Learning Cutset Networks by Integrating Data and Noisy, Local Estimates

Shasha Jin[1] **Vasundhara Komaragiri**[1] **Tahrima Rahman**[1] **Vibhav Gogate**[1]

[1]Computer Science Department, The University of Texas at Dallas

Abstract

We consider the following parameter learning task in cutset networks (CNs): given (1) fully observed data, (2) a large number of marginal probability distributions, each defined over a small subset of variables, and (3) a CN structure, find a setting of parameters such that the resulting CN efficiently integrates the statistical information present in both the data and marginal distributions. In many real-world applications, the marginal distributions are either available from experts or via external processes and are typically inconsistent in that they do not come from the same joint probability distribution. In order to filter the inconsistency, we propose to approximate the learning objective using a convex combination of two quantities: one that enforces closeness via KL divergence to the marginal distributions and another that enforces closeness to a CN learned from data. We develop a gradient-based algorithm for minimizing the above objective and show that although the gradients are NP-hard to compute on Bayesian and Markov networks, they can be efficiently computed over CNs yielding a polynomial time algorithm with convergence guarantees. We show via experiments that our approach yields tractable models that are significantly superior to the ones learned from data alone even when the marginal distributions exhibit a high degree of inconsistency.

1 INTRODUCTION

To date, algorithms for learning the structure and parameters of cutset networks (CNs), and other similar tractable probabilistic models (TPMs) such as sum-product networks Poon and Domingos [2011] and probabilistic sentential decision diagrams Liang et al. [2017] assume access to either full data which has no missing values or almost full data in which only a few variables have missing values. Often in the real world, however, the following scenario is quite common. The learning algorithm has access to a small amount of almost full or full data and a large number of local marginal statistics that are derived from some combination of noisy local data, imperfect domain knowledge, and local, independent predictive models. For instance: **(1)** Due to privacy concerns in application domains such as social networks Degenne and Forsé [1999], Scott [1988], only limited global data is available. But local statistics, such as information about your contacts/connections can be retrieved easily. **(2)** In lazy learning of generative models Rahman et al. [2019a], Zheng and Webb [2000], Zhang and Zhou [2007], we derive sufficient statistics needed for inducing a probabilistic model at test time (when a query is made) from various sources such as a local classifier for each statistic or by querying a large database. **(3)** In active learning Settles [2012], the learning algorithm interactively solicits a user for labels or in general local marginal probability distributions for certain variables that the user is an expert at given observations.

A hallmark of all the scenarios just outlined is that the local marginal statistics are often *inconsistent*, namely, they may not come from a (unique) joint probability distribution. For example, in the active learning case, consider two pairwise marginal distributions $P_A(x_1, x_2)$ and $P_B(x_2, x_3)$ over three random variables X_1, X_2, X_3 derived by querying two users A and B respectively. A joint probability distribution $P(x_1, x_2, x_3)$ that is consistent with the two marginal distributions exists iff $P_A(x_2) = P_B(x_2)$ where $P_A(x_2) = \sum_{x_1} P_A(x_1, x_2)$ and $P_B(x_2) = \sum_{x_3} P_B(x_2, x_3)$. Unfortunately, because of precision errors and user bias, $P_A(x_2)$ will often not equal $P_B(x_2)$.

In this paper, we focus on learning the parameters of cutset networks (CNs) given both data and such local, noisy statistics. Our learner takes a CN that is initially learned from a small amount of data with few or no missing variables as input and then given a set of statistics, iteratively updates its parameters such that the linear combination of the fol-

Accepted for the 5th Workshop on Tractable Probabilistic Modeling at UAI (TPM 2022).

lowing two quantities is minimized: 1) the distance between the distribution represented by the original parameters and the one represented by the updated parameters, and 2) the sum of the distances between the given local estimates and the ones computed using the updated distribution. We derive a gradient-based method for solving this optimization task. Since the gradients require computation of marginal probabilities over subsets of variables, they are NP-hard in general on arbitrary probabilistic models but can be computed efficiently on CNs Vergari et al. [2021]. This shows the virtue of using tractable models in our learning settings. Our empirical results on benchmark datasets show that our approach that leverages local estimates yields significant improvements in both generative and predictive performance over the original model learned from data alone, even when the local estimates are inconsistent. Moreover, since the optimization problem is smooth, our procedure is guaranteed to reach a local optimum under mild conditions.

Related Work. Vomlel [1999, 2004] studied the problem of integrating probabilistic knowledge bases where a joint probability distribution is constructed from low-dimensional probability distributions (local estimates). Vomlel used a classic optimization method called iterative proportional fitting procedure (IPFP) Deming and Stephan [1940] and proposed a variant called the generalized expectation maximization algorithm (GEMA) for solving this problem. Vomlel provided convergence proofs for these methods; showing that IPFP converges when the local estimates are consistent and GEMA converges even if the local estimates are inconsistent. Unfortunately, Vomlel's approach has high computational complexity (is exponential in the treewidth of the graph defined over the local estimates) and is not practical. The method proposed in this paper does not have this limitation. Peng and Ding Peng and Ding [2012] proposed two polynomial time approximations for IPFP and applied them to Bayesian networks. However, their preliminary experimental study demonstrates that the error due to their approximations is quite high and convergence is not guaranteed if the local estimates are inconsistent. In contrast, our proposed method makes very few approximations and leverages tractable inference to yield a practical scheme.

2 OUR APPROACH

2.1 THE LEARNING PROBLEM AND ITS RELAXATION

We begin by describing the required notation. Upper-case letters (e.g., X, Y, U, etc.) and lower-case letters (e.g., x, y, u, etc.) are used to denote discrete random variables and their assignments respectively while bold upper-case letters (e.g., $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{U}$, etc.) and bold lower-case letters (e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{u}$, etc.) are respectively used to denote sets of discrete random variables and assignments to them. For simplicity of exposition, we assume that all variables are binary taking values from the set $\{0, 1\}$.

