Incentivising Collaboration in Machine Learning via Synthetic Data Rewards

Main reason to choose incentive as Synthetic Data

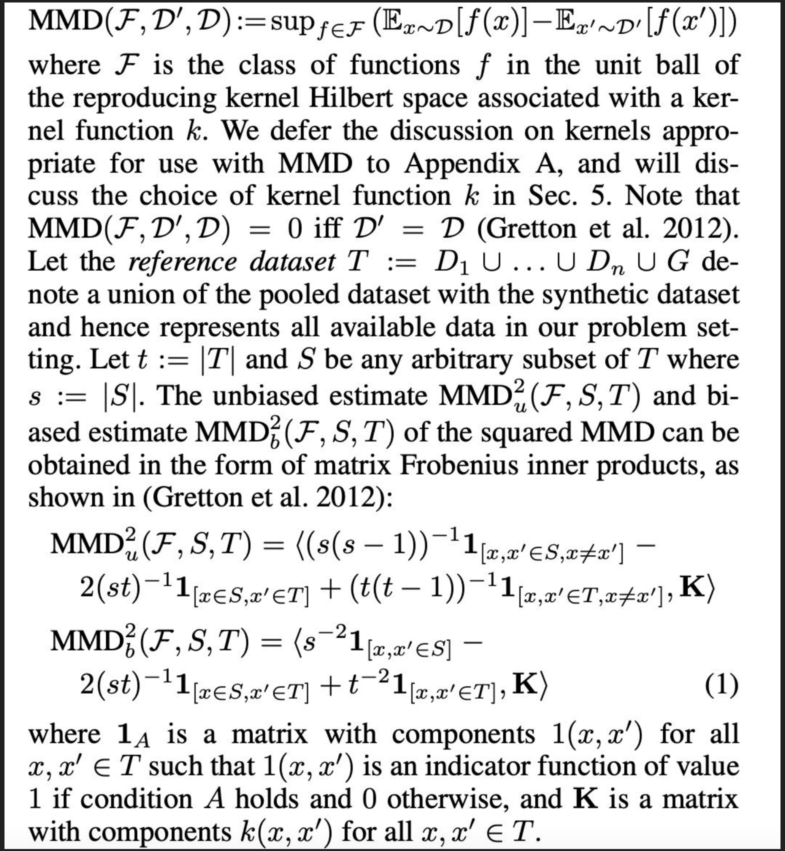
In previous methods

1. A user cannot use the received rewards on a different type of kernel/learning algorithm and limits each party's flexibility to experiment with different model parameters and architectures.
2. Cannot perform a different learning task on the same dataset as the reward is a trained model

Contributions of the paper

1. Data Valuation function using Maximum Mean Discrepancy that values data based on its quantity and quality in terms of its closeness to the true data distribution.
2. Formulate the reward scheme as a linear optimisation problem that when solved, guarantees certain incentives such as fairness in the CGM framework.
3. Weighted sampling algorithm for generating synthetic data such that it matches that value of reward assigned to the respective party.

