-\epsilon, 1+\epsilon)} \left| 1 - \sum_{i=-r}^{r} h_i x^i \right|$$

$$\text{s.t. } \max |h_i| \leq \frac{c}{r+1}, \quad |h_i - h_{i-1}| = \frac{c}{(r+1)^2}$$

A polynomial problem

- Approximating a polynomial with bounded co-efficients
- Let $x = \frac{np}{\mu} \approx 1$
- Minimize

$$\delta = \max_{x \in (1-\epsilon, 1+\epsilon)} \left| 1 - \sum_{i=-r}^{r} h_i x^i \right|$$

$$\text{s.t.} \quad \max |h_i| \leq \frac{c}{r+1}, \quad |h_i - h_{i-1}| = \frac{c}{(r+1)^2}$$

- $\sum_i h_i = 1 \implies \delta = \mathcal{O}(\epsilon)$

A polynomial problem

- Approximating a polynomial with bounded co-efficients
- Let $x = \frac{np}{\mu} \approx 1$
- Minimize

$$\delta = \max_{x \in (1-\epsilon, 1+\epsilon)} \left| 1 - \sum_{i=-r}^{r} h_i x^i \right|$$
$$\text{s.t.} \quad \max |h_i| \leq \frac{c}{r+1}, \quad |h_i - h_{i-1}| = \frac{c}{(r+1)^2}$$

- $\sum_i h_i = 1 \implies \delta = \mathcal{O}(\epsilon)$
- By symmetry, $h_i = h_{-i} \implies \delta = \mathcal{O}(\epsilon^2)$

A polynomial problem

- Approximating a polynomial with bounded co-efficients
- Let $x = \frac{np}{\mu} \approx 1$
- Minimize

$$\delta = \max_{x \in (1-\epsilon, 1+\epsilon)} \left| 1 - \sum_{i=-r}^{r} h_i x^i \right|$$

$$\text{s.t.} \quad \max |h_i| \leq \frac{c}{r+1}, \quad |h_i - h_{i-1}| = \frac{c}{(r+1)^2}$$

  - $\sum_i h_i = 1 \implies \delta = \mathcal{O}(\epsilon)$
  - By symmetry, $h_i = h_{-i} \implies \delta = \mathcal{O}(\epsilon^2)$
  - $\sum_i h_i i^2 = 0, \implies \delta = \mathcal{O}(\epsilon^4)$

A polynomial problem

- Approximating a polynomial with bounded co-efficients
- Let $x = \frac{np}{\mu} \approx 1$
- Minimize

$$\delta = \max_{x \in (1-\epsilon, 1+\epsilon)} \left| 1 - \sum_{i=-r}^{r} h_i x^i \right|$$

$$\text{s.t.} \quad \max |h_i| \leq \frac{c}{r+1}, \quad |h_i - h_{i-1}| = \frac{c}{(r+1)^2}$$

  - $\sum_i h_i = 1 \implies \delta = \mathcal{O}(\epsilon)$
  - By symmetry, $h_i = h_{-i} \implies \delta = \mathcal{O}(\epsilon^2)$
  - $\sum_i h_i i^2 = 0, \implies \delta = \mathcal{O}(\epsilon^4)$
- $\sum_i h_i = 1$, $\sum_i h_i i^2 = 0$, and $h_r = 0$ uniquely represents a second degree polynomial of the form $h_i = \alpha i^2 + \beta i + \gamma$ and satisfies above conditions

A polynomial problem

- Approximating a polynomial with bounded co-efficients
- Let $x = \frac{np}{\mu} \approx 1$
- Minimize

$$\delta = \max_{x \in (1-\epsilon, 1+\epsilon)} \left| 1 - \sum_{i=-r}^{r} h_i x^i \right|$$
$$\text{s.t.} \quad \max |h_i| \le \frac{c}{r+1}, \quad |h_i - h_{i-1}| = \frac{c}{(r+1)^2}$$

  - $\sum_i h_i = 1 \implies \delta = \mathcal{O}(\epsilon)$
  - By symmetry, $h_i = h_{-i} \implies \delta = \mathcal{O}(\epsilon^2)$
  - $\sum_i h_i i^2 = 0, \implies \delta = \mathcal{O}(\epsilon^4)$
- $\sum_i h_i = 1$, $\sum_i h_i i^2 = 0$, and $h_r = 0$ uniquely represents a second degree polynomial of the form $h_i = \alpha i^2 + \beta i + \gamma$ and satisfies above conditions
- Choose r to minimize bias-variance tradeoff