We focus on learning the parameters of cutset networks Rahman and Gogate [2016c]. See the supplement for a brief introduction to these tractable probabilistic models. At a high level, a CN is an AND/OR graph Mateescu et al. [2008] with a tree Bayesian network attached to each leaf node of the AND/OR graph. A CN is parameterized by conditional probabilities such that each probability is either attached to an edge from an OR node to an AND node in the AND/OR graph or associated with a tree Bayesian network at a leaf node. More formally, let Θ denote the set of parameters of a CN, such that $\theta_{x_i \boldsymbol{u_i}} \in \Theta$ is equal to the conditional probability $P(X_i = 1|\boldsymbol{U_i} = \boldsymbol{u_i})$.

We assume that we have access to local information that can be summarized via pairwise distributions. Note that our algorithm can be easily extended to arbitrary local marginal distributions and we make the assumption for clarity of presentation only. Let D denote the KL divergence (distance) between two probability distributions defined over the same set of variables and $\mathcal{E}$ denote a subset of pairs of random variables, namely $\mathcal{E} \subseteq \{(X_j, X_k)|X_j, X_k \in \boldsymbol{X}$ and $j < k\}$. Given this notation and assumptions, our learning problem can be stated as:

Given: A cutset network structure with parameters Θ representing a probability distribution $\mathcal{R}$, a set of pairwise local distributions $\mathcal{P}_{jk}(X_j, X_k)$ where $(X_j, X_k) \in \mathcal{E}$ and a fully observed dataset $\mathcal{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$.
To do: Find an assignment of values to all parameters in Θ such that the negative log-likelihood of $\mathcal{X}$ w.r.t. $\mathcal{R}_\Theta$ is minimized and the KL divergence between $\mathcal{P}_{jk}(X_j, X_k)$ and $\mathcal{R}_\Theta(X_j, X_k)$ equals zero for all $(X_j, X_k) \in \mathcal{E}$.

Mathematically, we can express it as:

$$\begin{gathered}\underset{\boldsymbol{\Theta}}{\operatorname{argmin}} \; - \sum_{\boldsymbol{x}^{(d)} \in \mathcal{X}} \log \mathcal{R}_\Theta(\boldsymbol{x}^{(d)}) \; s.t. \\ \forall \; (X_j, X_k) \in \mathcal{E}, \; D(\mathcal{P}_{jk}(x_j, x_k), R_\Theta(x_j, x_k)) = 0\end{gathered} \tag{1}$$

Unfortunately, if the set $\{\mathcal{P}_{jk}|(X_j, X_k) \in \mathcal{E}\}$ is *inconsistent*, namely there does not exist a joint probability distribution $\mathcal{P}$ such that for all $(X_j, X_k) \in \mathcal{E}, \mathcal{P}_{jk}(X_j, X_k)$ is a marginal probability distribution of $\mathcal{P}$ then the above constrained optimization problem is infeasible. To address this issue, we propose to use a linear combination of the objective and constraints (which is akin to Lagrange relaxation). Further, by negating the objective and expanding using the definition of KL divergence, we get

$$\begin{gathered}\underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \; \lambda_1 \sum_{(X_j, X_k) \in \mathcal{E}} \sum_{x_j, x_k} \mathcal{P}_{jk}(x_j, x_k) \log \mathcal{R}_\Theta(x_j, x_k) \\ +\lambda_2 \left(\sum_{\boldsymbol{x}^{(d)} \in \mathcal{X}} \log \mathcal{R}_\Theta(\boldsymbol{x}^{(d)}) \right)\end{gathered} \tag{2}$$

where $\lambda_1 \geq 0$ ad $\lambda_2 \geq 0$ are hyperparameters (technically, we only need one hyperparameter; we use two for convenience) that model the relative importance of the local and global statistics respectively.

The optimization problem given in Eq. (2) is not concave in the parameters Θ but smooth and therefore can be solved using an iterative, gradient ascent algorithm. However, to handle the gradient of the second term efficiently, we propose the following moment-matching approach.

Let $\mathcal{Q}$ denote the distribution associated with the cutset network having the same structure as $\mathcal{R}$. Thus, there is a one-to-one correspondence between the parameters of $\mathcal{Q}$ and $\mathcal{R}$. Let Π denote the set of parameters of $\mathcal{Q}$. Thus, given a parameter $\pi_{x_i,\boldsymbol{u}_i} \in \Pi$, there is a corresponding parameter $\theta_{x_i,\boldsymbol{u}_i}$ in Θ. Let the set of parameters Π be learned from data $\mathcal{X}$ by maximizing the log-likelihood. Since the parameters of cutset networks are conditional probability distributions, given $\mathcal{Q}$ learned from data, we can use negative cross-entropy between parameters of $\mathcal{Q}$ and $\mathcal{R}$ in lieu of the second term (log-likelihood) of Eq. (2). Mathematically,

$$\underset{\boldsymbol{\Theta}}{\operatorname{argmax}} \ \lambda_1 \sum_{(X_j,X_k)\in\mathcal{E}} \sum_{x_j,x_k} \mathcal{P}_{jk}(x_j,x_k) \log \mathcal{R}_\Theta(x_j,x_k) + \\ \lambda_2 \sum_{\theta_{x_i,\boldsymbol{u}_i}\in\Theta} \pi_{x_i,\boldsymbol{u}_i} \log \theta_{x_i,\boldsymbol{u}_i} + (1-\pi_{x_i,\boldsymbol{u}_i}) \log(1-\theta_{x_i,\boldsymbol{u}_i}) \tag{3}$$

2.2 SOLVING THE LEARNING PROBLEM VIA GRADIENT ASCENT

In this section, we derive the gradients w.r.t. each parameter $\theta_{x_i,\boldsymbol{u}_i}$. The partial derivative of the second term w.r.t. $\theta_{x_i,\boldsymbol{u}_i}$ is straight-forward and given by

$$\lambda_2 \left(\frac{\pi_{x_i,\boldsymbol{u}_i}}{\theta_{x_i,\boldsymbol{u}_i}} - \frac{1-\pi_{x_i,\boldsymbol{u}_i}}{1-\theta_{x_i,\boldsymbol{u}_i}} \right) \tag{4}$$

The partial derivative of the first term in Eq. (3) is more involved and we summarize it in the following proposition. (Proofs are given in the appendix.)

