From high-level inference algorithms to efficient code

Rajan Walia[1] Jacques Carette[2] Praveen Narayanan[1]
Chung-chieh Shan[1] Sam Tobin-Hochstadt[1]

[1]Indiana University [2]McMaster University

IFIP WG 2.11 2019-04-29

From high-level inference algorithms to efficient code

Rajan Walia[1] Jacques Carette[2] Praveen Narayanan[1]
Chung-chieh Shan[1] Sam Tobin-Hochstadt[1]

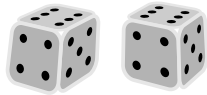
[1]Indiana University [2]McMaster University

IFIP WG 2.11 2019-04-29

"Algorithm"

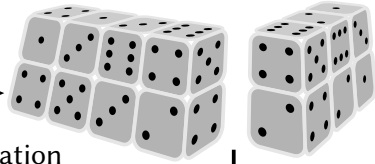
- arithmetic
- sorting
- differentiation
- sampling

scalar simplification

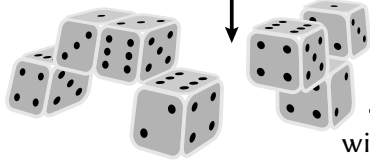


code gen

array simplification



histogram



code gen
with loop optimization

Array simplification

Goals

- Make fewer random choices
- Avoid weighting samples

Capabilities

- Eliminate latent variables
  (aka Rao-Blackwellization, collapse, marginalization, integrating out)
- Recognize primitive densities
  (in particular, conjugacy relationships between prior and likelihood)

Techniques

- Random arrays denote high- and arbitrary-dimensional integrals
- The unproduct operation:
  if unproduct$(e, \vec{x}, i)$ returns the expression pair $(e', g(\vec{x}[i]))$ then $e = e' \cdot \prod_i g(\vec{x}[i])$

Before histogram

Gaussian model

$$\sum_j (y[j] - x)^2$$

$$= \sum_j y[j]^2$$

$$- 2x \sum_j y[j]$$

$$+ x^2 \sum_j 1$$

Gaussian mixture model

$$\sum_j (y[j] - x[\ell[j]])^2 = \sum_k \sum_j \begin{cases} (y[j] - x[k])^2 & k = \ell[j] \\ 0 & \text{otherwise} \end{cases}$$

$$= \sum_k \sum_j \begin{cases} y[j]^2 & k = \ell[j] \\ 0 & \text{otherwise} \end{cases}$$

$$- \sum_k 2x[k] \sum_j \begin{cases} y[j] & k = \ell[j] \\ 0 & \text{otherwise} \end{cases}$$

$$+ \sum_k x[k]^2 \sum_j \begin{cases} 1 & k = \ell[j] \\ 0 & \text{otherwise} \end{cases}$$

After histogram

$$\sum_{j=0}^{n-1} y[j]^2 \quad = \quad \begin{array}{l}\text{let } \mathit{hist} = \mathrm{Hist}_{j=0}^{n-1}\left(\mathrm{Add}(y[j]^2)\right) \\ \text{in } \mathit{hist}\end{array}$$

$$\sum_{j=0}^{n-1} \begin{cases} y[j]^2 & k = \ell[j] \\ 0 & \text{otherwise} \end{cases} \quad = \quad \begin{array}{l}\text{let } \mathit{hist} = \mathrm{Hist}_{j=0}^{n-1}\left(\mathrm{Idx}_k^m(\ell[j], \mathrm{Add}(y[j]^2))\right) \\ \text{in } \mathit{hist}[k]\end{array}$$

After histogram

$$\sum_{j=0}^{n-1} y[j]^2 \quad = \quad \begin{array}{l}\text{let } \mathit{hist} = \mathrm{Hist}_{j=0}^{n-1}\left(\mathrm{Add}(y[j]^2)\right) \\ \text{in } \mathit{hist}\end{array}$$

$$\sum_{j=0}^{n-1} \begin{cases} y[j]^2 & k = \ell[j] \\ 0 & \text{otherwise} \end{cases} \quad = \quad \begin{array}{l}\text{let } \mathit{hist} = \mathrm{Hist}_{j=0}^{n-1}\left(\mathrm{Idx}_k^m(\ell[j], \mathrm{Add}(y[j]^2))\right) \\ \text{in } \mathit{hist}[k]\end{array}$$