Introduction to MMD

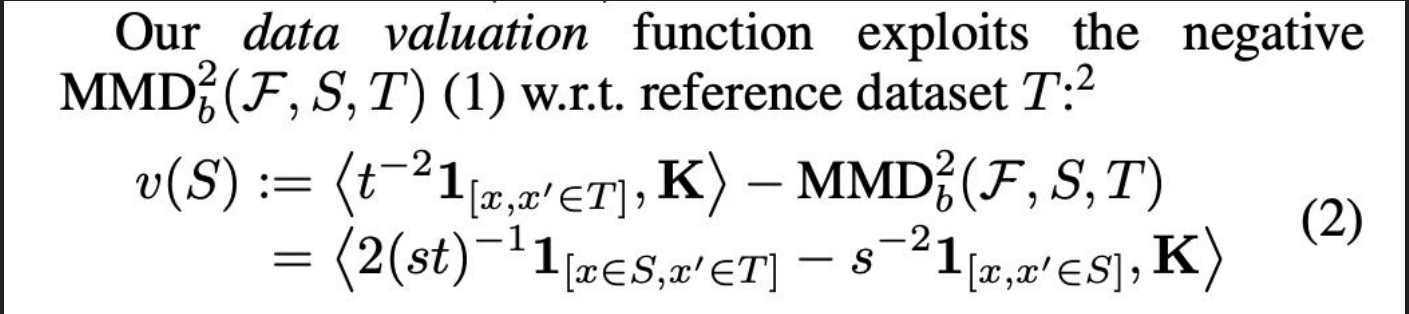


$\mathrm{MMD}(\mathcal{F},\mathcal{D}',\mathcal{D}) := \sup_{f\in\mathcal{F}} \left(\mathbb{E}_{x\sim\mathcal{D}}[f(x)] - \mathbb{E}_{x'\sim\mathcal{D}'}[f(x')]\right)$ where $\mathcal{F}$ is the class of functions f in the unit ball of the reproducing kernel Hilbert space associated with a kernel function k. We defer the discussion on kernels appropriate for use with MMD to Appendix A, and will discuss the choice of kernel function k in Sec. 5. Note that $\mathrm{MMD}(\mathcal{F},\mathcal{D}',\mathcal{D}) = 0$ iff $\mathcal{D}' = \mathcal{D}$ (Gretton et al. 2012). Let the *reference dataset* $T := D_1 \cup \ldots \cup D_n \cup G$ denote a union of the pooled dataset with the synthetic dataset and hence represents all available data in our problem setting. Let $t := |T|$ and S be any arbitrary subset of T where $s := |S|$. The unbiased estimate $\mathrm{MMD}_u^2(\mathcal{F}, S, T)$ and biased estimate $\mathrm{MMD}_b^2(\mathcal{F}, S, T)$ of the squared MMD can be obtained in the form of matrix Frobenius inner products, as shown in (Gretton et al. 2012):

$$\begin{aligned}
\mathrm{MMD}_u^2(\mathcal{F}, S, T) &= \langle (s(s-1))^{-1}\mathbf{1}_{[x,x'\in S, x\neq x']} - \\
&\quad 2(st)^{-1}\mathbf{1}_{[x\in S, x'\in T]} + (t(t-1))^{-1}\mathbf{1}_{[x,x'\in T, x\neq x']}, \mathbf{K}\rangle \\
\mathrm{MMD}_b^2(\mathcal{F}, S, T) &= \langle s^{-2}\mathbf{1}_{[x,x'\in S]} - \\
&\quad 2(st)^{-1}\mathbf{1}_{[x\in S, x'\in T]} + t^{-2}\mathbf{1}_{[x,x'\in T]}, \mathbf{K}\rangle
\end{aligned} \tag{1}$$

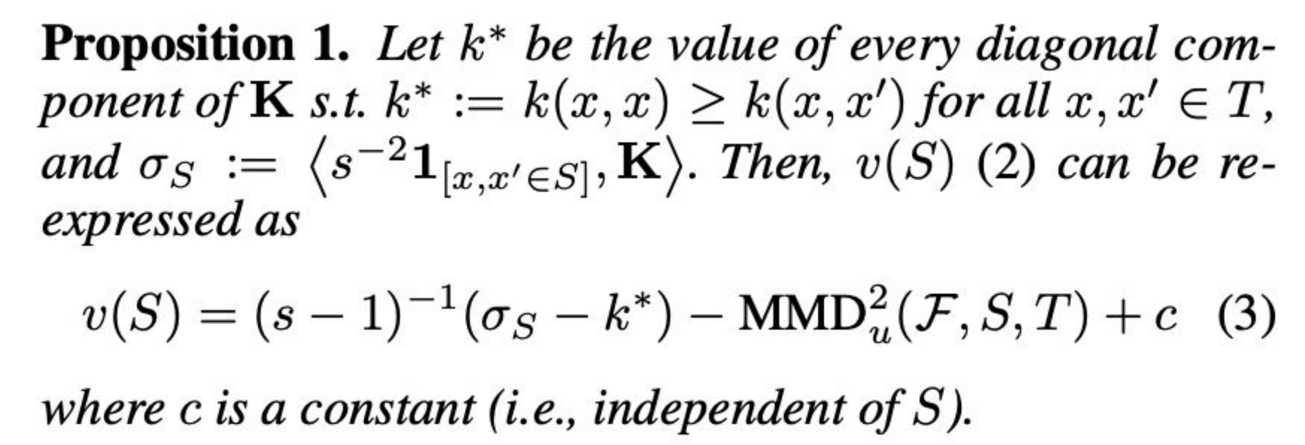
where $\mathbf{1}_A$ is a matrix with components $1(x, x')$ for all $x, x' \in T$ such that $1(x, x')$ is an indicator function of value 1 if condition A holds and 0 otherwise, and $\mathbf{K}$ is a matrix with components $k(x, x')$ for all $x, x' \in T$.