Proposition 1. *The partial derivative of*

$$\lambda_1 \sum_{(X_j,X_k)\in\mathcal{E}} \sum_{x_j,x_k} \mathcal{P}_{jk}(x_j,x_k) \log \mathcal{R}_\Theta(x_j,x_k)$$

w.r.t. $\theta_{x_i,\boldsymbol{u}_i}$ *is given by*

$$\lambda_1 \sum_{(X_j,X_k)\in\mathcal{E}} \sum_{x_j,x_k} \mathcal{P}_{jk}(x_j,x_k) \left(\frac{\mathcal{R}_\Theta(\boldsymbol{u}_i, X_i=1|x_j,x_k)}{\theta_{x_i,\boldsymbol{u}_i}} \right. \\ \left. - \frac{\mathcal{R}_\Theta(\boldsymbol{u}_i, X_i=0|x_j,x_k)}{1-\theta_{x_i,\boldsymbol{u}_i}} \right) \tag{5}$$

With the above derivatives, we formally present our algorithm for learning cutset networks with local inconsistent statistics or LCN-LIS (see Algorithm 1) in the supplement.

The main virtue of LCN-LIS is that it has polynomial computational complexity: the time (and space) complexity is $O(|\mathcal{E}| \times |\Theta| \times T)$ where $|\mathcal{E}|$ denotes the number of pairwise (local) probability distributions provided to the algorithm as input, $|\Theta|$ the number of parameters and T the bound on the number of iterations.

Remarks. Note that a feasible solution to the optimization problem given in Eq. (3), and thus the one returned by Algorithm 1 *filters inconsistency* in the local estimates $\{\mathcal{P}_{jk}|(X_j,X_k) \in \mathcal{E}\}$ because it yields a globally consistent model $\mathcal{R}_\Theta$. Unlike previously proposed techniques for solving the optimization task in Eq. (1) such as the iterative proportional fitting procedure (IPFP) Deming and Stephan [1940], Vomlel [2004] which will not converge when the local estimates are inconsistent, Algorithm 1 will converge to a local optimum (because the objective is smooth).

To summarize, we derived and presented a gradient-based algorithm for learning the parameters of cutset networks in the presence of local estimates (see Algorithm 1) and showed that the algorithm requires time and space that scales linearly with the number of given local statistics. Note that Proposition 1 can be easily extended to Bayesian and Markov networks. However, the problem is that computing the terms in the numerator in Eq. 5 will be NP-hard in general on these models. This highlights another virtue of tractable models, local information, even if it is inconsistent can be efficiently integrated into a tractable model.

3 EXPERIMENTS

We performed a detailed, controlled experimental study to evaluate the impact of using inconsistent local statistics on the quality of the learned model. Specifically, we used the following controls: (1) the accuracy of the cutset network $\mathcal{Q}$ learned from data $\mathcal{X}$ in Algorithm 1; (2) the strength/level of inconsistency in the local statistics $\mathcal{P}_{jk}(X_j,X_k)$; (3) the cutset network architecture; and (4) the number of evidence variables or observations available at test time (to test discriminative performance). We used 20 benchmark datasets that have been widely used in previous studies Rooshenas and Lowd [2014, 2013] to evaluate our new approach.

We used *negative cross-entropy* between the true $\mathcal{P}$ and the learned model (under various conditions) as our evaluation criteria. The higher the negative cross-entropy, the better the model. For each dataset, we learned a mixtures of cutset networks Rahman et al. [2014] and used it as the true model $\mathcal{P}$. We used 10% of randomly chosen examples in the training set to learn $\mathcal{Q}$. Thus, the dataset used by Algorithm 1 has 90% fewer examples than the one used for learning $\mathcal{P}$. We did this to ensure that the true model differs significantly

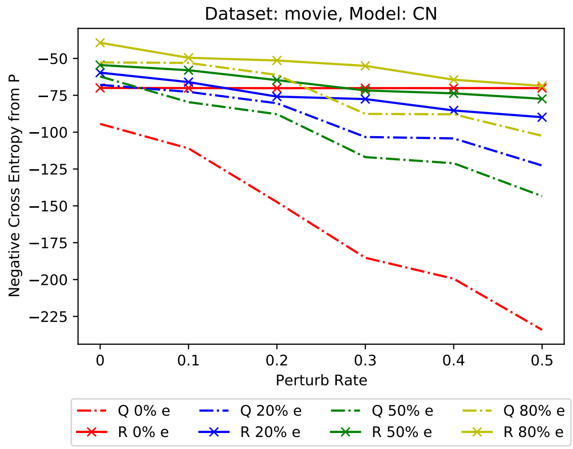


Figure 1: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50%, and 80% on CN, as a function of perturb rate for the "Movie" dataset

Table 1: Generative (0% evidence) performance measured using negative cross entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$ on three different models: CLTs, CNs, MCNs.

