Sparse Adaptive Dirichlet-Multinomial-like Processes

Marcus Hutter

Canberra, ACT, 0200, Australia
http://www.hutter1.net/

THE AUSTRALIAN NATIONAL UNIVERSITY

2013

Abstract

Online estimation and modelling of i.i.d. data for short sequences over large or complex "alphabets" is a ubiquitous (sub)problem in machine learning, information theory, data compression, statistical language processing, and document analysis. The Dirichlet-Multinomial distribution (also called Polya urn scheme) and extensions thereof are widely applied for online i.i.d. estimation. Good a-priori choices for the parameters in this regime are difficult to obtain though. I present an optimal adaptive choice for the main parameter via tight, data-dependent redundancy bounds for a related model. The 1-line recommendation is to set the 'total mass' = 'precision' = 'concentration' parameter to $m/[2\ln\frac{n+1}{m}]$, where n is the (past) sample size and m the number of different symbols observed (so far). The resulting estimator is simple, online, fast, and experimental performance is superb.

Keywords: sparse coding; adaptive parameters; Dirichlet-Multinomial; Polya urn; data-dependent redundancy bound; small/large alphabet; data compression.

Contents

- The Dirichlet-Multinomial distribution
- Main new related model S^β
- Optimizing the concentration parameter β
- CodeLength and Redundancy of S for optimal β^*
- Algorithms & Computation Time
- Experiments on Artificial and Real Data
- Discussion, Summary, Conclusion, References

Problem Setup

- Data: Short sequence over large alphabet from unknown source.
- Regime: Base alphabet $\mathcal{X}$ larger than sequences length n.
- Problem: Estimation, Modelling, Prediction, *Compression*.
- Online alg: Predict next symbol x_{t+1} given only past symbols $x_{1:t}$.
- Applications: machine learning, information theory, data compression, language modelling, document analysis.
- I.i.d: Assume unknown i.i.d. sampling distribution. Data often not i.i.d. but subsequence with given context is (closer to) i.i.d.
- Example: Typical documents comprise a small fraction of the available 100 000+ English words,
  and words have different length/complexity/frequency.
- Problem pronounced in n-gram models: Many counts are zero. Subsequence for given context can be very short.

The Dirichlet-Multinomial Distribution

$=$ generalized Laplace rule $=$ Carnap's inductive inference scheme
$=$ Polya urn scheme $=$ Chinese restaurant process

$$\text{DirM}(x_{n+1} = i | x_{1:n}) = \frac{n_i + \alpha_i}{n + \alpha_+}$$

- n_i = number of times $i \in \mathcal{X}$ appeared in $x_{1:n} \equiv (x_1, ..., x_n)$.
- α_i = parameter = fictitious prior counts of i.
- $\alpha_+ := \sum_{i \in \mathcal{X}} \alpha_i$ = total mass = precision = concentration.

Theoretically motivated choices for α_i (all equal by symmetry):

Dirichlet	Laplace	KT&others	Perks	Haldane	Hutter
$\alpha_i = \frac{\alpha_+}{\lvert\mathcal{X}\rvert}$	1	$\frac{1}{2}$	$\frac{1}{\lvert\mathcal{X}\rvert}$	0	$\frac{m}{2\lvert\mathcal{X}\rvert \ln \frac{n+1}{m}}$

- They are all problematic for large base alphabet $\mathcal{X}$.
- Existing solutions: empirically optimize or sample or average $\boldsymbol{\alpha}$.
- New solution (last column): Analytically optimize exact redundancy.
  m is the number of different symbols that appear in $x_{1:n}$.

Main Contribution

- Introduce an estimator S closely related to DirM
  but easier to analyze and slightly superior.
- Reserve escape probability to symbols not seen so far.
- Derive optimal adaptive escape parameter $\beta\widehat{=}\alpha_+$
  based on data-dependent redundancy,
  rather than expected or worst-case bounds.