Data Valuation using MMD



Our *data valuation* function exploits the negative $\mathrm{MMD}_b^2(\mathcal{F}, S, T)$ (1) w.r.t. reference dataset T:[2]

$$\begin{aligned} v(S) &:= \left\langle t^{-2}\mathbf{1}_{[x,x'\in T]}, \mathbf{K}\right\rangle - \mathrm{MMD}_b^2(\mathcal{F}, S, T) \\ &= \left\langle 2(st)^{-1}\mathbf{1}_{[x\in S, x'\in T]} - s^{-2}\mathbf{1}_{[x,x'\in S]}, \mathbf{K}\right\rangle \end{aligned} \tag{2}$$



Proposition 1. *Let k^* be the value of every diagonal component of $\mathbf{K}$ s.t. $k^* := k(x, x) \geq k(x, x')$ for all $x, x' \in T$, and $\sigma_S := \left\langle s^{-2}\mathbf{1}_{[x,x'\in S]}, \mathbf{K}\right\rangle$. Then, $v(S)$ (2) can be re-expressed as*

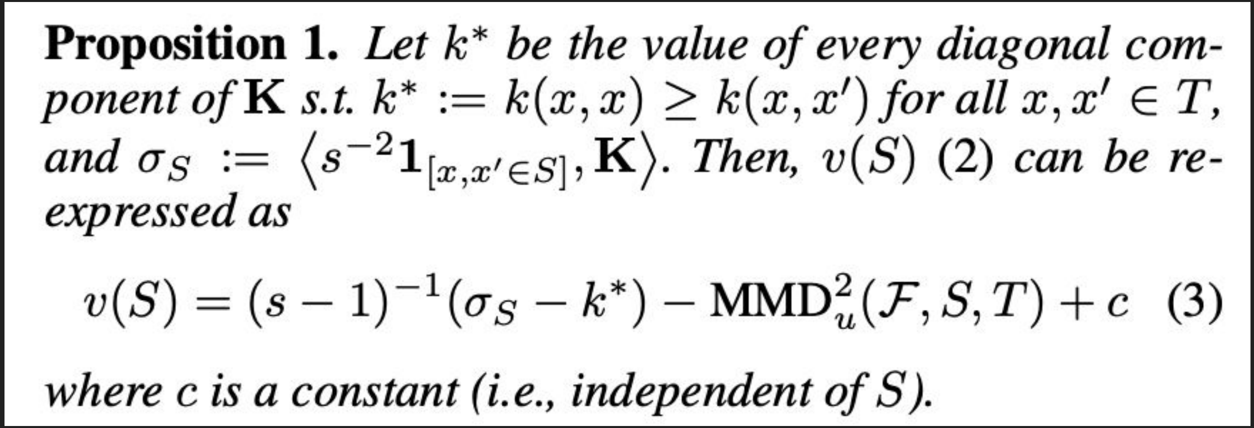
$$v(S) = (s-1)^{-1}(\sigma_S - k^*) - \mathrm{MMD}_u^2(\mathcal{F}, S, T) + c \tag{3}$$

where c is a constant (i.e., independent of S).

Proving that the valuation satisfies the condition that having more data is never worse off

First, we can see that v(s) in the format in the previous slide is directly proportional to (s-1) but we cannot directly conclude as the value of σ_S can change with s.

Hence a more formal proof is



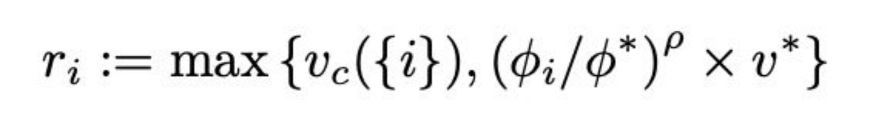
Proposition 1. *Let k^* be the value of every diagonal component of $\mathbf{K}$ s.t. $k^* := k(x,x) \geq k(x,x')$ for all $x, x' \in T$, and $\sigma_S := \langle s^{-2}\mathbf{1}_{[x,x' \in S]}, \mathbf{K}\rangle$. Then, $v(S)$ (2) can be re-expressed as*

$$v(S) = (s-1)^{-1}(\sigma_S - k^*) - \mathrm{MMD}_u^2(\mathcal{F}, S, T) + c \quad (3)$$

