Hyper-parameter Estimation for the Dirichlet Prior

Wray Buntine
Machine Learning Research Group, NICTA
Canberra, ACT
wray.buntine@nicta.com.au

September 28, 2012

Abstract

A common scenario in topic modelling is where one has samples from a Dirichlet, but the Dirichlet parameters themselves are unknown. This technical note describes theory and algorithms for estimation for this situation using Poisson Dirichlet processes. Some of the algorithms are coded up as `samplea()` and `sampleb()` in the `libstb` library, documented in "psample.h".

1 Basic Theory

The background theory for this note is given in [1].

1.1 The Model

Topics for each document come from a Dirichlet with parameter $\vec{\alpha} = b\vec{m}$. The mean parameter $\vec{m}$ itself comes from a symmetric Dirichlet with concentration parameter b_0. So we have:

$$\begin{aligned} \vec{n}_i &\sim \text{multinomial}_K\left(\vec{\mu}_i, N_i\right) && \forall i \\ \vec{\mu}_i &\sim \text{Dirichlet}_K(b\vec{m}) && \forall i \\ \vec{m} &\sim \text{Dirichlet}_K\left(\frac{b_0}{K}\vec{1}\right) \\ b &\sim \text{Gamma}(\sigma_b, s_b) \end{aligned}$$

Here s_b is the scale and σ_b is the shape of the gamma prior on concentration parameter b. Using Dirichlet processes for the first Dirichlets on $\vec{\mu}_i$ and integrating out both $\vec{\mu}_i$ and $\vec{m}$ yields a posterior

$$p(b, \forall i : \vec{n}_i, \vec{t}_i | b_0, K, \forall i : N_i) \; = \tag{1}$$

$$e^{-b/s_b} b^{\sigma_b - 1} \prod_i \left(\frac{b^{t_{i,\cdot}} \Gamma(b)}{\Gamma(N_i + b)} \prod_k S^{n_{i,k}}_{t_{i,k},0} \right) \frac{\Gamma(b_0)}{\Gamma(b_0/K)^K} \frac{\prod_k \Gamma(t_{\cdot,k} + b_0/K)}{\Gamma(t_{\cdot,\cdot} + b_0)}$$

Our problem is we need to sample the matrix of $t_{i,k}$ values. These satisfy $t_{i,l} \leq n_{i,k}$ and $t_{i,l} = 0$ if and only if $n_{i,k} = 0$.

Alternatively, one can use a Poisson-Dirichlet process on the $\vec{\mu}_i$, for $0 < a < 1$ and $b > -a$,

$$\vec{\mu}_i \sim \text{PDP}(a, b, \vec{m}) \qquad \forall i$$

and the term in the large brackets in Equation (1) becomes

$$\left(a^{t_{i,\cdot}} \frac{\Gamma(t_{i,\cdot} + b/a)\Gamma(b)}{\Gamma(b/a)\Gamma(N_i + b)} \prod_k S^{n_{i,k}}_{t_{i,k},a} \right)$$

1.2 Strategy

For a fixed a, we begin by doing a crude estimate for the $\vec{t}_i$, and then jointly sample $\vec{t}_i$ and b starting from the estimate. From this, estimates can be got of b and

$$\widehat{\vec{m}} = \frac{t_{\cdot,k} + b_0/K}{t_{\cdot,\cdot} + b_0}$$

1.3 Sampling b

Introduce the auxiliary variable $q_i \sim \text{Beta}(b, N_i)$, then the above terms in b get changed to

$$p(b, \forall i : q_i | a = 0, b_0, K, \forall i : \vec{n}_i, N_i) \propto \quad e^{-b/s_b} b^{t_{\cdot,\cdot} + \sigma_b - 1} \prod_i q_i^{b-1} (1 - q_i)^{N_i - 1}$$

$$p(b, \forall i : q_i | a > 0, b_0, K, \forall i : \vec{n}_i, N_i) \propto \quad e^{-b/s_b} b^{\sigma_b - 1} \prod_i \frac{\Gamma(t_{i,\cdot} + b/a)}{\Gamma(b/a)} q_i^{b-1} (1 - q_i)^{N_i - 1}$$