$$\frac{\begin{array}{c}[j:\mathbb{N}] \\ \vdots \\ e:\mathbb{R}\end{array}}{\mathrm{Add}(e) \triangleright_j \mathbb{R}} \qquad \frac{\begin{array}{cc}[j:\mathbb{N}] & [k:\mathbb{N}] \\ \vdots & \vdots \\ b:\mathbb{N} \quad e:\mathbb{N} & r \triangleright_j T\end{array}}{\mathrm{Idx}_k^b(e, r) \triangleright_j \mathbb{A}\, T} \qquad \frac{\begin{array}{c}[j:\mathbb{N}] \\ \vdots \\ e:\mathbb{B} \quad r_1 \triangleright_j T_1 \quad r_2 \triangleright_j T_2\end{array}}{\mathrm{Split}(e, r_1, r_2) \triangleright_j T_1 \times T_2}$$

$$\frac{r_1 \triangleright_j T_1 \quad r_2 \triangleright_j T_2}{\mathrm{Fanout}(r_1, r_2) \triangleright_j T_1 \times T_2} \qquad \frac{}{\mathrm{Nop} \triangleright_j \mathbb{1}} \qquad \frac{a:\mathbb{N} \quad b:\mathbb{N} \quad r \triangleright_j T}{\mathrm{Hist}_{j=a}^b(r) : T}$$

After histogram

$$\sum_{j=0}^{n-1} y[j]^2 \quad = \quad \begin{array}{l}\text{let } hist = \text{Hist}_{j=0}^{n-1}\left(\text{Add}(y[j]^2)\right) \\ \text{in } hist\end{array}$$

$$\sum_{j=0}^{n-1} \begin{cases} y[j]^2 & k = \ell[j] \\ 0 & \text{otherwise} \end{cases} \quad = \quad \begin{array}{l}\text{let } hist = \text{Hist}_{j=0}^{n-1}\left(\text{Idx}_k^m(\ell[j], \text{Add}(y[j]^2))\right) \\ \text{in } hist[k]\end{array}$$

$$\sum_{j=0}^{n-1} \begin{cases} y[j]^2 & k = \ell[j] \wedge k' = \ell'[j] \\ 0 & \text{otherwise} \end{cases} \quad = \quad \begin{array}{l}\text{let } hist = \text{Hist}_{j=0}^{n-1}\left(\text{Idx}_k^m(\ell[j], \text{Idx}_{k'}^{m'}(\ell'[j], \text{Add}(y[j]^2)))\right) \\ \text{in } hist[k][k']\end{array}$$

$$\sum_{j=0}^{n-1} \begin{cases} y[j]^2 & k = \begin{cases} \ell[j] & j \neq j^* \\ \ell^* & \text{otherwise} \end{cases} \\ 0 & \text{otherwise} \end{cases} \quad = \quad \begin{array}{l}\text{let } (hist_1, (hist_2, ())) \\ \qquad = \text{Hist}_{j=0}^{n-1}\left(\text{Split}(j \neq j^*, \text{Idx}_k^m(\ell[j], \text{Add}(y[j]^2)),\right. \\ \qquad\qquad \left.\text{Fanout}(\text{Add}(y[j]^2), \text{Nop})\right) \\ \text{in } hist_1[k] + \begin{cases} hist_2 & k = \ell^* \\ 0 & \text{otherwise} \end{cases}\end{array}$$

After histogram

$$\frac{\begin{array}{c}[j:\mathbb{N}]\\ \vdots\\ e:\mathbb{R}\end{array}}{\mathrm{Add}(e) \triangleright_j \mathbb{R}} \qquad \frac{\begin{array}{cc}[j:\mathbb{N}] & [k:\mathbb{N}]\\ \vdots & \vdots\\ b:\mathbb{N} \quad e:\mathbb{N} & r \triangleright_j T\end{array}}{\mathrm{Idx}_k^b(e, r) \triangleright_j \mathbb{A}\, T} \qquad \frac{\begin{array}{c}[j:\mathbb{N}]\\ \vdots\\ e:\mathbb{B} \quad r_1 \triangleright_j T_1 \quad r_2 \triangleright_j T_2\end{array}}{\mathrm{Split}(e, r_1, r_2) \triangleright_j T_1 \times T_2}$$

$$\frac{r_1 \triangleright_j T_1 \quad r_2 \triangleright_j T_2}{\mathrm{Fanout}(r_1, r_2) \triangleright_j T_1 \times T_2} \qquad \frac{}{\mathrm{Nop} \triangleright_j \mathbb{1}} \qquad \frac{a:\mathbb{N} \quad b:\mathbb{N} \quad r \triangleright_j T}{\mathrm{Hist}_{j=a}^b(r) : T}$$