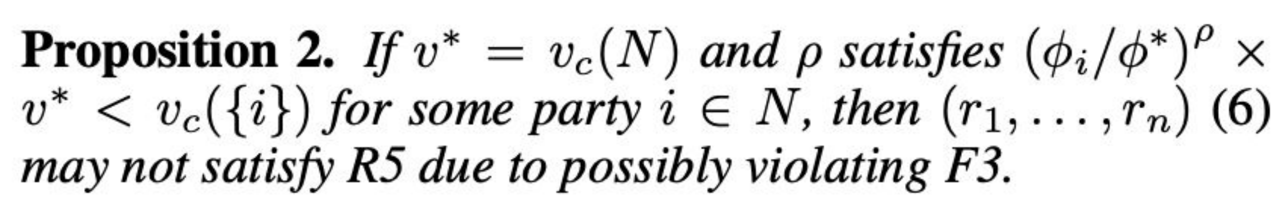
where c is a constant (i.e., independent of S).

Reward Scheme for Guaranteeing Incentives in CGM Framework

Modified p-Shapley Fair Reward Values

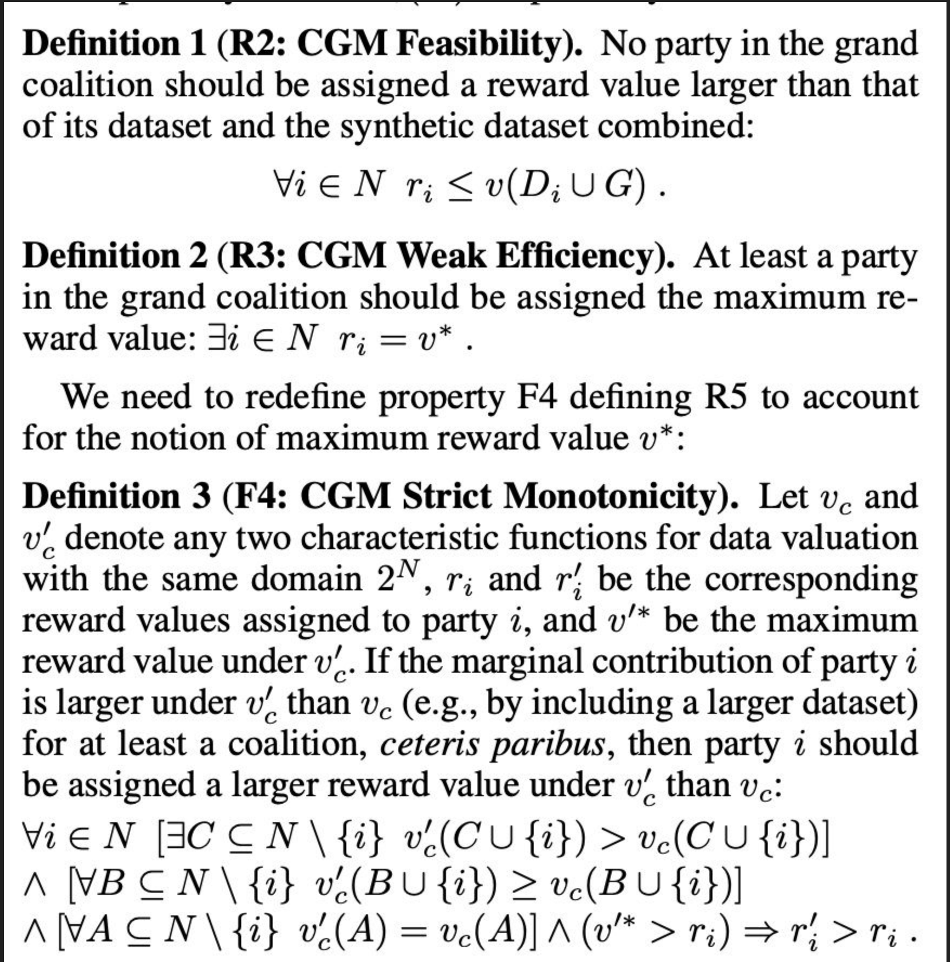


$$r_i := \max \{v_c(\{i\}), (\phi_i/\phi^*)^\rho \times v^*\}$$



Proposition 2. *If* $v^* = v_c(N)$ *and* ρ *satisfies* $(\phi_i/\phi^*)^\rho \times v^* < v_c(\{i\})$ *for some party* $i \in N$*, then* $(r_1, \ldots, r_n)$ (6) *may not satisfy R5 due to possibly violating F3.*