1.3.1 Sampling b when $a = 0$

Thus for $a = 0$ for posterior sampling we have the following probability equations

$$\begin{aligned} b | \vec{q} &\sim \text{Gamma}\left(\sigma_b + t_{\cdot,\cdot}, 1/s_b + \sum_i \log 1/q_i \right) , \\ q_i | b &\sim \text{Beta}(b, N_i) , \end{aligned}$$

according to which we can do Gibbs sampling. Note that

$$\mathbb{E}_{q_i|b} \left[log 1/q_i \right] = \psi(N_i + b) - \psi(b) ,$$

where $\psi(\cdot)$ is the digamma function. so this could be approximated as

$$b \sim \text{Gamma}\left(\sigma_b + t_{\cdot,\cdot}, 1/s_b + \sum_i \left(\psi(N_i + b) - \psi(b) \right) \right) ,$$

or

$$b \approx \frac{\sigma_b + t_{\cdot,\cdot}}{1/s_b + \sum_i \left(\psi(N_i + b) - \psi(b) \right)} .$$

1.3.2 Sampling b when $a > 0$

When $a > 0$, the condition posterior on b given $\vec{q}$ is easily seen to be log concave when $\sigma_b \geq 1$. Note that assuming $\sigma_b = 1$ makes the power of b disappear.

For this task one can use slice sampling or adaptive rejection sampling, both seem to work well. Slice sampling is easier to implement but needs a reasonable starting point (starting at a very low probability point will mean the slice sampler takes a long while to warm up). Thus for slice sampling, we also need to find an approximate MAP point. We do this by running a few iterations of a fixed point maximiser.

The maximum can be found by differentiation and is given by

$$0 \;=\; \frac{1}{a}\sum_i \left(\psi(t_{i,\cdot} + b/a) - \psi(b/a)\right) - 1/s_b + \frac{(\sigma_b - 1)}{b} + \sum_i \log 1/q_i \;.$$

Letting $Q = 1/s_b + \sum_i \log 1/q_i$. Note the maximum can be found fairly quickly via a fixed point

$$b` \;\leftarrow\; a\psi^{-1}\left(\frac{1}{I}\sum_i \psi(t_{i,\cdot} + b/a) - \frac{aQ}{I} + \frac{a(\sigma_b - 1)}{bI}\right)$$

because $\left|\frac{\mathrm{d}b'}{\mathrm{d}b}\right| < 1$ when some $t_{i,\cdot} > 0$. The inverse of the digamma function is provided by Minka, Appendix C in [2].

1.4 Sampling $t_{i,k}$

Considering just the terms in $t_{i,k}$ we have the marginal posterior

$$p\left(t_{i,k}|b, b_0, K, \forall i : \vec{n}_i, \vec{t}_i\right) \;\propto\; \begin{cases} b^{t_{i,k}} S^{n_{i,k}}_{t_{i,k},0} \frac{\Gamma(t_{\cdot,k} + b_0/K)}{\Gamma(t_{\cdot,\cdot} + b_0)} & \text{if } a \equiv 0 \\ (b|a)_{t_{i,\cdot}} S^{n_{i,k}}_{t_{i,k},a} \frac{\Gamma(t_{\cdot,k} + b_0/K)}{\Gamma(t_{\cdot,\cdot} + b_0)} & \text{if } a > 0 \end{cases}$$

Note when $n_{i,k} \leq 1$, then $t_{i,k} = n_{i,k}$ and no sampling is needed. So only consider those cases where $n_{i,k} > 1$.

Making an approximation for $\frac{\Gamma(t_{\cdot,k} + b_0/K)}{\Gamma(t_{\cdot,\cdot} + b_0)}$ of $\psi_k^{t_{i,k}}$ we get an initialisation where $t_{i,k}$ can be independently initialised according to

$$t_{i,k} \;\leftarrow\; \min\left(n_{i,k}, \mathrm{argmax}_{t_{i,k}} \left((b + a\, t_{i,\cdot})\psi_k\right)^{t_{i,k}} S^{n_{i,k}}_{t_{i,k},a}\right) \;,$$

before standard Gibb sampling begins.

For sampling $\vec{t}$ we will use Chen *et al.*'s table indicators approach. For this,

$$p\left(t_{i,k}, \vec{r}_i|b, b_0, K, \forall i : \vec{n}_i, \vec{t}_i\right) \;\propto\; \binom{n_{i,k}}{t_{i,k}}^{-1} p\left(t_{i,k}|b, b_0, K, \forall i : \vec{n}_i, \vec{t}_i\right)$$

Sampling proceeds as follows:

1. Remove the old indicator with probability $t_{i,k}/n_{i,k}$.

2. If $t_{i,k} \equiv 0$, necessarily add a new indicator. Otherwise, add a new indicator depending on

$$(t_{i,k}+1)(b+a\,t_{i,\cdot})S^{n_{i,k}}_{t_{i,k}+1,a}\frac{t_{\cdot,k}+b_0/K}{t_{\cdot,\cdot}+b_0} \qquad \text{versus} \qquad (n_{i,k}-t_{i,k}+1)S^{n_{i,k}}_{t_{i,k},a}$$

