Capacity Releasing Diffusion for Speed and Locality

A. CRD inner procedure

We first fill in the missing details in the *CRD-inner* subroutine (Algorithm 1).

Note an ineligible arc (v, u) must remain ineligible until the next relabel of v, so we only need to check each arc out of v once between consecutive relabels. We use $\text{current}(v)$ to keep track of the arcs out of v that we have checked since the last relabel of v. We always pick an active vertex v with the lowest label. Then for any eligible arc (v, u), we know $m(u) \leq d(u)$, so we can push at least 1 along (v, u) (without violating $m(u) \leq 2d(u)$), which is crucial to bound the total work. We keep the list Q in non-decreasing order of the vertices' labels, for efficient look-up of the lowest labeled active vertex, and *Add, Remove, Shift* are the operations to maintain this order. Note these operations can be implemented to take $O(1)$ work. In particular, when we add a node u to Q, it will always be the active node with lowest label, so will be put at the beginning. We only remove the first element v from Q, and when we shift a node v in Q, we know $l(v)$ increases by exactly 1. To maintain Q, we simply need to pick two linked lists, one containing all the active nodes with non-decreasing labels, and another linked list containing one pointer for each label value, as long as there is some active node with that label, and the pointer contains the position of first such active node in Q. Maintaining this two lists together can give $O(1)$ time *Add, Remove, Shift*.

Now we proceed to prove the main theorem of *CRD-inner*.

Theorem 1. *Given $G, m(\cdot)$, and $\phi \in (0, 1]$, such that $|m(\cdot)| \leq \text{vol}(G)$, and $\forall v : m(v) \leq 2d(v)$ at the start,* CRD-inner *terminates with one of the following cases*

(1) CRD-inner *finishes the full CRD step: $\forall v : m(v) \leq d(v)$.*

(2) There are nodes with excess, and we can find a cut A of conductance $O(\phi)$. Moreover, $\forall v \in A : 2d(v) \geq m(v) \geq d(v)$, and $\forall v \in \bar{A} : m(v) \leq d(v)$.

The running time is $O(|m(\cdot)| \log |m(\cdot)|/\phi)$.

Proof. Let $l(\cdot)$ be the labels of vertices at termination, and let $B_i = \{v | l(v) = i\}$. We make the following observations: $l(v) = h \Rightarrow 2d(v) \geq m(v) \geq d(v)$; $h > l(v) \geq 1 \Rightarrow m(v) = d(v)$; $l(v) = 0 \Rightarrow m(v) \leq d(v)$.

Algorithm 1 *CRD-inner*(G,$m(\cdot)$,ϕ)

. **Initialization:**
. . $\forall \{v, u\} \in E$, $m(u, v) = m(v, u) = 0$.
. . $Q = \{v | m(v) > d(v)\}$, $h = \frac{3 \log |m(\cdot)|}{\phi}$
. . $\forall v$, $l(v) = 0$, and $\text{current}(v)$ is the first edge in v's list of incident edges.
. **While** Q is not empty
. . Let v be the lowest labeled vertex in Q.
. . *Push/Relabel*(v).
. . **If** *Push/Relabel*(v) pushes mass along (v, u)
. . . **If** v becomes in-active, *Remove*(v, Q)
. . . **If** u becomes active, *Add*(u, Q)
. . **Else If** *Push/Relabel*(v) increases $l(v)$ by 1
. . . **If** $l(v) < h$, *Shift*(v, Q)
. . . **Else** *Remove*(v, Q)

Push/Relabel(v)
. Let $\{v, u\}$ be $\text{current}(v)$.
. **If** arc (v, u) is eligible, then *Push*(v, u).
. **Else**
. . **If** $\{v, u\}$ is not the last edge in v's list of edges.
. . . Set $\text{current}(v)$ be the next edge of v.
. . **Else** (i.e., $\{v, u\}$ is the last edge of v)
. . . *Relabel*(v), and set $\text{current}(v)$ be the first edge of v's list of edges.