Datasets	**#var**	Negative cross-entropy with **0% Evidence**					
		CLTs		**CNs**		**MCNs**	
		$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$
nltcs	16	-7.62	**-6.85**	-6.86	**-6.17**	-7.32	**-6.03**
plants	69	-18.64	**-17.23**	-17.70	**-15.55**	-17.04	**-15.05**
accidents	111	-38.00	**-34.21**	-37.06	**-33.14**	-35.72	**-32.43**
kosarek	190	-21.04	**-12.98**	-19.46	**-12.45**	-18.23	**-10.68**
msweb	294	-24.57	**-12.28**	-24.37	**-12.33**	-22.48	**-11.05**
book	500	-58.07	**-41.80**	-56.28	**-41.46**	-51.81	**-34.49**
webkb	839	-207.12	**-172.65**	-199.76	**-162.30**	-181.05	**-126.06**
20newsg	910	-194.55	**-157.70**	-181.47	**-155.48**	-177.83	**-152.65**
bbc	1058	-325.51	**-270.74**	-265.46	**-210.53**	-243.87	**-223.60**
ad	1556	-112.01	**-47.27**	-93.96	**-41.34**	-96.79	**-53.70**
Total AVG		-100.71	**-77.37**	-90.22	**-69.06**	-85.21	**-66.57**

from $\mathcal{Q}$, which in turn will help us evaluate how local information improves the quality of an inferior starting point. We further controlled the quality of $\mathcal{Q}$ using a parameter h which we call the *perturb rate*. h lies between 0 and 100, and given a value for h, we replaced $h\%$ of the parameters in $\mathcal{Q}$ with a random number. We normalized $\mathcal{Q}$ to ensure that it is a valid probability distribution.

We used three types of cutset network structures to learn $\mathcal{Q}$: (1) cutset networks with depth 0 which are equivalent to Chow-Liu trees Chow and Liu [1968](CLTs); (2) cutset networks without any latent variables (CNs); and (3) mixtures of cutset networks (MCNs). The latter is a state-of-the-art model Rahman et al. [2019b]. We learned both discriminative and generative cutset networks. In discriminative networks, we set L% of random variables as evidence $\boldsymbol{E}$ and learn a probability distribution over the variables $\boldsymbol{X} \setminus \boldsymbol{E}$ given an assignment $\boldsymbol{e}$ to the evidence variables. We used 4 values for L: 0, 20, 50, and 80. When $L = 0$, we get a generative model while the remaining models are discriminative. We generated local statistics from $\mathcal{P}$ as follows. Since $\mathcal{P}$ is a tractable model, we can efficiently (in linear time) compute $\mathcal{P}_{jk}(X_j, X_k)$ for all $X_j, X_k \in \boldsymbol{X}$. To make them inconsistent, we added a value ϵ that is randomly sampled from a normal distribution with 0 mean and standard deviation σ. We experimented with five values of σ: 0.001, 0.01, 0.05, 0.1, 0.2. Note that after adding ϵ, we have to normalize the distributions to ensure that they are valid.

Results. Table 1 shows the generative (0% evidence) performance of $\mathcal{Q}$ and $\mathcal{R}$ respectively on 10 datasets (see Tables 2, 3, 4, and 5 for more results). We used $\sigma = 0.1$ and $h = 0$ to generate the experimental data given in the tables. Each number in each table is an average over 5 runs. To avoid clutter, we do not report the standard deviation because it was fairly small over all runs. These results help us analyze the impact of the cutset network architecture and number of evidence variables on our evaluation criteria. We observe that on average, using local inconsistent statistics improves the negative cross entropy of each architecture by 17-23%. MCNs are the best performing model overall and Chow-Liu trees (CLTs) are significantly worse. There was no significant difference in the amount of improvement as we increased the number of evidence nodes. This suggests that our approach is equally useful for both discriminative and generative models.

Figure 1 presents the negative cross-entropy scores achieved by a CN trained on the movie dataset as a function of the perturb rate (h) .We present the plots for the remaining datasets in the supplementary material. These plots help us evaluate the impact that the quality of $\mathcal{Q}$ has (namely the impact of the starting point) on the model learned by Algorithm 1. We observe that as the perturb rate increases, there is a substantial drop in the negative cross-entropy between $\mathcal{P}$ and $\mathcal{Q}$. However, the negative cross entropy of $\mathcal{P}$ and $\mathcal{R}$ remains relatively flat. This shows that the use of local statistics significantly improves the model quality, especially when the model based on global data is inaccurate.

4 CONCLUSION

In this paper, we presented a new method for learning the parameters of cutset networks in the presence of noisy local estimates. Unlike conventional algorithms which use full i.i.d. data during the learning process, we proposed a novel approach that uses noisy local information to learn a more accurate and robust model. The key advantage of using local estimates is that they are often readily available as compared to full i.i.d. data. We also showed via experiments on benchmark datasets that our new algorithm greatly improves the quality of the initial model learned from i.i.d. data, even when the local estimates are inconsistent and noisy.

Acknowledgements

This work was supported in part by the DARPA Explainable Artificial Intelligence (XAI) Program under contract number N66001-17-2-4032, by the DARPA Perceptually-enabled Task Guidance (PTG) Program under contract number HR00112220005 and by the National Science Foundation CAREER award IIS-1652835.

References

C. K. Chow and C. N Liu. Approximating discrete probability distributions with dependence trees. *IEEE*, 14: 462–467, 1968.

A. Darwiche. A Differential Approach to Inference in Bayesian Networks. *Journal of the ACM*, 50:280–305, 2003.

R. Dechter and R. Mateescu. AND/OR Search Spaces for Graphical Models. *Artificial Intelligence*, 171(2-3):73–106, 2007.

Alain Degenne and Michel Forsé. *Introducing social networks*. Sage, 1999.

W. E. Deming and F. F. Stephan. On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known. *The Annals of Mathematical Statistics*, 11(4):427 – 444, 1940.

Y. Liang, J. Bekker, and Guy Van den Broeck. Learning the structure of probabilistic sentential decision diagrams. In *Uncertainty in Artificial Intelligence*, 2017.

R. Mateescu, R. Dechter, and R. Marinescu. AND/OR Multi-Valued Decision Diagrams (AOMDDs) for Graphical Models. *Journal of Artificial Intelligence Research*, 33:465–519, 2008.