1.5 Sampling a

When $a > 0$ it too can be sampled.

$$p(a|a>0,b,b_0,K,\forall i:\vec{n}_i,N_i) \propto \prod_i \left(a^{t_{i,\cdot}} \frac{\Gamma(t_{i,\cdot}+b/a)}{\Gamma(b/a)} \prod_k S^{n_{i,k}}_{t_{i,k},a} \right) \tag{2}$$

This is known to be log concave[1], moreover the posterior from experience is known to be moderately flat. The only problem is that computing the posterior for different discounts a requires recomputing the Stirling numbers. To make this more efficient, the maximum used $n_{i,k}$ and $t_{i,k}$ is first computed.

Alternatively, one can expand the term $S^n_{t,a}$ into its parts. Recall from [1] is the normaliser for the Chinese Restaurant Distribution (CRD) for all partitions of n of size t. So we sample such a partition $(n_1,...,n_t)$ according to the CRD and use its probability in place of $S^n_{t,a}$. That is, let $I_n = (n_1,...,n_t)$ denote a partition of n for $t = 1,...,n$ for CRD with discount $a > 0$.

$$\begin{aligned} p(I_n \mid CRD,a,b,n) &= \frac{(b|a)_t}{(b)_n}\prod_{k=1}^{t}(1-a)_{n_k-1} \\ p(t \mid CRD,a,b,n) &= \frac{(b|a)_t}{(b)_n}S^n_{t,a} \\ p(I_n \mid CRD,a,b,n,t) &= \frac{1}{S^n_{t,a}}\prod_{k=1}^{t}(1-a)_{n_k-1} \end{aligned}$$

Thus for each term $S^{n_{i,k}}_{t_{i,k},a}$ in Equation (2), we sample a partition of $n_{i,k}$ of size $t_{i,k}$ using the CRD giving $\vec{m}_{i,k}$, and then consider the derived marginal for a

$$p\left(a|a>0,b,b_0,K,\forall i:\vec{n}_i,N_i;\forall_{i,k}\vec{m}_{i,k}\right) \propto \prod_i \left(a^{t_{i,\cdot}} \frac{\Gamma(t_{i,\cdot}+b/a)}{\Gamma(b/a)} \prod_k \prod_{l=1}^{t_{i,k}} (1-a)_{m_{i,k,l}-1} \right)$$

This yields an expression that can be sampled for a without recomputing the Stirling number tables. It too is log concave using the same proof as for Equation (2).

Sampling a partition $I_n = (m_1,...,m_t)$ of n of size t can done using a variation of the CRD probabilities above that can be verified using properties of the

[1] Although, this is quite complicated to prove and is done elsewhere.

generalised Stirling number (see Theorem 17(ii) in [1]). The partition of n of size t, I_n, can be sampled recursively using a partition of $n - m_t$ of size $t - 1$ denoted I_{n-1}. So

$$p(m_t \mid CRD, a, b, n, t) \;=\; \frac{S^{n-m_t}_{t-1,a}}{S^n_{t,a}} \binom{n-1}{m_t - 1} (1-a)_{m_t - 1}$$

So sample m_t according to this formula and then sample a partition of $n - m_t$ of size $t - 1$. This requires using the table of Stirling numbers already existing for discount a.

2 Algorithms

As initialisation, we need to compute a table of Stirling numbers.
The algorithm keeps the following internal counts which do not need to be kept between calls but are used in the one call. Note the matrix $\vec{t}$ could be highly compressed, because its counts could be assumed to be one-two bytes only and only values needed are where $n_{d,k} > 1$.

T; ▷ t totals
$\vec{Tk}$; ▷ t totals over docs, indexed by k
$\vec{Td}$; ▷ t totals over topics, indexed by d
$\vec{t}$; ▷ t, indexed by d, k

Note that $\vec{Td}$ is only used when $a > 0$.

Note that BETA$(\cdot,\cdot)$, GAUSSIAN$(\cdot,\cdot)$, and GAMMA$(\cdot)$ functions are samplers according to the distributions. The second argumenbt to the Gaussian is the standard deviation, and the argument to the gamma is the shape parameter (with the scale left as 1). The U() function returns a uniform random number on the unit interval. The ST.V() function is the Stirling number ratio V from [1].