Push(v, u)
. **Assertion**: $r_m(v, u) > 0, l(v) \geq l(u) + 1$. $\text{ex}(v) > 0, m(u) < 2d(u)$.
. $\psi = \min(\text{ex}(v), r_m(v, u), 2d(u) - m(u))$
. Send ψ units of mass from v to u:
$m(v, u) \leftarrow m(v, u) + \psi, m(u, v) \leftarrow m(u, v) - \psi$.
$m(v) \leftarrow m(v) - \psi, m(u) \leftarrow m(u) + \psi$.

Relabel(v)
. **Assertion**: v is active, and $\forall u \in V$, $r_m(v, u) > 0 \implies l(v) \leq l(u)$.
. $l(v) \leftarrow l(v) + 1$.

Since $|m(\cdot)| \leq \text{vol}(G)$, if $B_0 = \emptyset$, it must be $|m(\cdot)| = \text{vol}(G)$, and every v has $m(v) = d(v)$, so we get case (1). If $B_h = \emptyset$, we also get case (1).

If $B_h, B_0 \neq \emptyset$, let $S_i = \cup_{j=i}^{h} B_j$ be the set of nodes with label at least i. We have h level cuts $S_h, \ldots, S_1$, where $\text{vol}(S_h) \geq 1$, and $S_j \subseteq S_i$ if $j > i$. We claim one of these level cuts must have conductance $O(\phi)$. For any S_i, we divide the edges from S_i to $\overline{S_i}$ into two groups: 1) edge across one level (i.e., from node in B_i to node in B_{i-1}), and 2) edges across more than one level. Let $z_1(i), z_2(i)$ be the number of edges in the two groups respectively, and define $\phi_g(i) \stackrel{\text{def}}{=} z_g(i)/\text{vol}(S_i)$ for $g = 1, 2$.

First we show that, there must be a i^* between h and $h/2$ such that $\phi_1(i^*) \leq \phi$. By contradiction, if $\phi_1(i) > \phi$ for all $i = h, \ldots, h/2$, since $\text{vol}(S_{i-1}) \geq \text{vol}(S_i)(1 + \phi_1(S_i))$, we get $\text{vol}(S_{h/2}) \geq (1 + \phi)^{h/2}\text{vol}(S_h)$. With $h = 3\log|m(\cdot)|/\phi$, we have $\text{vol}(S_{h/2}) \geq \Omega(|m(\cdot)|^{3/2})$, and since nodes in $S_{h/2}$ are all saturated, we get a contradiction since we must have $\text{vol}(S_{h/2}) \leq |m(\cdot)|$.

Now we consider any edge $\{v, u\}$ counted in $z_2(i^*)$ (i.e., $v \in S_{i^*}, u \in \overline{S_{i^*}}, l(v) - l(u) \geq 2$). Since $i^* \geq h/2 > 1/\phi$, $\hat{c}(v, u) = 1/\phi$. $l(v) - l(u) > 2$ suggests $r_m(v, u) = 0$, thus $m(v, u) = 1/\phi$ (i.e., $1/\phi$ mass pushed out of S_{i^*} along each edge counted in $z_2(i^*)$). Each edge counted in $z_1(i^*)$ can have at most $1/\phi$ mass pushed into S_{i^*}, and at most $2\text{vol}(S_{i^*})$ mass can start in S_{i^*}, then we know

$$z_2(i^*)/\phi \leq z_1(i^*)/\phi + 2\text{vol}(S_{i^*})$$

We will let A be S_{i^*}, and we have

$$\phi(A) = \frac{z_1(i^*) + z_2(i^*)}{\text{vol}(S_{i^*})} \leq 4\phi = O(\phi)$$

Here we assume S_{i^*} is the smaller side of the cut to compute the conductance. If this is not the case, i.e. $\text{vol}(S_{i^*}) > \text{vol}(G)/2$, we just carry out the same argument as above, but run the region growing argument from level $h/4$ up to level $h/2$, and get a low conductance cut, and still let A to be the side containing S_h. The additional properties of elements in A follows from $S_h \subseteq A \subseteq S_{h/4}$.