Modified Definitions



Definition 1 (R2: CGM Feasibility). No party in the grand coalition should be assigned a reward value larger than that of its dataset and the synthetic dataset combined:

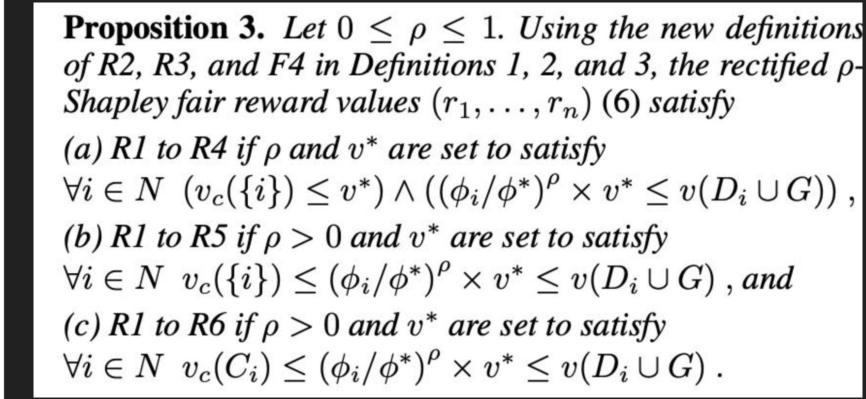
$$\forall i \in N \;\; r_i \leq v(D_i \cup G) \; .$$

Definition 2 (R3: CGM Weak Efficiency). At least a party in the grand coalition should be assigned the maximum reward value: $\exists i \in N \;\; r_i = v^*$.

We need to redefine property F4 defining R5 to account for the notion of maximum reward value v^*:

Definition 3 (F4: CGM Strict Monotonicity). Let v_c and v'_c denote any two characteristic functions for data valuation with the same domain 2^N, r_i and r'_i be the corresponding reward values assigned to party i, and v'^* be the maximum reward value under v'_c. If the marginal contribution of party i is larger under v'_c than v_c (e.g., by including a larger dataset) for at least a coalition, *ceteris paribus*, then party i should be assigned a larger reward value under v'_c than v_c:

$$\begin{aligned}
&\forall i \in N \;\; [\exists C \subseteq N \setminus \{i\} \;\; v'_c(C \cup \{i\}) > v_c(C \cup \{i\})] \\
&\wedge \;\; [\forall B \subseteq N \setminus \{i\} \;\; v'_c(B \cup \{i\}) \geq v_c(B \cup \{i\})] \\
&\wedge \, [\forall A \subseteq N \setminus \{i\} \;\; v'_c(A) = v_c(A)] \wedge (v'^* > r_i) \Rightarrow r'_i > r_i \; .
\end{aligned}$$



Proposition 3. *Let $0 \leq \rho \leq 1$. Using the new definitions of R2, R3, and F4 in Definitions 1, 2, and 3, the rectified ρ-Shapley fair reward values $(r_1, \ldots, r_n)$ (6) satisfy*