$$\sum_{j=0}^{n-1} \begin{cases} y[j]^2 & k = \ell[j] \wedge k' = \ell'[j] \\ 0 & \text{otherwise} \end{cases} = \begin{array}{l} \text{let } \mathit{hist} = \mathrm{Hist}_{j=0}^{n-1}\left(\mathrm{Idx}_k^m(\ell[j], \mathrm{Idx}_{k'}^{m'}(\ell'[j], \mathrm{Add}(y[j]^2)))\right) \\ \text{in } \mathit{hist}[k][k'] \end{array}$$

$$\sum_{j=0}^{n-1} \begin{cases} y[j]^2 & k = \begin{cases} \ell[j] & j \neq j^* \\ \ell^* & \text{otherwise} \end{cases} \\ 0 & \text{otherwise} \end{cases} = \begin{array}{l} \text{let } (\mathit{hist}_1, (\mathit{hist}_2, ())) \\ \quad = \mathrm{Hist}_{j=0}^{n-1}\left(\mathrm{Split}(j \neq j^*, \mathrm{Idx}_k^m(\ell[j], \mathrm{Add}(y[j]^2)),\right. \\ \qquad\qquad \left.\mathrm{Fanout}(\mathrm{Add}(y[j]^2), \mathrm{Nop})\right) \\ \text{in } \mathit{hist}_1[k] + \begin{cases} \mathit{hist}_2 & k = \ell^* \\ 0 & \text{otherwise} \end{cases} \end{array}$$

Ablation study

Table: Run time in seconds (mean over 1000 trials and standard error) of one sweep of Gibbs sampling with $m = 50$ and $n = 10000$. Slowdown is compared to full optimization.

Optimizations	Time in seconds	Slowdown
No optimizations	471.441 ± 0.5973	1848 ×
No histogram	460.596 ± 0.1514	1805 ×
No LICM and loop fusion	328.736 ± 0.1019	1289 ×
No loop fusion	0.471 ± 0.0032	1.8×
No run-time specialization	2.422 ± 0.0054	9.5×
Full optimization	0.255 ± 0.0005	—

Histogram implementation

$$\text{histogram}\big(C\big[\begin{cases} e_1 & \\ e_2 & \\ \text{otherwise} & e \end{cases}\big], j\big) \longrightarrow \Big(\text{Fanout}(m_1, m_2), \lambda(hist_1, hist_2).\ \begin{cases} f_1(hist_1) & \\ f_2(hist_2) & \\ \text{otherwise} & e \end{cases}\Big)$$

where $(m_k, f_k) = \text{histogram}(C[e_k], j)$ and e does not depend on j

$$\text{histogram}\big(C\big[\begin{cases} e_1 & \\ e_2 & \\ \text{otherwise} & e \end{cases}\big], j\big) \longrightarrow \big(\text{Split}(e, m_1, m_2), \lambda(hist_1, hist_2).\ f_1(hist_1) + f_2(hist_2)\big)$$

where $(m_k, f_k) = \text{histogram}(C[e_k], j)$

$$\text{histogram}\big(\begin{cases} a & i = e \\ 0 & \text{otherwise} \end{cases}, \quad j\big) \longrightarrow \Big(\text{Idx}_i^m(e, r), \lambda hist.\ \begin{cases} f(hist[i]) & i \in \{0, \ldots, m-1\} \\ 0 & \text{otherwise} \end{cases}\Big)$$

where $(r, f) = \text{histogram}(a, j)$, i is a loop-bound variable that does not depend on j, and the context entails that $i \in \{0, \ldots, m-1\}$ or $e \in \{0, \ldots, m-1\}$

$$\text{histogram}(0, \quad j) \longrightarrow (\text{Nop}, \lambda hist.\ 0)$$

$$\text{histogram}(e, \quad j) \longrightarrow (\text{Add}(e), \lambda hist.\ hist)$$

Figure: The histogram operation: if $\text{histogram}(e, j)$ returns the expression pair (r, f) then $\sum_{j=0}^{n-1} e = f\big(\text{Hist}_{j=0}^{n-1}(r)\big)$. C denotes a context. These rewriting rules are applied top-down, except the second and third rules are prioritized by choosing the rule for which the innermost scope of the free variables $FV(e) \setminus \{j\}$ is outermost.

Code gen with loop optimization