Now we proceed to the running time. The initialization takes $O(|m(\cdot)|)$. Subsequently, each iteration takes $O(1)$ work. We will first attribute the work in each iteration to either a push or a relabel. Then we will charge the work on pushes and relabels to the absorbed mass, such that each unit of absorbed mass gets charged $O(h)$ work. Recall the absorbed mass at v are the first up to $d(v)$ mass starting at or pushed into v, and these mass never leave v, as the algorithm only pushes excess mass. This will prove the result, as there are at most $|m(\cdot)|$ units of (absorbed) mass in total.

In each iteration of *Unit-Flow*, the algorithm picks a lowest labeled active node v. If *Push/Relabel*(v) ends with a push of ψ mass, we charge $O(\psi)$ to that push operation. Since $\psi \geq 1$, we charged the push enough to cover the work in that iteration. If the call to *Push/Relabel*(v) doesn't push, we charge the $O(1)$ work of the iteration to the next relabel of v (or the last relabel if there is no next relabel). The latter can happen at most $d(v)$ times between consecutive relabels of v, so each relabel of v is charged $O(d(v))$ work.

We now charge the work on pushes and relabels to the absorbed mass. Note each time we relabel v, there are $d(v)$ units of absorbed mass at v, so we charge the $O(d(v))$ work on the relabel to the absorbed mass, and each unit gets charged $O(1)$. There is at most h relabels of v, so each unit of absorbed mass is charged $O(h)$ in total by all the relabels.

For the work on pushes, we consider the potential function $\Lambda = \sum_v \text{ex}(v)l(v)$. Λ is always non-negative, and as we only push excess mass downhill, each push of ψ units of mass decrease Λ by at least ψ, so we can charge the work on pushes to the increment of Λ. It only increases at relabel. When we relabel v, Λ is increased by $\text{ex}(v)$. Since $\text{ex}(v) \leq d(v)$, we can charge $O(1)$ to each unit of absorbed mass at v to cover Λ's increment. In total we can charge all pushes (via Λ) to absorbed mass, and each unit is charged with $O(h)$.

If we need to compute the cut A in case (2), the running time is $O(\text{vol}(S_1))$, which is $O(|m(\cdot)|)$. □

B. Local Clustering

Recall we assume B to satisfy the following conditions.

Assumption 1. $\sigma_1 \stackrel{\text{def}}{=} \frac{\phi_S(B)}{\phi(B)} \geq \Omega(1)$

Assumption 2. *There exists $\sigma_2 \geq \Omega(1)$, such that any $T \subset B$ with $vol_B(T) \leq \text{vol}_B(B)/2$ satisfies*

$$\frac{|E(T, B \setminus T)|}{|E(T, V \setminus B)| \log \text{vol}(B) \log \frac{1}{\phi_S(B)}} \geq \sigma_2.$$

Now we proceed to prove the main lemma.

Lemma 1. *In the j-th CRD step, let M_j be the total amount of mass in B at the start, and L_j be the amount of mass that ever leaves B during the diffusion, then $L_j \leq O(\frac{1}{\sigma_2 \log \text{vol}(B)}) \cdot M_j$ when $M_j \leq \text{vol}_B(B)/2$, and $L_j \leq O(\frac{1}{\sigma_1}) \cdot M_j$ when $M_j \geq \text{vol}_B(B)/2$.*

Proof. For simplicity, we assume once a unit of mass leaves B, it is never routed back. Intuitively, mass coming back into B should only help the algorithm, and indeed the results don't change without this assumption. We denote $|M_j(S)|$ as the amount of mass on nodes in a set S at the start of the *CRD-inner* call.

We have two cases, corresponding to whether the diffusion already spread a lot of mass over B. If $M_j \geq \text{vol}_B(B)/2$,

we use the upperbound $1/\phi$ that is enforced on the net mass over any edge to limit the amount of mass that can leak out. In particular $L_j \leq O(\mathrm{vol}(B)\phi(B)/\phi_S(B))$, since there are $\mathrm{vol}(B)\phi(B)$ edges from B to $\bar{B}$, and $\phi = \Theta(\phi_S(B))$ in *CRD-inner*. As $M_j \geq \Omega(\mathrm{vol}(B))$, we have $L_j \leq O(\frac{1}{\sigma_1}) \cdot M_j$.