The resulting estimator:

(*i*) is simple, (*ii*) online, (*iii*) fast,
(*iv*) performs well for all m, small, middle and large,
(*v*) is independent of the base alphabet size,
(*vi*) non-occurring symbols induce no redundancy,
(*vii*) the constant sequence has constant redundancy,
(*viii*) symbols that appear only finitely often have
bounded/constant contribution to the redundancy,
(*ix*) is competitive with (slow) Bayesian mixing over all sub-alphabets.

Main Model S

$$S(x_{t+1}=i|x_{1:t}) \;:=\; \begin{cases} \dfrac{n_i^t}{t+\beta_t} & \text{for} \quad n_i^t>0 \\[2ex] \dfrac{\beta_t w_i^t}{t+\beta_t} & \text{for} \quad n_i^t=0 \end{cases}$$

- β_t = concentration parameter.
- w_i^t = weight of new symbol i at time t.
- n_i^t = number of times i appears in $x_{1:t}$.

Difference to $\text{DirM}(x_{t+1}=i|x_{1:t})=\frac{n_i^t+\beta w_i}{t+\beta}$:

- Cases instead of sum.
- Time-dependent parameters.

Closed-form of joint sequence probability for constant β (Γ=Gamma fct.):

$$S(x_{1:n}) \;=\; \prod_{t=0}^{n-1} S(x_{t+1}|x_{1:t}) \;=\; \beta^{|\mathcal{A}|}\frac{\Gamma(\beta)}{\Gamma(n+\beta)} \prod_{t:n_{x_{t+1}}^t>0} w_{x_{t+1}}^t \prod_{j\in\mathcal{A}} \Gamma(n_j)$$

- $\mathcal{A}=\{x_1,...,x_n\}$ = symbols actually appearing in $x_{1:n}$.

CodeLength and Redundancy

Performance measure(s):

Code Length = –log-likelihood = $n \times \log(\text{perplexity})$

$$\text{CL}_S(x_{1:n}) \;:=\; \ln 1/S(x_{1:n}) \;=\; n \times \ln[1/S(x_{1:n})^{1/n}]$$

$\stackrel{+}{=}$ Redundancy = log-loss regret w.r.t. ML i.i.d. source:

$$R_S(a_{1:n}) \;:=\; \text{CL}_S(x_{1:n}) - n\,H(\hat{\boldsymbol{\theta}}), \quad \text{where} \quad \hat{\theta}_i := n_i/n$$

Optimal Constant β

Code length $\mathrm{CL}_S^\beta(x_{1:n})$ is minimized for

$$0 \stackrel{!}{=} \frac{\partial \mathrm{CL}_S^\beta(x_{1:n})}{\partial\beta} = -\frac{m}{\beta} + \Psi(n+\beta) - \Psi(\beta)$$

where $\Psi(x) := \mathrm{d}\ln\Gamma(x)/\mathrm{d}x$ is the diGamma function.

Approximate solution: $\beta^{min} \approx \beta^* := \dfrac{m}{2\ln\frac{n}{m}}$

Discussion: $m \gg \ln n \Rightarrow$ "frequently" new symbols $\Rightarrow$ reserve more probability mass for new symbols $\Rightarrow$ make β large. $\surd$.

Discussion: $m \ll \ln n \Rightarrow$ new symbol rare $\Rightarrow$ reserve most probability mass for old symbols $\Rightarrow$ make β small. $\surd$.

More regimes ($0 < c < \infty$ and $0 \leq \alpha < 1$ and $n \to \infty$):

m	$\to c$	$\propto \ln n$	$\propto n^\alpha$	$\propto n$	$\geq n - c$	$= n$
β^*	$\sim c/2\ln n$	$\to c$	$\propto n^\alpha/\ln n$	$\propto n$	$\propto n^2$	∞

Redundancy of S for "optimal" constant β^*

$$R_S^{\beta^*}(x_{1:n}) \;\leq\; \underbrace{\mathrm{CL}_w(\mathcal{A}) - m\ln m}_{\text{CL of unsorted }\mathcal{A}} + \sum_{j\in\mathcal{A}} \underbrace{\tfrac{1}{2}\ln n_j}_{R\text{ of }j} + \underbrace{m\ln\ln\tfrac{en}{m} + 0.6m}_{\text{small}}$$