Putting pieces back together

Putting pieces back together

- The error: bias $+\sqrt{\text{variance}}$

Putting pieces back together

- The error: bias $+\sqrt{\text{variance}}$
  - Good-Turing: $\frac{\sqrt{N_\mu}\mu}{n}$

Putting pieces back together

- The error: bias $+\sqrt{\text{variance}}$
  - Good-Turing: $\frac{\sqrt{N_\mu}\mu}{n}$
  - Empirical: $\frac{N_\mu\sqrt{\mu}}{n}$

Putting pieces back together

- The error: bias $+\sqrt{\text{variance}}$
  - Good-Turing: $\frac{\sqrt{N_\mu}\mu}{n}$
  - Empirical: $\frac{N_\mu\sqrt{\mu}}{n}$
- New error: $\frac{N_\mu^{3/4}\sqrt{\mu}}{n}$

Putting pieces back together

- The error: bias $+\sqrt{\text{variance}}$
  - Good-Turing: $\frac{\sqrt{N_\mu}\mu}{n}$
  - Empirical: $\frac{N_\mu\sqrt{\mu}}{n}$
- New error: $\frac{N_\mu^{3/4}\sqrt{\mu}}{n}$
- Adding over all multiplicities and maximizing for N_μ yields

$$\widetilde{\mathcal{O}}(n^{-1/4})$$

Putting pieces back together

- The error: bias $+\sqrt{\text{variance}}$
  - Good-Turing: $\frac{\sqrt{N_\mu}\mu}{n}$
  - Empirical: $\frac{N_\mu\sqrt{\mu}}{n}$
- New error: $\frac{N_\mu^{3/4}\sqrt{\mu}}{n}$
- Adding over all multiplicities and maximizing for N_μ yields

$$\widetilde{\mathcal{O}}(n^{-1/4})$$

- $\forall$ estimator there is a distribution with error $\widetilde{\Omega}(n^{-1/4})$

Summary

Summary

- Probability estimation

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of $\#$ samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification
  - Can't compete with oracle classifier that knows p, q

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification
  - Can't compete with oracle classifier that knows p, q
  - Label-invariant classifiers, or oracle knows multisets

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification
  - Can't compete with oracle classifier that knows p, q
  - Label-invariant classifiers, or oracle knows multisets
  - Proposed classifier: additional error $\widetilde{\mathcal{O}}(n^{-1/5})$

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification
  - Can't compete with oracle classifier that knows p, q
  - Label-invariant classifiers, or oracle knows multisets
  - Proposed classifier: additional error $\widetilde{\mathcal{O}}(n^{-1/5})$
  - Independent of alphabet size

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification
  - Can't compete with oracle classifier that knows p, q
  - Label-invariant classifiers, or oracle knows multisets
  - Proposed classifier: additional error $\widetilde{\mathcal{O}}(n^{-1/5})$
  - Independent of alphabet size
  - Converse: additional error $\widetilde{\Omega}(n^{-1/3})$

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification
  - Can't compete with oracle classifier that knows p, q
  - Label-invariant classifiers, or oracle knows multisets
  - Proposed classifier: additional error $\widetilde{\mathcal{O}}(n^{-1/5})$
  - Independent of alphabet size
  - Converse: additional error $\widetilde{\Omega}(n^{-1/3})$
- Prediction/universal compression

Summary

- Probability estimation
  - Estimating p_x requires $n = \Theta(k)$
  - Estimating S_μ independent of k
  - ℓ_1 distance as function of # samples
  - Good-Turing: $\widetilde{\mathcal{O}}(n^{-1/6})$
  - Proposed estimator: $\widetilde{\mathcal{O}}(n^{-1/4})$
  - Optimal
  - Linear-time complexity
- Classification
  - Can't compete with oracle classifier that knows p, q
  - Label-invariant classifiers, or oracle knows multisets
  - Proposed classifier: additional error $\widetilde{\mathcal{O}}(n^{-1/5})$
  - Independent of alphabet size
  - Converse: additional error $\widetilde{\Omega}(n^{-1/3})$
- Prediction/universal compression
  - Per-symbol redundancy $\widetilde{\mathcal{O}}(n^{-1/2})$

Xie Xie