The second case is when $M_j \leq \mathrm{vol}_B(B)/2$. In this case, a combination of Assumption 2 and capacity releasing controls the leakage of mass. Intuitively, there are still many nodes in B that the diffusion can spread mass to. For the nodes in B with excess on them, when they push their excess, most of the downhill directions go to nodes inside B. As a consequence of capacity releasing, only a small fraction of mass will leak out.

In particular, let $l(\cdot)$ be the labels on nodes when *CRD-inner* finishes, we consider $B_i = \{v \in B | l(v) = i\}$ and the level cuts $S_i = \{v \in B | l(v) \geq i\}$ for $i = h, \ldots, 1$. As $M_j \leq \mathrm{vol}_B(B)/2$, we know $\mathrm{vol}(S_h) \leq \mathrm{vol}(S_{h-1}) \leq \ldots \leq \mathrm{vol}(S_1) \leq \mathrm{vol}_B(B)/2$. In this case, we can use Assumption 2 on all level cuts $S_h, \ldots, S_1$. Moreover, for a node $v \in B_i$, the "'effective"' capacity of an arc from v to $\bar{B}$ is $\min(i, 1/\phi)$. Formally, we can bound L_j by the total (effective) outgoing capacity, which is

$$\sum_{i=1}^{h} |E(B_i, \bar{B})| \cdot \min(i, \frac{1}{\phi}) = \sum_{i=1}^{\frac{1}{\phi}} |E(S_i, \bar{B})| \qquad (1)$$

where h is the bound on labels used in unit flow.

We design a charging scheme to charge the above quantity (the right hand side) to the mass in $\Delta_j(B)$, such that each unit of mass is charged $O(1/(\sigma_2 \log \mathrm{vol}(B)))$. It follows that $L_j \leq O(\frac{1}{\sigma_2 \log \mathrm{vol}(B)}) \cdot |\Delta_j(B)|$.

Recall that, $|E(S_i, \bar{B})| \leq \frac{|E(S_i, B \setminus S_i)|}{\sigma_2 \log \mathrm{vol}(B) \log(1/\phi)}$ from Assumption 2. We divide edges in $E(S_i, B \setminus S_i)$ into two groups: : 1) edges across one level, and 2) edges across more than one level. Let $z_1(i), z_2(i)$ be the number of edges in the two groups respectively.

If $z_1(i) \geq |E(S_i, B \setminus S_i)|/3$, we charge $3/(\sigma_2 \log \mathrm{vol}(B) \log(1/\phi))$ to each edge in group 1. These edges in turn transfer the charge to the absorbed mass at their endpoints in B_i. Since each node v in level $i \geq 1$ has $d(v)$ absorbed mass, each unit of absorbed mass is charged $O(1/(\sigma_2 \log \mathrm{vol}(B) \log(1/\phi)))$. Note that the group 1 edges of different level i's are disjoint, so each unit of absorbed mass will only be charged once this way.

If $z_1(i) \leq |E(S_i, B \setminus S_i)|/3$, we know $z_2(i) - z_1(i) \geq |E(S_i, B \setminus S_i)|/3$. Group 2 edges in total send at least $(i-1)z_2(i)$ mass from S_i to $B \setminus S_i$, and at most $(i-1)z_1(i)$ of these mass are pushed into S_i by group 1 edges. Thus, there are at least $(i-1)|E(S_i, B \setminus S_i)|/3$ mass that start in S_i, and are absorbed by nodes at level below i (possibly outside B). In particular, this suggests $|M_j(S_i)| \geq (i-1)|E(S_i, B \setminus S_i)|/3$, and we split the total charge $|E(S_i, \bar{B})|$ evenly on these mass, so each unit of mass is charged $O(1/(i\sigma_2 \log \mathrm{vol}(B) \log(1/\phi)))$. Since we sum from $i = 1/\phi$ to 1 in (RHS of) Eqn (1), we charge some mass multiple times (as S_i's not disjoint), but we can bound the total charge by $\sum_{i=1}^{1/\phi} \frac{1}{i} \cdot O(1/(\sigma_2 \log \mathrm{vol}(B) \log(1/\phi)))$, which is $O(1/(\sigma_2 \log \mathrm{vol}(B)))$. This completes the proof. □

