Appendix

SPLICE Complexity Analysis

Algorithm 4 presents the complete SPLICE procedure. SPLICE operates over a sequence of micro-batches. In the analysis below, we assume that each of these micro-batches contains on average $|\mathcal{Q}|$ ground query atoms and $|\mathcal{E}|$ ground evidence atoms, and derive the complexity of SPLICE for the t-th micro-batch $\mathcal{D}_t$ that is received. We denote as $|\mathcal{Q}_L|$ and $|\mathcal{Q}_U|$ the number of labelled and unlabelled ground query atoms in a micro-batch respectively. The algorithmic analysis can be decomposed into four steps, namely data partitioning, label caching and filtering, graph construction and supervision completion (graph cut minimisation). Note that SPLICE also includes a structure learning step, but we do not present any analysis for structure learning, since it can be any online learner.

Data partitioning (see Algorithm 1), invoked at line 5 of Algorithm 4, requires a pass over all true ground evidence atoms for each ground query atom, in order to construct a vertex, and therefore yields a total time $T_1 = |\mathcal{Q}||\mathcal{E}|$.

Cache update (see Algorithm 3, lines 1–5) needs to compare all labelled vertices $\mathcal{V}_L$, present in the current micro-batch, with all cached labelled examples. We assume the worst case scenario where each labelled example in the dataset is unique, leading cache update to yield a total time $T_2 = (t-1)|\mathcal{Q}_L|^2$. Label filtering using the Hoeffding bound (see Algorithm 3, lines 7–16) needs to go through all unique labelled vertices accumulated so far, in order to remove the contradicting ones, if any, yielding a total time of $T_3 = t|\mathcal{Q}_L|$, assuming each labelled example is unique. Both cache update and filtering take place at line 7 of Algorithm 4.

Graph construction (see Algorithm 2, lines 2–5) requires a pass over all pair of vertices, in order compute their similarity. Again we assume the worst case scenario, where no contradicting labels exist and therefore the Hoeffding test does not remove any labelled examples. Thus, graph construction yields a total time $T_4 = (t|\mathcal{Q}_L| + |\mathcal{Q}_U|)^2$.

Graph cut minimisation (Algorithm 2, lines 6–12) includes a matrix subtraction, an inversion of the Laplacian matrix containing the unlabelled vertices and three matrix multiplications. The matrix subtraction yields time $(t|\mathcal{Q}_L| + |\mathcal{Q}_U|)^2$, the inversion of the Laplacian yields $|\mathcal{Q}_U|^3$ and the multiplications yield $|\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + |\mathcal{Q}_U|t|\mathcal{Q}_L|$. The final thresholding step yields time $|\mathcal{Q}_U|$. Therefore altogether the optimisation step yields a total time:

$$T_5 = |\mathcal{Q}_U|^3 + (t|\mathcal{Q}_L| + |\mathcal{Q}_U|)^2 + |\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + |\mathcal{Q}_U|t|\mathcal{Q}_L| + |\mathcal{Q}_U|$$

Graph construction and minimisation take place during supervision completion, invoked at line 9 of Algorithm 4. Combining all four steps, the total time cost of the SPLICE algorithm is given by $T = T_1 + T_2 + T_3 + T_4 + T_5$:

$$
\begin{aligned}
T =& T_1 + T_2 + T_3 + T_4 + T_5 \\
=& |\mathcal{Q}||\mathcal{E}| + (t-1)|\mathcal{Q}_L|^2 + t|\mathcal{Q}_L| + (t|\mathcal{Q}_L| + |\mathcal{Q}_U|)^2 + \\
& |\mathcal{Q}_U|^3 + (t|\mathcal{Q}_L| + |\mathcal{Q}_U|)^2 + |\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + |\mathcal{Q}_U|t|\mathcal{Q}_L| + |\mathcal{Q}_U| \\
=& |\mathcal{Q}_U|^3 + |\mathcal{Q}||\mathcal{E}| + (t-1)|\mathcal{Q}_L|^2 + t|\mathcal{Q}_L| + 2(t|\mathcal{Q}_L| + |\mathcal{Q}_U|)^2 + \\
& |\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + |\mathcal{Q}_U|t|\mathcal{Q}_L| + |\mathcal{Q}_U| \\
=& |\mathcal{Q}_U|^3 + |\mathcal{Q}||\mathcal{E}| + (t-1)|\mathcal{Q}_L|^2 + t|\mathcal{Q}_L| + 2t^2|\mathcal{Q}_L|^2 + 4t|\mathcal{Q}_L||\mathcal{Q}_U| + \\
& 2|\mathcal{Q}_U|^2 + |\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + |\mathcal{Q}_U|t|\mathcal{Q}_L| + |\mathcal{Q}_U| \\
=& |\mathcal{Q}_U|^3 + |\mathcal{Q}||\mathcal{E}| + (2t^2 + t - 1)|\mathcal{Q}_L|^2 + t|\mathcal{Q}_L| + 5t|\mathcal{Q}_L||\mathcal{Q}_U| + \\
& |\mathcal{Q}_U|^2(t|\mathcal{Q}_L| + 2) + |\mathcal{Q}_U| \\
=& |\mathcal{Q}_U|^3 + |\mathcal{Q}||\mathcal{E}| + (2t^2 + t - 1)|\mathcal{Q}_L|^2 + t|\mathcal{Q}_L| + |\mathcal{Q}_U|^2(t|\mathcal{Q}_L| + 2) + \\
& |\mathcal{Q}_U|(5t|\mathcal{Q}_L| + 1) \\
\leq& |\mathcal{Q}_U|^3 + |\mathcal{Q}||\mathcal{E}| + (2t^2 + t - 1)|\mathcal{Q}_L|^2 + t|\mathcal{Q}_L| + 2|\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + \\
& 6|\mathcal{Q}_U|t|\mathcal{Q}_L| \\
\leq& |\mathcal{Q}_U|^3 + |\mathcal{Q}||\mathcal{E}| + (2t^2 + t - 1)|\mathcal{Q}_L|^2 + 3|\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + 6|\mathcal{Q}_U|t|\mathcal{Q}_L|
\end{aligned}
$$

Therefore the algorithm has an asymptotic upper bound complexity:

$$
\mathcal{O}\Big(|\mathcal{Q}_U|^3 + |\mathcal{Q}_U|^2 t|\mathcal{Q}_L| + t^2|\mathcal{Q}_L|^2 + |\mathcal{Q}||\mathcal{E}|\Big)
$$