Yun Peng and Zhongli Ding. Modifying bayesian networks by probability constraints. *CoRR*, abs/1207.1356, 2012.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *ICCV Workshops*, pages 689–690. IEEE, 2011.

T. Rahman and V. Gogate. Learning ensembles of cutset networks. In *AAAI conference on Artificial Intelligence*, pages 3301–3307, 2016a.

T. Rahman and V. Gogate. Merging strategies for sum-product networks: From trees to graphs. In *Proceedings of the Thirty-Second Conference Conference on Uncertainty in Artificial Intelligence*, pages 617–626, 2016b.

T. Rahman and V. Gogate. Merging Strategies for Sum-Product Networks: From Trees to Graphs. In *Proceedings of the Thirty-Second Conference Conference on Uncertainty in Artificial Intelligence*, pages 617–626, 2016c.

T. Rahman, P. Kothalkar, and V. Gogate. Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of chow-liu trees. In *Proceedings of the Twenty-Fifth European Conference on Machine Learning*, pages 630–645, 2014.

T. Rahman, S. Jin, and V. Gogate. Cutset Bayesian Networks: A New Representation for Learning Rao-Blackwellised Graphical Models. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5751–5757, 2019a.

T. Rahman, S. Jin, and V. Gogate. Look Ma, No Latent Variables: Accurate Cutset Networks via Compilation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the Thirty-Sixth International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5311–5320. PMLR, 2019b.

A. Rooshenas and D. Lowd. Learning sum-product networks with direct and indirect variable interactions. In *Proceedings of the Thirty-First International Conference on Machine Learning*, pages 710–718, 2014.

Amirmohammad Rooshenas and Daniel Lowd. Learning tractable graphical models using mixture of arithmetic circuits. In *AAAI*, 2013.

John Scott. Social network analysis. *Sociology*, 22(1):109–127, 1988.

B. Settles. *Active Learning*. Morgan & Claypool Publishers, 2012. ISBN 1608457257.

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, dec 2021.

J. Vomlel. *Methods of Probabilistic Knowledge Integration*. Phd thesis, Czech Technical University, 1999.

J. Vomlel. Integrating inconsistent data in a probabilistic model. *Journal of Applied Non-Classical Logics*, 14(3): 367–386, 2004.

Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

Zijian Zheng and Geoffrey I Webb. Lazy learning of bayesian rules. *Machine Learning*, 41(1):53–84, 2000.

Supplement

Shasha Jin[1] **Vasundhara Komaragiri**[1] **Tahrima Rahman**[1] **Vibhav Gogate**[1]

[1]Computer Science Department, The University of Texas at Dallas

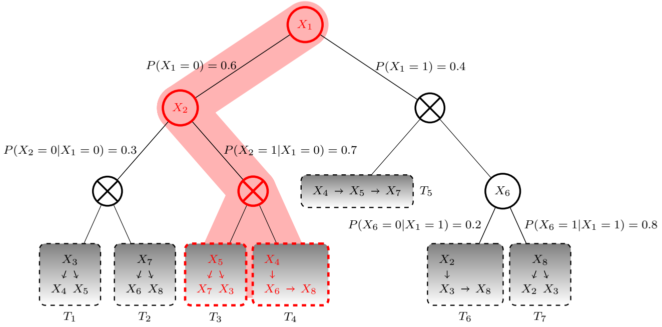


Figure 2: A cutset network defined over variables $\{X_1, \ldots, X_8\}$. OR nodes are denoted with variables in circles, AND nodes by cross marked circles and leaf nodes (tree Bayesian networks) by shadowed dotted rectangles. Arcs emanating from an OR node are labeled with conditional probabilities. The highlighted paths show the sub-tree consistent with the assignment $\{X_1 = 0, X_2 = 1, X_3 = \cdots = X_8 = 0\}$.

A CUTSET NETWORKS

Cutset networks (CNs) Rahman and Gogate [2016b] are tractable probabilistic models that combine two well-known classes of tractable models: AND/OR graphs Dechter and Mateescu [2007] and tree Bayesian networks. Formally, a CN over a set of variables $\boldsymbol{X}$ ($\boldsymbol{X}$ may include latent variables) is defined recursively using the following three conditions: (1) A tree Bayesian network over $\boldsymbol{X}$ is a CN; (2) An OR node labeled by a variable $X_i \in \boldsymbol{X}$ such that $|\boldsymbol{X}| > 1$ with two child CNs, each defined over the set $\boldsymbol{X} \setminus \{X_i\}$ is a CN. We follow the convention that the left child of the OR node labeled by X_i represents conditioning over $X_i = 0$ and the right child represents conditioning over $X_i = 1$. The arcs from the OR node to its child nodes are labeled with probability values in $\mathbb{R}^+$ such that they sum to 1; and (3) Let $(\boldsymbol{X}_1, \boldsymbol{X}_2)$ be a partition of $\boldsymbol{X}$ such that $|\boldsymbol{X}| > 1$. Then, an AND node with two child CNs, one defined over $\boldsymbol{X}_1$ and the second defined over $\boldsymbol{X}_2$ is a CN. Fig. 2 shows a CN defined over eight variables. In general, a full assignment $\boldsymbol{x}$ yields a rooted sub-graph in a CN.