C. Empirical Set-up and Results

C.1. Datasets

We chose the graphs of John Hopkins, Rice, Simmons and Colgate universities/colleges. The actual IDs of the graphs in Facebook100 dataset are Johns_Hopkins55, Rice31, Simmons81 and Colgate88. These graphs are anonymized Facebook graphs on a particular day in September 2005 for student social networks. The graphs are unweighted and they represent "friendship ties". The data form a subset of the Facebook100 dataset from (Traud et al., 2012). We chose these 4 graphs out of 100 due to their large assortativity value in the first column of Table A.2 in (Traud et al., 2012), where the data were first introduced and analyzed. Details about the graphs are shown is Table 1.

Graph	volume	nodes	edges
John Hopkins	373144	5157	186572
Rice	369652	4083	184826
Simmons	65968	1510	32984
Colgate	310086	3482	155043

Table 1. Graphs used for experiments.

Each graph in the Facebook dataset comes along with 6 features, i.e., second major, high school, gender, dorm, major index and year. We construct "ground truth" clusters by using the features for each node. In particular, we consider nodes with the same value of a feature to be a cluster, e.g., students of year 2009. We loop over all possible clusters and consider as ground truth the ones that have volume larger than 1000, conductance smaller than 0.5 and gap larger than 0.5. Filtering results in moderate scale clusters for which the internal volume is at least twice as much as the volume of the edges that leave the cluster. Additionally, gap at least 0.5 means that the smallest nonzero eigenvalue of the normalized Laplacian of the subgraph defined by the cluster is at least twice larger than the conductance of the cluster in the whole graph. The clusters per graph that satisfy the latter constraints are shown in Table 2. Notice that the clusters which remain after filtering correspond to features year or dorm. This agrees with (Traud et al., 2012)

in which it is stated that the features with clusters with the best assortativity value correspond to the feature of year or dorm.

	year/dorm	volume	size	gap	cond.
Hop.	217	10696	200	1.48	0.26
	2009	32454	886	0.67	0.19
Rice	203	43321	403	0.58	0.46
	2009	30858	607	0.73	0.33
Sim.	2007	14424	281	0.57	0.47
	2009	11845	277	5.35	0.1
Colgate	2006	62064	556	0.57	0.48
	2007	68381	588	0.69	0.41
	2008	62429	640	1.19	0.29
	2009	35369	638	3.49	0.11

Table 2. Clusters of chosen graphs in Table 1, see Subsection C.1 for details.

C.2. Performance criteria and parameter tuning

For real-world Facebook graphs since we calculate the ground truth clusters in Table 2 then we measure performance by calculating precision and recall for the output clusters of the algorithms.

We set the parameters of CRD to $\phi = 1/3$ for all experiments. At each iteration we use sweep cut on the labels returned by the *CRD-inner* subroutine to find a cut of small conductance, and over all iterations of CRD we return the cluster with the lowest conductance.

ACL has two parameters, the teleportation parameter α and a tolerance parameter ϵ. Ideally the former should be set according to the reciprocal of the mixing time of a a random walk within the target cluster, which is equal to the smallest nonzero eigenvalue of the normalized Laplacian for the subgraph that corresponds to the target cluster. Let us denote the eigenvalue with λ. In our case the target cluster is a ground truth cluster from Table 2. We use this information to set parameter α. In particular, for each node in the clusters in Table 2 we run ACL 4 times where α is set based on a range of values in $[\lambda/2, 2\lambda]$ with a step of $(2\lambda - \lambda/2)/4$. The tolerance parameter ϵ is set to 10^{-7} for all experiments in order to guarantee accurate solutions for the PageRank linear system. For each parameter setting we use sweep cut to find a cluster of low conductance, and over all parameter settings we return the cluster with the lowest conductance value as an output of ACL.

For real-world experiments we show results for ACLopt. In this version of ACL, for each parameter setting of α we use sweep cut algorithm to obtain a low conductance cluster and then we compute its precision and recall. Over all parameter settings we keep the cluster with the best F1-score; a combination of precision and recall. This is an extra level of supervision for the selection of the teleportation parameter α, which is not possible in practice since it requires ground truth information. However, the performance of ACLopt demonstrates the performance of ACL in case that we could make optimal selection of parameter α among the given range of parameters (which also includes ground truth information) for the precision and recall criteria.