- Similar lower bound for all β exists with different constants.
- Bound also holds for DirM with matching parameters.
- Bound is independent of base alphabet size D.
  $\Rightarrow$ Holds even for infinite and continuous alphabet $\mathcal{X}$.
  The weights w_i^t become (sub)probability densities.
- Extreme $m \approx n \approx D$: Redundancy is negative!
  Code is better than ML i.i.d. oracle!
- Extreme $m = 1$: Constant sequence $x_t = j \forall t \quad\Rightarrow\quad \beta^* = 1/2 \ln n$,
  $\mathrm{CL}_S^{\beta^*} = \mathrm{CL}_w(j) + 1 =$ theoretical optimum $=$ finite. Similarly $m \leq c$.

Code Length of Used Alphabet $\mathcal{A}$

- Code Length of ordered $\mathcal{A}$ is $\mathrm{CL}_w(\mathcal{A}) := \sum_{t:n^t_{x_{t+1}}=0} \ln(1/w^t_{x_{t+1}})$
- Interpretation: Whenever we see a new symbol $x_{t+1} \notin \{x_1, ..., x_t\}$,

  we code it in $\ln(1/w^t_{x_{t+1}})$ nits.
- Of course, arithmetic coding with S does *not* work like this.
- Example: Uniform: $w^t_i = \frac{1}{D-m_t} \quad \Rightarrow \quad \mathrm{CL}_w(\mathcal{A}) = \ln \frac{D!}{(D-m)!}$

  $\Rightarrow \mathrm{CL}_w(\mathcal{A}) - m \ln m \;\approx\; \ln \binom{D}{m}$ = CL of unordered $\mathcal{A}$.
- Code-length based: $w^t_i = \mathrm{e}^{-\mathrm{CL}(i)} \quad \Rightarrow \quad \mathrm{CL}_w(\mathcal{A}) = \sum_{j\in\mathcal{A}} \mathrm{CL}(j)$,

  $\mathrm{CL}(j)$ is some prefix-free code length of new symbol j.

Code length of frequencies n_j

$$\sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln n_j \quad \overset{(1a)}{\leq} \quad \frac{m}{2}\ln\frac{n}{m} \quad \overset{(1b)}{\leq} \quad \frac{m}{2}\ln n \quad \overset{(2)}{\leq} \quad \frac{D}{2}\ln n$$

- R.h.s. is minimax redundancy of i.i.d. source, $\frac{1}{2}\ln n$ nits per base alphabet symbol, achieved by KT estimator.

- My model (l.h.s.) improves upon this in two significant ways:

  (1) Each symbol j that appears only finitely often, induces finite bounded code length $\frac{1}{2}\ln n_j + 1$.

  (2) Symbols k that do not appear in $x_{1:n}$ induce zero code length.

- Only symbols appearing with non-vanishing frequency $n_i/n \not\to 0$ have asymptotic redundancy $\frac{1}{2}\ln n$.

Adaptive Variable β^*_t

Problem: $\beta^* = m/2\ln\frac{n}{m}$ depends on m and n $\Rightarrow$ S^{β^*} not online.

Solution: Replace $n \rightsquigarrow t$ and $m \rightsquigarrow m_t$, both known at time t and converging to n and m respectively, and regularize $t \rightsquigarrow t+1$:

Adaptive Variable $\beta^*_t \;:=\; \frac{m_t}{2\ln\frac{t+1}{m_t}}$

- Compact representation of $S(x_{1:n})$ is no longer possible.
- Resulting process no longer exchangeable, but still approximately.
- Still same redundancy bound but somewhat worse constants.
- Bound also holds for DirM with corresponding adaptive parameters.