Following our notation for Bayesian networks, let Θ denote the set of parameters of the cutset network, such that $\theta_{x_i \boldsymbol{u}_i} \in \Theta$ is equal to the conditional probability $P(X_i = 1|\boldsymbol{U}_i = \boldsymbol{u}_i)$. Given an assignment $\boldsymbol{x}$, when X_i is an OR node, $\boldsymbol{u}_i$ denotes the assignment along the path from the root of the CN to X_i. Alternatively, when X_i belongs to a tree Bayesian network in the CN, $\boldsymbol{u}_i$ denotes the assignment formed by composing the assignment along the path from the root of the CN to the tree Bayesian network with the assignment to the parent of X_i (if X_i has a parent in the network). Given this parameterization, the probability distribution associated with the cutset network is given by

$$\Pr(\boldsymbol{x}) = \prod_{\substack{i=1 \\ \boldsymbol{u}_i \sim \boldsymbol{x}}}^{n} (\theta_{x_i \boldsymbol{u}_i})^{x_i} (1 - \theta_{x_i \boldsymbol{u}_i})^{1 - x_i} \tag{6}$$

A distinguishing feature of CNs is that when they have no latent variables, both MAR and MPE inference can be performed over them in time that scales linearly with the size of the network. This can be accomplished by converting the CN in linear time to an AND/OR graph Dechter and Mateescu [2007] or an arithmetic circuit Darwiche [2003] and performing two-passes over these circuits. When latent variables are present, CNs admit linear time MAR inference only while MPE inference is intractable in general.

B PROOF OF PROPOSITION 1

To prove Proposition 1, we derive the partial derivatives step by step.

Proof.

$$\frac{\partial \log \mathcal{R}_\theta(x_j, x_k)}{\partial \theta_{x_i, \boldsymbol{u}_i}} = \frac{1}{\mathcal{R}_\theta(x_j, x_k)} \frac{\partial \mathcal{R}_\theta(x_j, x_k)}{\partial \theta_{x_i, \boldsymbol{u}_i}} \tag{7}$$

Accepted for the 5th Workshop on Tractable Probabilistic Modeling at UAI (TPM 2022).

Using the sum rule of probability theory, $\mathcal{R}_\theta(x_j, x_k)$ equals

$$\begin{aligned}
&= \sum_{x_i,\boldsymbol{u}_i} \mathcal{R}_\theta(x_j, x_k, \boldsymbol{u}_i, x_i) \\
&= \sum_{x_i,\boldsymbol{u}_i} \mathcal{R}_\theta(\boldsymbol{u}_i, x_i)\mathcal{R}_\theta(x_j, x_k, |\boldsymbol{u}_i, x_i) \qquad (8) \\
&= \sum_{x_i,\boldsymbol{u}_i} \mathcal{R}_\theta(x_i|\boldsymbol{u}_i)\mathcal{R}_\theta(\boldsymbol{u}_i)\mathcal{R}_\theta(x_j, x_k|\boldsymbol{u}_i, x_i) \qquad (9)
\end{aligned}$$

Since $\mathcal{R}_\theta(x_i|\boldsymbol{u}_i) = \theta_{x_i,\boldsymbol{u}_i}$ if $X_i = 1$ and $\mathcal{R}_\theta(x_i|\boldsymbol{u}_i) = 1 - \theta_{x_i,\boldsymbol{u}_i}$ if $X_i = 0$, we have:

$$\begin{aligned}
\mathcal{R}_\theta(x_j, x_k) = \sum_{\boldsymbol{u}_i} \theta_{x_i,\boldsymbol{u}_i}\mathcal{R}_\theta(\boldsymbol{u}_i)\mathcal{R}_\theta(x_j, x_k|\boldsymbol{u}_i, X_i = 1) \\
+ \sum_{\boldsymbol{u}_i}(1 - \theta_{x_i,\boldsymbol{u}_i})\mathcal{R}_\theta(\boldsymbol{u}_i)\mathcal{R}_\theta(x_j, x_k|\boldsymbol{u}_i, X_i = 0)
\end{aligned} \qquad (10)$$

Differentiating the Equation above (Eq. (10)) w.r.t. $\theta_{x_i,\boldsymbol{u}_i}$, we get:

$$\begin{aligned}
\frac{\partial \mathcal{R}_\theta(x_j, x_k)}{\partial \theta_{x_i,\boldsymbol{u}_i}} = \mathcal{R}_\theta(\boldsymbol{u}_i)\mathcal{R}_\theta(x_j, x_k|\boldsymbol{u}_i, X_i = 1) \\
- \mathcal{R}_\theta(\boldsymbol{u}_i)\mathcal{R}_\theta(x_j, x_k|\boldsymbol{u}_i, X_i = 0)
\end{aligned} \qquad (11)$$

Since

$$\mathcal{R}_\theta(\boldsymbol{u}_i)\mathcal{R}_\theta(x_j, x_k|\boldsymbol{u}_i, X_i = 1) = \frac{\mathcal{R}_\theta(x_j, x_k, \boldsymbol{u}_i, X_i = 1)}{\theta_{x_i,\boldsymbol{u}_i}} \qquad (12)$$

and

$$\mathcal{R}_\theta(\boldsymbol{u}_i)\mathcal{R}_\theta(x_j, x_k|\boldsymbol{u}_i, X_i = 0) = \frac{\mathcal{R}_\theta(x_j, x_k, \boldsymbol{u}_i, X_i = 0)}{1 - \theta_{x_i,\boldsymbol{u}_i}} \qquad (13)$$

we can rewrite Eq. (11) as:

$$\begin{aligned}
\frac{\partial \mathcal{R}_\theta(x_j, x_k)}{\partial \theta_{x_i,\boldsymbol{u}_i}} = \frac{\mathcal{R}_\theta(x_j, x_k, \boldsymbol{u}_i, X_i = 1)}{\theta_{x_i,\boldsymbol{u}_i}} \\
- \frac{\mathcal{R}_\theta(x_j, x_k, \boldsymbol{u}_i, X_i = 0)}{1 - \theta_{x_i,\boldsymbol{u}_i}}
\end{aligned} \qquad (14)$$