Finally, we set the reference set of FlowI to be the output set of best conductance of ACL out of its 4 runs for each node. By this we aim to obtain an improved cluster to ACL in terms of conductance. Note that FlowI is a global algorithm, which means that it accesses the information from the whole graph compared to CRD and ACL which are local algorithms.

C.3. Real-world experiments

For clusters in Table 2 we sample uniformly at random half of their nodes. For each node we run CRD, ACL and ACL+FlowI. We report the results using box plots, which graphically summarizes groups of numerical data using quartiles. In these plot the orange line is the median, the blue box below the median is the first quartile, the blue box above the median is the third quartile, the extended long lines below and above the box are the maximum and minimum values and the circles are outliers.

The results for John Hopkins university are shown in Figure 1. Notice in this figure that CRD performs better than ACL and ACLopt, which both use ground truth information, see parameter tuning in Subsection C.2. CRD performs similarly to ACL+FlowI, where FlowI is a global algorithm, but CRD is a local algorithm. Overall all methods have large medians for this graph because the clusters with dorm 217 and year 2009 are clusters with low conductance compared to the ones in other universities/colleges which we will discuss in the remaining experiments of this subsection.

The results for Rice university are shown in Figure 2. Notice that both clusters of dorm 203 and year 2009 for Rice university are worse in terms of conductance compared to the clusters of John Hopkins university. Therefore the performance of the methods is decreased. For the cluster of dorm 203 with conductance 0.46 CRD has larger median than ACL, ACLopt and ACL+Flow in terms of precision. The latter methods obtain larger median for recall, but this is because ACL leaks lots of probability mass outside of the ground truth cluster since as indicated by its large conductance value many nodes in this cluster are connected externally. For cluster of year 2009 CRD outperforms ACL, which fails to recover the cluster because it leaks mass outside the cluster, FlowI corrects the problem and locates the correct cluster at the expense of touching the whole graph.

Notice that all methods have a significant amount of variance and outliers, which is also explained by the large conductance values of the clusters.

The results for Simmons college are shown in Figure 3. Notice that Simmons college in Table 2 has two clusters, one with poor conductance 0.47 for students of year 2007 and one low conductance 0.1 for students of year 2009. The former with conductance 0.47 means that the internal volume is nearly half the volume of the outgoing edges. This has a strong implication in the performance of CRD, ACL and ACLopt which get median precision about 0.5. This happens because the methods push half of the flow (CRD) and half of the probability mass (ACL) outside the ground truth cluster, which results in median precision 0.5. ACL achieves about 20% more (median) recall than CRD but this is because ACL touched more nodes than CRD during execution of the algorithm. Notice that ACL+FlowI fails for the cluster of year 2007, this is because FlowI is a global algorithm, hence it finds a cluster that has low conductance but it is not the ground truth cluster. The second cluster of year 2009 has low conductance hence all methods have large median performance with CRD being slightly better than ACL, ACLopt and ACL+FlowI.

The results for Colgate university are shown in Figure 4. The interesting property of the clusters in Table 2 for Colgate university is that their conductance varies from low 0.1 to large 0.48. Therefore in Figure 4 we see a smooth transition of performance for all methods from poor to good performance. In particular, for the cluster of year 2006 the conductance is 0.48 and CRD, ACL and ACLopt perform poorly by having median precision about 50%, recall is slightly better for ACL but this is because we allow it touch a bigger part of the graph. ACL+FlowI fails to locate the cluster. For the cluster of year 2007 the conductance is 0.41 and the performance of CRD, ACL and ACLopt is increased with CRD having larger (median) precision and ACL having larger (median) recall as in the previous cluster. Conductance is smaller for the cluster of year 2008, for which we observe substantially improved performance for CRD with large median precision and recall. On the contrary, ACL, ACLopt and ACL+FlowI have nearly 30% less median precision in the best case and similar median recall, but only because a large amount of probability mass is leaked and a big part of the graph is touched which includes the ground truth cluster. Finally, the cluster of year 2009 has low conductance 0.11 and all methods have good performance for precision and recall.