Algorithms & Computation Time

- S and DirM require $O(1)$ time and $O(D)$ space for computing $P(x_{t+1}|x_{1:t})$ and for updating the relevant parameters like n_i, m_t, β^*_t.
- Space can be reduced to $O(m)$ by hashing.
- $P(x_{t+1}|x_{1:t})$ is sufficient for e.g. model selection.
- Data compression via arithmetic coding requires $P(X_{t+1} < x_{t+1}|x_{1:t})$, which naively requires $O(D)$ time per t.
- Improvement to $O(\log D)$: Maintain a binary tree of depth $\lceil \log_2 D \rceil$ with counts $n_1, n_2, ..., n_D$ and unnormalized weights at the leafs in this order. Inner nodes store the sum of their two children.
- Time can be reduced to $O(\log m)$ and space to $O(m)$ by maintaining a self-balancing binary tree of only the non-zero counts.
- Bayes-optimal decisions can be computed/updated in $O(1)$ time.
- Lazy update of logarithm in β^*_t possible.

Experiments

Online Estimators:

- $S^{\vec{\beta}^*}$: My model with optimal variable $\beta_t^* = m_t/2\ln\frac{t+1}{m_t}$ [Hut13]
- $\text{KT}_{\mathcal{X}}$: KT-estimator with base alphabet $\mathcal{X}$
- Perks: DirM with $\alpha_i = 1/D$
- SSDC: KT-estimator w.r.t. $\mathcal{A}_t$ and escape probability ${}^1/_{t+1}$ [VH12]
- $\vec{\text{DirM}}^*$: Dirichlet-multinomial optimal variable $\alpha_i^{t*} = \beta_t^*/D$
- SAW-Bayes: Bayesian sub-alphabet weighting [TSW93]

Offline Estimators:

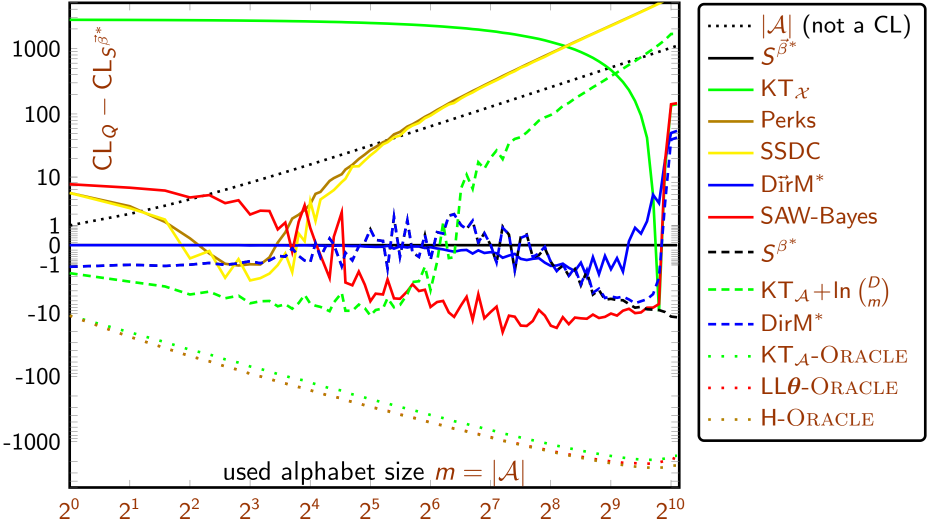
- S^{β^*}: My model with optimal constant $\beta^* = m/2\ln\frac{n}{m}$
- $\text{KT}_{\mathcal{A}}+\ln\binom{D}{m}$: KT-estimator w.r.t. $\mathcal{A}$ plus CL of unsorted $\mathcal{A}$
- DirM^*: Dirichlet-multinomial optimal constant $\alpha_i^* = \beta^*/D$

Oracle Estimators:

- $\text{KT}_{\mathcal{A}}$-ORACLE: KT-estimator with used alphabet $\mathcal{A}$
- LL$\boldsymbol{\theta}$-ORACLE: log-likelihood of the sampling distr. $\ln 1/P_{iid}^{\boldsymbol{\theta}}$
- H-ORACLE: Empirical entropy $nH(\frac{\mathbf{n}}{n})$