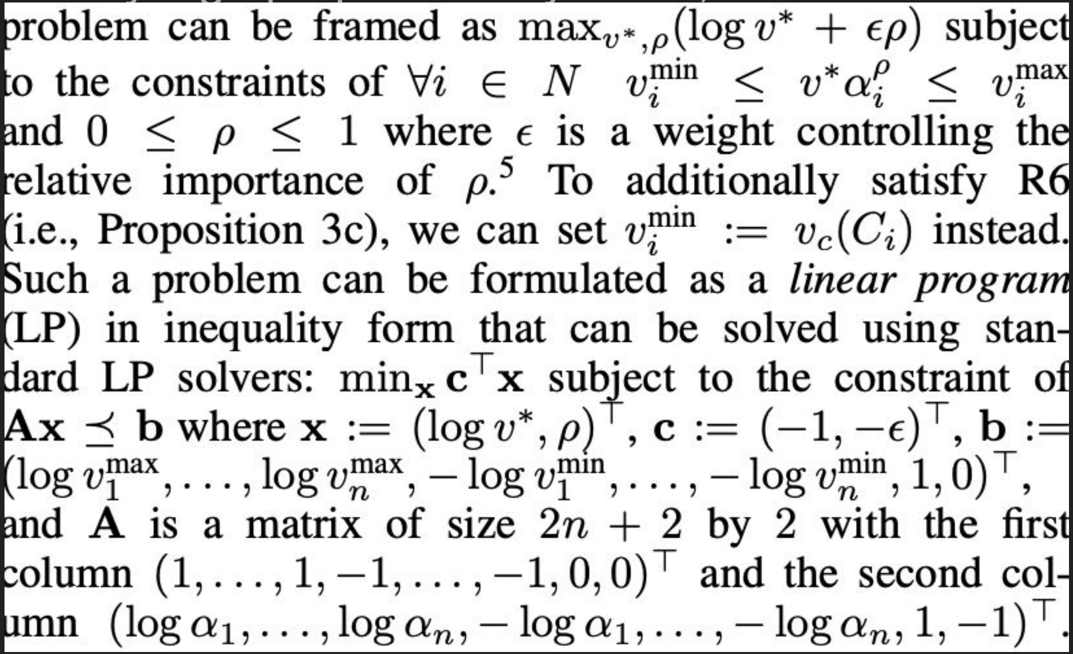
(a) R1 to R4 if ρ and v^* are set to satisfy
$\forall i \in N \;\; (v_c(\{i\}) \leq v^*) \wedge ((\phi_i/\phi^*)^\rho \times v^* \leq v(D_i \cup G))$,

(b) R1 to R5 if $\rho > 0$ and v^* are set to satisfy
$\forall i \in N \;\; v_c(\{i\}) \leq (\phi_i/\phi^*)^\rho \times v^* \leq v(D_i \cup G)$ *, and*

(c) R1 to R6 if $\rho > 0$ and v^* are set to satisfy
$\forall i \in N \;\; v_c(C_i) \leq (\phi_i/\phi^*)^\rho \times v^* \leq v(D_i \cup G)$.

Optimising r_i via a linear program

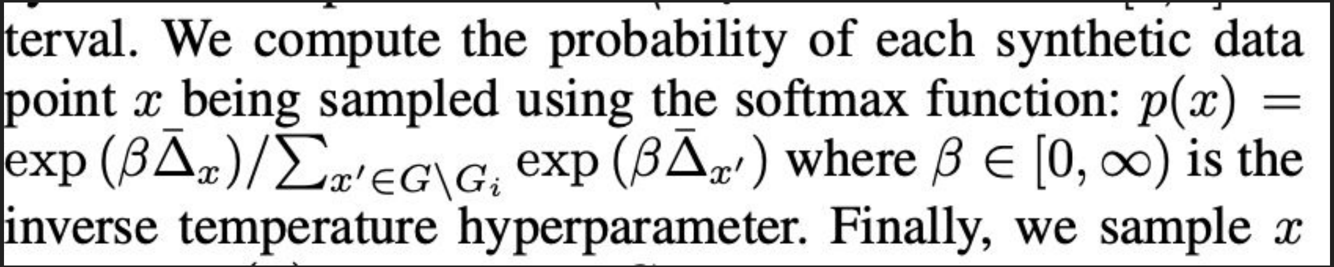
We want to maximise the summation of all r_i to satisfy group welfare and at the same time hold a relatively high proportionality factor, hence



problem can be framed as $\max_{v^*,\rho}(\log v^* + \epsilon\rho)$ subject to the constraints of $\forall i \in N \quad v_i^{\min} \le v^*\alpha_i^{\rho} \le v_i^{\max}$ and $0 \le \rho \le 1$ where ϵ is a weight controlling the relative importance of ρ.[5] To additionally satisfy R6 (i.e., Proposition 3c), we can set $v_i^{\min} := v_c(C_i)$ instead. Such a problem can be formulated as a *linear program* (LP) in inequality form that can be solved using standard LP solvers: $\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x}$ subject to the constraint of $\mathbf{A}\mathbf{x} \preceq \mathbf{b}$ where $\mathbf{x} := (\log v^*, \rho)^\top$, $\mathbf{c} := (-1, -\epsilon)^\top$, $\mathbf{b} := (\log v_1^{\max}, \ldots, \log v_n^{\max}, -\log v_1^{\min}, \ldots, -\log v_n^{\min}, 1, 0)^\top$, and $\mathbf{A}$ is a matrix of size $2n+2$ by 2 with the first column $(1, \ldots, 1, -1, \ldots, -1, 0, 0)^\top$ and the second column $(\log \alpha_1, \ldots, \log \alpha_n, -\log \alpha_1, \ldots, -\log \alpha_n, 1, -1)^\top$.