Substituting Eq. (14) in Eq. (7) and using the definition of conditional probability, we get:

$$\begin{aligned}
\frac{\partial \log \mathcal{R}_\theta(x_j, x_k)}{\partial \theta_{x_i,\boldsymbol{u}_i}} = \frac{\mathcal{R}_\Theta(\boldsymbol{u}_i, X_i = 1|x_j, x_k)}{\theta_{x_i,\boldsymbol{u}_i}} - \\
\frac{\mathcal{R}_\Theta(\boldsymbol{u}_i, X_i = 0|x_j, x_k)}{1 - \theta_{x_i,\boldsymbol{u}_i}}
\end{aligned} \qquad (15)$$

□

C THE LEARNING ALGORITHM

We formally present an algorithm that leverages the gradient equations given in Eqs. (4)–(5) (see Algorithm 1) for solving the learning problem given in Eq. (3). The algorithm,

Algorithm 1: LCN-LIS ($\mathcal{X}$,$\mathcal{P}$, λ_1, λ_2, T)

```
Input  : (1) Training examples $\mathcal{X}$ defined over a set of
         variables $\boldsymbol{X}$; (2) a set of inconsistent pairwise
         marginal statistics $\mathcal{P}_{jk}(X_j, X_k)$ where
         $(X_j, X_k) \in \mathcal{E}$ and
         $\mathcal{E} \subseteq \{(X_j, X_k)|X_j, X_k \in \boldsymbol{X}$ and $j < k\}$; (3) Two
         hyper parameters $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ and (4) A
         bound $T$ on the number of iterations
Output : A Cutset Network $\mathcal{R}$
begin
    Learn a Cutset Network $\mathcal{Q}$ from Data $\mathcal{X}$ using learning
      algorithms from literature (cf. Rahman et al. [2014],
      Rahman and Gogate [2016b,a]).
    Initialize cutset network $\mathcal{R}$ to have the same structure as $\mathcal{Q}$.
      Let $\Theta = \{\theta_{x_i,\boldsymbol{u}_i}\}$ and $\Pi = \{\pi_{x_i,\boldsymbol{u}_i}\}$ denote the set of
      parameters of $\mathcal{R}$ and $\mathcal{Q}$ respectively.
    Initialize: all parameters $\theta_{x_i,\boldsymbol{u}_i}$ of $\mathcal{R}$ to a random number
      between 0 and 1. Let $\theta^0_{x_i,\boldsymbol{u}_i}$ denote the initial value of
      $\theta_{x_i,\boldsymbol{u}_i}$
    Initialize: $t = 0$
    repeat
        Initialize: $g_{x_i,\boldsymbol{u}_i} = 0$ for each $\theta_{x_i,\boldsymbol{u}_i} \in \Theta$
        for each possible value assignment $X_j = x_j$ and
          $X_k = x_k$ where $(X_j, X_k) \in \mathcal{E}$ do
            Set $X_j = x_j$ and $X_k = x_k$ as evidence in $\mathcal{R}$ and
              compute the conditional probabilities
              $\mathcal{R}_{\Theta^t}(\boldsymbol{u}_i, x_i|x_j, x_k)$ for each $\theta_{x_i,\boldsymbol{u}_i} \in \Theta$
            for each parameter $\theta_{x_i,\boldsymbol{u}_i} \in \Theta$ do
                Let:
                    $\delta^+ = \frac{\mathcal{R}_{\Theta^t}(\boldsymbol{u}_i, X_i = 1|x_j, x_k)}{\theta^t_{x_i,\boldsymbol{u}_i}}$
                    $\delta^- = \frac{\mathcal{R}_{\Theta^t}(\boldsymbol{u}_i, X_i = 0|x_j, x_k)}{(1 - \theta^t_{x_i,\boldsymbol{u}_i})}$
                Update:
                    $g_{x_i,\boldsymbol{u}_i} = g_{x_i,\boldsymbol{u}_i} + \lambda_1 \mathcal{P}_{jk}(x_j, x_k)(\delta^+ - \delta^-)$
                    $g_{x_i,\boldsymbol{u}_i} = g_{x_i,\boldsymbol{u}_i} + \lambda_2 \left(\frac{\pi_{x_i,\boldsymbol{u}_i}}{\theta^t_{x_i,\boldsymbol{u}_i}} - \frac{1 - \pi_{x_i,\boldsymbol{u}_i}}{1 - \theta^t_{x_i,\boldsymbol{u}_i}}\right)$
            end
        end
        for each parameter $\theta_{x_i,\boldsymbol{u}_i} \in \Theta$ do
            $\theta^{t+1}_{x_i,\boldsymbol{u}_i} = \theta^t_{x_i,\boldsymbol{u}_i} + \alpha \times g_{x_i,\boldsymbol{u}_i}$ // $\alpha$: learning rate
        end
        $t = t + 1$
    until convergence or t $\geq T$;
    return $\mathcal{R}$ with parameters $\Theta^t$
end
```

which we call learning cutset networks with local inconsistent statistics (LCN-LIS), takes as input training dataset $\mathcal{X}$, local statistics $P_{jk}(X_j, X_k)$, two hyperparameters λ_1 and λ_2 (real numbers) and an integer bound T on the number of iterations. It begins by learning a cutset network $\mathcal{Q}$ from the dataset $\mathcal{X}$ (step 2) and initializes $\mathcal{R}$ to have the same structure as $\mathcal{Q}$ (step 3). In steps 4–19, the algorithm runs the gradient ascent steps. The gradient ascent begins by setting all parameters to a random number between 0 and 1. Then, at each iteration t, for each pair $(X_j, X_k) \in \mathcal{E}$, it sets X_j and X_k as evidence and runs a two-pass inference algorithm over the cutset network Dechter and Mateescu [2007] to compute the required conditional probabilities $\mathcal{R}_{\Theta^t}(\boldsymbol{u}_i, X_i = 1|x_j, x_k)$ and $\mathcal{R}_{\Theta^t}(\boldsymbol{u}_i, X_i = 0|x_j, x_k)$ (step 9). The algorithm then updates the gradient for each parameter $\theta_{x_i,\boldsymbol{u}_i}$ (steps 10–13; also see Eqs. (4)–(5)) given the assignment $X_j = x_j$ and $X_k = x_k$. In steps 15–17, the algorithm updates the parameters using the gradient estimates $g_{x_i,\boldsymbol{u}_i}$ and learning rate α. The algorithm terminates the gradient ascent on convergence or when the bound T on the number of iterations is reached. At termination, the algorithm returns $\mathcal{R}$ with parameters Θ^t.