References

Traud, A. L., Mucha, P. J., and Porter, M. A. Social structure of facebook networks. *Physica A: Statistical Mechanics and its Applications*, 391(16):4165–4180, 2012.

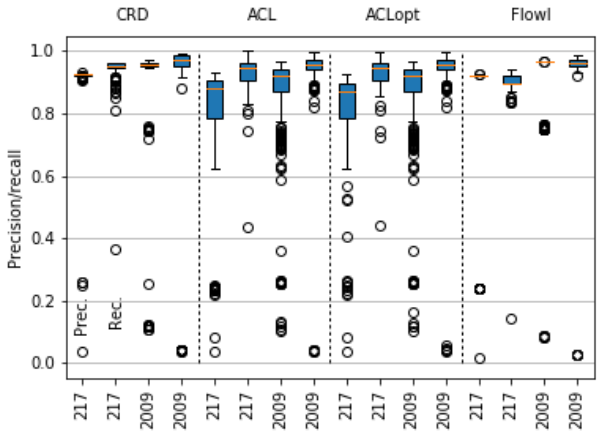

Figure 1. Precision and recall results for John Hopkins university

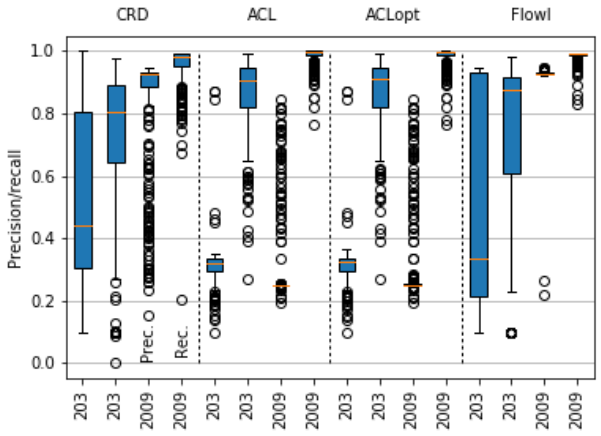

Figure 2. Precision and recall results for Rice university

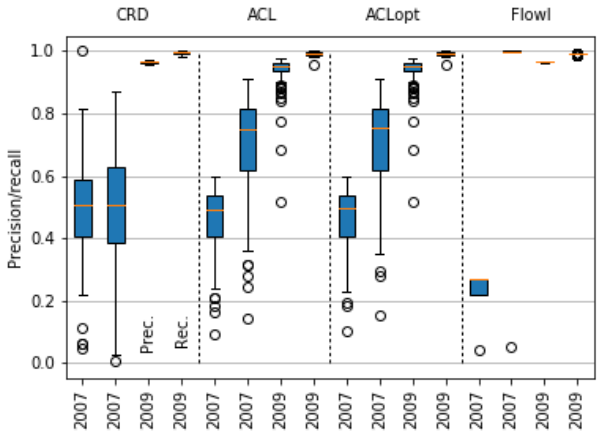

Figure 3. Precision and recall results for Simmons college

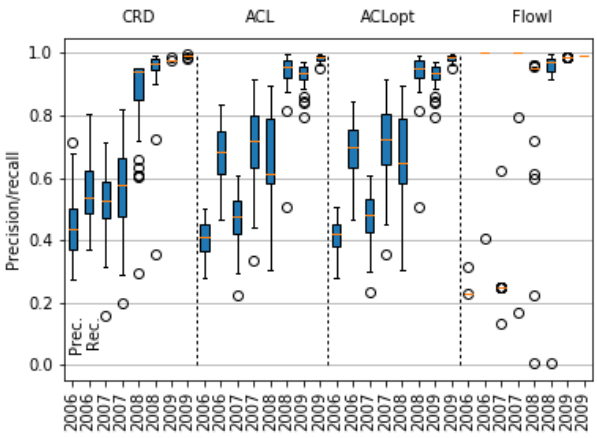

Figure 4. Precision and recall results for Colgate university