Distributing Synthetic data to Parties via Weighted Sampling

We define β as

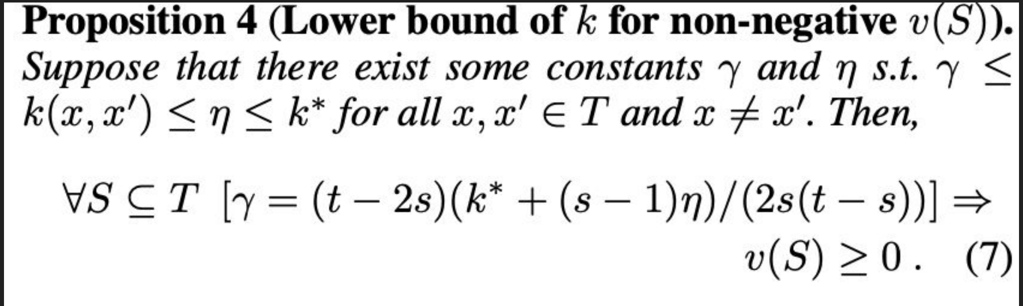


terval. We compute the probability of each synthetic data point x being sampled using the softmax function: $p(x) = \exp(\beta\bar{\Delta}_x)/\sum_{x'\in G\setminus G_i}\exp(\beta\bar{\Delta}_{x'})$ where $\beta \in [0,\infty)$ is the inverse temperature hyperparameter. Finally, we sample x

Varying β from 0 where data points have smaller Δx hence large number of samples which in turn increase the MMD to β = infinity which gives large Δx hence small number of samples. B is trade off between number of samples and how close it is to $D_i \cup G_i$

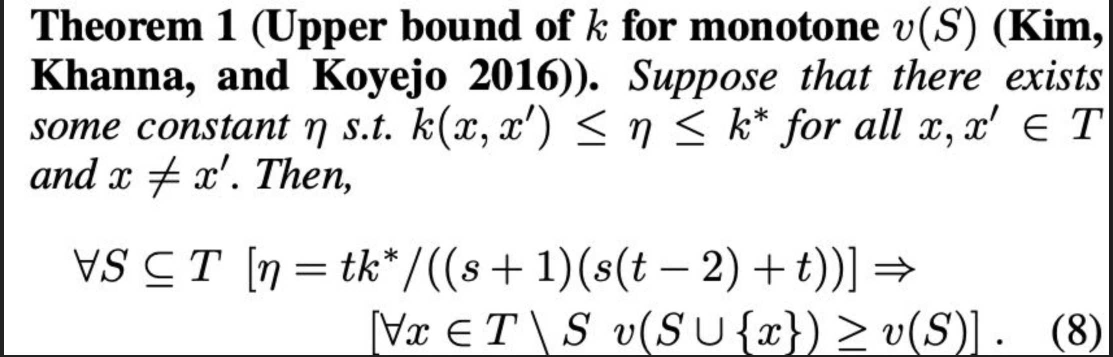
Kernel Selection

Upper And Lower Bound



Proposition 4 (Lower bound of k for non-negative $v(S)$). *Suppose that there exist some constants γ and η s.t. $\gamma \leq k(x, x') \leq \eta \leq k^*$ for all $x, x' \in T$ and $x \neq x'$. Then,*