Data: Uniform $\theta_{1:m} \sim U(\Delta)$, $\theta_{m+1:D} = 0$; Zipf $\theta_i = i^{-\gamma}$; Real Calgary

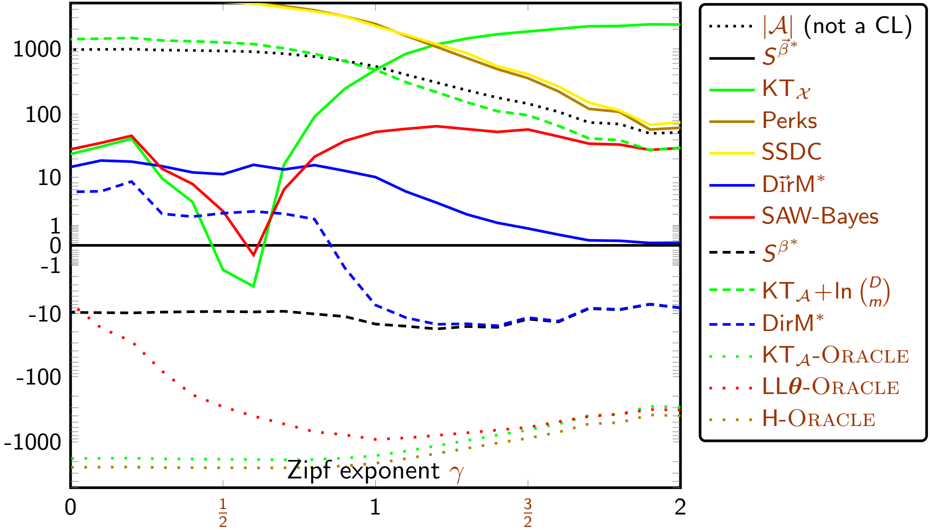
Artificial Uniform Data



$\theta_{1:m}$ ~Uniform, $\theta_{m+1:D} = 0$, $n = 1024$, $D = 10\,000$, varying m.

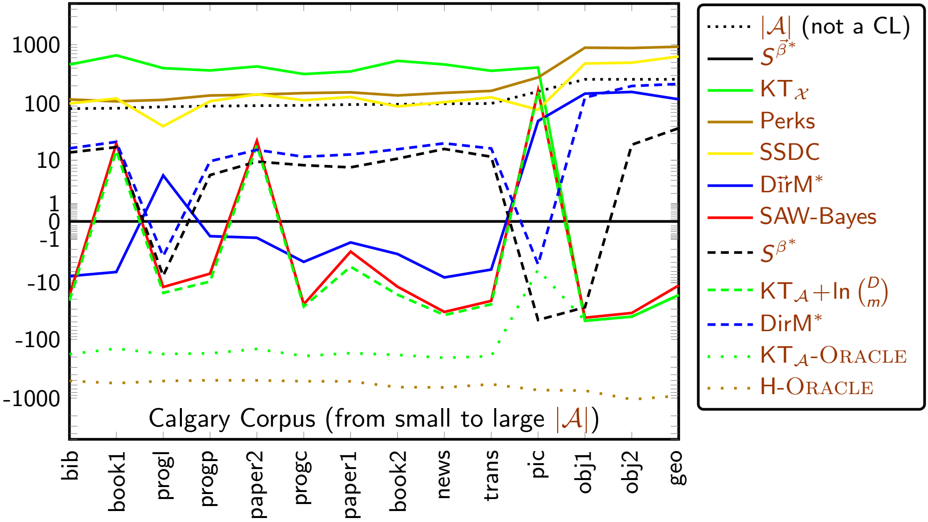
The online/offline/oracle estimators have solid/dashed/dotted lines.

Artificial Zip-Distributed Data ($\theta_i = i^{-\gamma}$)



$\theta_i = i^{-\gamma}$, $n = 1024$, $D = 10\,000$, varying Zipf exponent $0 \le \gamma \le 2$.
The online/offline/oracle estimators have solid/dashed/dotted lines.

Real Data: Calgary Corpus



Real data: 14 files with $21\,504 \leq n \leq 768\,771$ byte alphabet ($D = 256$).
The online/offline/oracle estimators have solid/dashed/dotted lines.