2.1 Approximating MAP for $\vec{t}$

Initialisation goes as follows. This is called every time $\vec{\alpha}$ is sampled/optimised.

function INITIALISE$(b, \forall_i \vec{n}_i)$
  $T = 0$, $\vec{Tk} = \vec{0}$, $\vec{Td} = \vec{0}$
  for documents $i = 1..I$ **do** ▷ First pass assigns default initialisation
    for topics $k = 1..K$ **do**
      if $n_{i,k} > 0$ **then**
        $t_{i,k} = 1$, $Td_i{+}{+}$, $Tk_k{+}{+}$, $T{+}{+}$
      else
        $t_{i,k} = 0$
      end if
    end for
  end for
  local $\vec{\psi} = \text{Norm}(\vec{Tk})$

```
    for documents i = 1..I do                    ▷ Second pass does approximation
        for topics k = 1..K do
            local fact = (b + a (T d_i − t_{i,k}/2)) ψ_k
            if n_{i,k} > 2 then
                local t' = 1
                while fact ∗ ST.V(n_{i,k}, t' + 1, 0) > 1 and t' < n_{i,k} do
                    t'++
                end while
                if t' > 1 then
                    t_{i,k} = t'
                    t'−−
                    Tk_k += t', T += t'
                end if
            end if
        end for
    end for
end function
```

2.2 Sampling b

Sampling b when $a = 0$ goes as follows. Note it uses the existing value of b too.

```
    function SAMPLE-B1(b, s_b, σ_b ≥ 1, T, N⃗, a ≡ 0)
        local logQ = 1/s_b
        for documents i = 1..I do
            local q ∼ BETA(b, N_i),          ▷ Careful, large N_i canhave q underflow
            logQ −=  log(q)
        end for
        T += σ_b.
        if T > 400 then
            b ∼ GAUSSIAN (T, √T)             ▷ use if Gamma samplers broken for large T
        else
            b ∼ GAMMA (T)
        end if
        b ← b/logQ
        return b
    end function
```

Sampling b when $a > 0$ is more complicated. We need a maximiser as well.

```
    function ψ^{-1}(x)
        locale guess
        if x < −2.22 then
            guess ← −1/(x − ψ(1.0))
        else
            guess ← exp(x) + 0.5
        end if
```

```
    for i = 0...4 do
        guess-= (ψ(guess) − x)/ψ1(guess)
    end for
    return guess
end function
function PROB-B2(b, σb ≥ 1, Q, T⃗, a > 0)
    local logprob ← −b Q + (σb − 1) log b
    for documents i = 1..I do
        logprob+= Γ(Ti + b/a) − Γ(b/a)
    end for
    return logprob
end function
function MAP-B2(b, σb ≥ 1, Q, T⃗, a > 0)
    local psiinv = 0
    for documents i = 1..I do
        psiinv+= ψ(Ti + b/a)
    end for
    psiinv = psiinv/I + a(σb−1)/(bI) − aQ/I
    return a ψ^{-1}(psiinv)
end function
function SAMPLE-B2(b, sb, σb ≥ 1, T⃗, N⃗, a > 0)
    local logQ = 1/sb
    for documents i = 1..I do
        local q ∼ BETA(b, Ni), logQ -= log(q)
    end for
    b ← MAP-B2(b, σb, Q, T⃗, a)
    b ← SLICESAMPLE (b, PROB-B2(b, σb, Q, T⃗, a))
    return b
end function
```

2.3 Main

The main algorithm.

```
INITIALISE (b, ∀i n⃗i)
for documents i = 1..I do
    for topics k = 1..K do
        if ni,k > 1 then
            local t′ = tn,k
            Tdi -= t′, Tkk -= t′, T -= t′
            for ni,k/2 times do
                if U() < t′/ni,k then
                    t′--
                end if
                if t′ ≡ 0 then
                    t′ ← 1
```

else
local $rp = \frac{t'+1}{n_{i,k}-t'+1}\,(b+a(Td_i+t'))\,\mathrm{ST.V}\,(n_{i,k}, t'+1, 0)\,\frac{Tk_k+t'+b_0/K}{T+t'+b_0}$
if U() $< rp/(1+rp)$ **then**
t'++
end if
end if
end for
$t_{i,k} \leftarrow t'$
Td_i += t', Tk_k += t', T += t'
end if
end for
end for
$\vec{m} \leftarrow \frac{1}{T+b_0}\left(\vec{Tk} + \frac{b_0}{K}\right)$
$b \leftarrow$ SAMPLE-B$(b, T, \vec{N})$
return b and $\vec{m}$

References

[1] Buntine, W., Hutter, M.: A Bayesian view of the Poisson-Dirichlet process. Technical Report arXiv:1007.0296v2, *ArXiv*, Cornell, February, 2012.

[2] Minka, T.P.: Estimating a Dirichlet distribution. Technical report, MIT (2000). Revised 2012. `http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/`