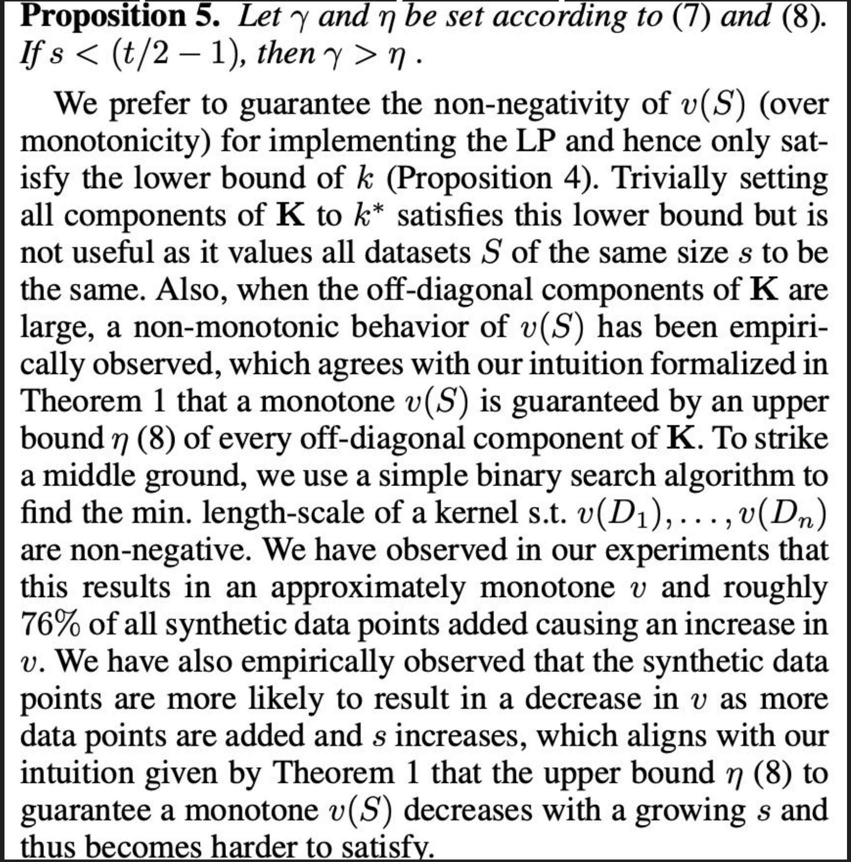
$$\forall S \subseteq T \;\; [\gamma = (t-2s)(k^* + (s-1)\eta)/(2s(t-s))] \Rightarrow v(S) \geq 0 \,. \quad (7)$$



Theorem 1 (Upper bound of k for monotone $v(S)$ (Kim, Khanna, and Koyejo 2016)). *Suppose that there exists some constant η s.t. $k(x, x') \leq \eta \leq k^*$ for all $x, x' \in T$ and $x \neq x'$. Then,*

$$\forall S \subseteq T \;\; [\eta = tk^*/((s+1)(s(t-2)+t))] \Rightarrow [\forall x \in T \setminus S \;\; v(S \cup \{x\}) \geq v(S)] \,. \quad (8)$$

Lower and Upper bound don’t exist at the same time,lower bound is preferred(Non negativity over monotonicity)



Proposition 5. *Let γ and η be set according to* (7) *and* (8). *If $s < (t/2 - 1)$, then $\gamma > \eta$.*

We prefer to guarantee the non-negativity of $v(S)$ (over monotonicity) for implementing the LP and hence only satisfy the lower bound of k (Proposition 4). Trivially setting all components of $\mathbf{K}$ to k^* satisfies this lower bound but is not useful as it values all datasets S of the same size s to be the same. Also, when the off-diagonal components of $\mathbf{K}$ are large, a non-monotonic behavior of $v(S)$ has been empirically observed, which agrees with our intuition formalized in Theorem 1 that a monotone $v(S)$ is guaranteed by an upper bound η (8) of every off-diagonal component of $\mathbf{K}$. To strike a middle ground, we use a simple binary search algorithm to find the min. length-scale of a kernel s.t. $v(D_1), \ldots, v(D_n)$ are non-negative. We have observed in our experiments that this results in an approximately monotone v and roughly 76% of all synthetic data points added causing an increase in v. We have also empirically observed that the synthetic data points are more likely to result in a decrease in v as more data points are added and s increases, which aligns with our intuition given by Theorem 1 that the upper bound η (8) to guarantee a monotone $v(S)$ decreases with a growing s and thus becomes harder to satisfy.

Advantages

1. A user can use the received rewards on a different type of kernel/learning algorithm and limits each party's flexibility to experiment with different model parameters and architectures.
2. Can perform a different learning task on the same dataset as the reward is a trained model
3. No data privacy issue as synthetic data is rewarded

Disadvantages

1. Struggles to achieve non negativity and monotonicity at the same time.
2. Has a tradeoff between high number of between high number of synthetic data points and low MMD with desired data set.