Discussion of Experiments

- The results generally confirm the theory with few/small surprises.
- $\mathrm{D\vec{ir}M}^*$ and $S^{\vec{\beta}^*}$ are very close for most m.
- The offline estimators mostly coincide with their online versions.
- Off-line $\mathrm{KT}_{\mathcal{A}}+\ln\binom{D}{m}$ significantly improves upon $\mathrm{KT}_{\mathcal{X}}$ for small m, but breaks down for medium and large m,
- Observations are mostly consistent across uniform, Zipf, and real data. But for Zipf data, SAW-Bayes and $\mathrm{KT}_{\mathcal{A}}+\ln\binom{D}{m}$ seem to be worse & relative performance of many estimators on b&w fax pic is reversed.
- Oracles possess significant extra knowledge:
  $\mathrm{KT}_{\mathcal{A}}$-ORACLE the used alphabet $\mathcal{A}$,
  and LL$\boldsymbol{\theta}$-ORACLE and H-ORACLE even the counts $\mathbf{n}$.
  The plots show the magnitude of this extra knowledge.

Summary of Experiments

Results are similar for other (n, D, m) and (n, D, γ) combinations but code length differences can be more or less pronounced but are seldom reversed.

In short,

- $\text{KT}_{\mathcal{X}}$ performs very poorly unless $m \approx D$;
- Perks and SSDC perform poorly unless $m \lesssim \ln n$;
- $\text{KT}_{\mathcal{A}} + \ln \binom{D}{m}$, DirM*, S^{β^*} are not online;
- LL$\boldsymbol{\theta}$-ORACLE, H-ORACLE, $\text{KT}_{\mathcal{A}}$-ORACLE are not realizable;
- SAW-Bayes performs well but is extremely slow (factor $\tilde{O}(m)$);
- which leaves $\vec{\text{Dir}}\text{M}^*$ and $S^{\vec{\beta}^*}$ as winners.
- Winners perform very similar unless m gets very close to $\min\{n, D\}$ in which case $S^{\vec{\beta}^*}$ wins.

Conclusion

- New model S related to the Dirichlet-multinomial distribution.
- Tight bounds for codelength $\widehat{=}$ redundancy $\widehat{=}$ likelihood $\widehat{=}$ perplexity.
- Data-(n_i)-dependent (rather then expected or worst-case) bounds.
- Optimal choice of β different from traditional recommendations.
- Constant offline β^* and variable online $\vec{\beta}^*$.
- Zero CL for unused symbols,
  finite CL for symbols occurring only finitely often,
  still optimal minimax redundancy $\frac{1}{2}\ln n$ in general.
- Bounds independent of size of $\mathcal{X}$ and even hold for continuous $\mathcal{X}$.
- Experimentally, $S^{\vec{\beta}^*}$'s performance is superb.
- $S^{\vec{\beta}^*}$ is simple, online, fast, i.i.d. estimator.
- Useful sub-component in non-i.i.d. online algorithms [VNHB12, OHSS12, Mah12]
- Redundancy bounds are of theoretical interest.

Some References

M. Hutter.
Sparse adaptive Dirichlet-multinomial-like processes.
Journal of Machine Learning Research, W&CP: COLT, 2013.

M. Mahoney.
Data Compression Explained, 2012.

A. O'Neill, M. Hutter, W. Shao, and P. Sunehag.
Adaptive context tree weighting.
In *Proc. Data Compression Conference (DCC 2012)*, pages 317–326.

T. J. Tjalkens, Y. M. Shtarkov, and F. M. J. Willems.
Sequential weighting algorithms for multi-alphabet sources, 1993.

J. Veness and M. Hutter.
Sparse sequential dirichlet coding.
Technical Report arXiv:1206.3618, UoA and ANU, 2012.

J. Veness, K. S. Ng, M. Hutter, and M. Bowling.
Context tree switching.
In *Proc. Data Compression Conference (DCC 2012)*, pages 327–336.