C.1 TIME AND SPACE COMPLEXITY OF ALGORITHM 1

Proof. The time (and space) complexity of step 9 is $O(|\Theta|)$ using a two-pass algorithm over the CN that calculates the conditional probabilities of the parameters given evidence (see Darwiche [2003], Dechter and Mateescu [2007]). The time complexity of updating the gradient (steps 10-13) is also $O(|\Theta|)$. Thus, the time complexity of steps 9–13 is $O(|\Theta|)$. Since these steps can be executed a maximum of $O(|\mathcal{E}| \times T)$ times (step 8 and step 6), the overall complexity is $O(|\mathcal{E}| \times T \times |\Theta|)$. □

D ADDITIONAL EXPERIMENTAL RESULTS

In this section, we present the full experimental results for all the datasets.

D.1 GENERATIVE AND DISCRIMINATIVE PERFORMANCE

In the main paper, we presented the generative performance of $\mathcal{Q}$ and $\mathcal{R}$ on the models CLTs, CNs, MCNs for a subset of the datasets. We present the results on all the 20 datasets in Table 2.

Further, we use Tables 3, 4 and 5 to illustrate the discriminative performance of $\mathcal{Q}$ and $\mathcal{R}$ on model CLTs, CNs, MCNs when the number of evidence variables is 20%, 50% and 80% respectively.

Table 2: Generative (0% evidence) performance measured using negative cross entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$ on three different models: CLTs, CNs, MCNs.

		Negative cross-entropy with **0% Evidence**					
Datasets	**#var**	**CLTs**		**CNs**		**MCNs**	
		$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$
nltcs	16	-7.62	**-6.85**	-6.86	**-6.17**	-7.32	**-6.03**
msnbc	17	-7.01	**-6.58**	-6.99	**-6.36**	-7.08	**-6.08**
kdd	64	-5.20	**-2.80**	-5.86	**-2.77**	-5.52	**-2.69**
plants	69	-18.64	**-17.23**	-17.70	**-15.55**	-17.04	**-15.05**
audio	100	-47.16	**-45.52**	-47.15	**-44.01**	-44.20	**-42.57**
jester	100	-60.95	**-59.62**	-61.08	**-57.45**	-57.92	**-56.70**
netflix	100	-62.98	**-61.46**	-63.85	**-60.93**	-58.31	**-57.73**
accidents	111	-38.00	**-34.21**	-37.06	**-33.14**	-35.72	**-32.43**
retail	135	-15.54	**-11.59**	-17.22	**-11.61**	-16.68	**-11.61**
pumsb*	163	-37.23	**-32.86**	-33.25	**-27.61**	-38.55	**-30.46**
dna	180	-103.18	**-95.89**	-124.06	**-99.41**	-98.53	**-90.58**
kosarek	190	-21.04	**-12.98**	-19.46	**-12.45**	-18.23	**-10.68**
msweb	294	-24.57	**-12.28**	-24.37	**-12.33**	-22.48	**-11.05**
book	500	-58.07	**-41.80**	-56.28	**-41.46**	-51.81	**-34.49**
movie	500	-97.27	**-77.21**	-94.39	**-70.00**	-64.99	**-34.14**
webkb	839	-207.12	**-172.65**	-199.76	**-162.30**	-181.05	**-126.06**
reuters	889	-152.53	**-113.42**	-146.47	**-109.21**	-149.49	**-111.29**
20newsg	910	-194.55	**-157.70**	-181.47	**-155.48**	-177.83	**-152.65**
bbc	1058	-325.51	**-270.74**	-265.46	**-210.53**	-243.87	**-223.60**
ad	1556	-112.01	**-47.27**	-93.96	**-41.34**	-96.79	**-53.70**
Total AVG		-79.81	**-64.03**	-75.14	**-59.01**	-69.67	**-55.43**

Table 3: Discriminative (20% evidence) performance of $\mathcal{Q}$ and $\mathcal{R}$ on model CLTs, CNs, MCNs.

		20% Evidence					
Datasets	**#var**	**CLTs**		**CNs**		**MCNs**	
		$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$
nltcs	16	-4.81	**-4.42**	-4.91	**-4.66**	-4.34	**-4.10**
msnbc	17	-5.78	**-5.61**	-6.23	**-6.10**	-5.25	**-4.89**
kdd	64	-5.10	**-3.55**	-5.34	**-3.06**	-4.81	**-3.38**
plants	69	-13.63	**-12.82**	-13.32	**-12.88**	-12.59	**-12.02**
audio	100	-42.35	**-40.19**	-40.58	**-37.15**	-37.86	**-35.27**
jester	100	-55.06	**-52.38**	-51.99	**-51.58**	-49.06	**-47.08**
netflix	100	-60.74	**-60.16**	-55.43	**-53.05**	-52.32	**-50.85**
accidents	111	-37.18	**-36.25**	-36.72	**-34.97**	-33.30	**-32.68**
retail	135	-15.12	**-12.42**	-15.05	**-13.02**	-14.14	**-11.65**
pumsb*	163	-34.21	**-32.23**	-31.19	**-28.78**	-29.17	**-28.14**
dna	180	-95.59	**-91.40**	-115.77	**-94.19**	-92.93	**-82.54**
kosarek	190	-16.98	**-12.71**	-13.32	**-12.01**	-11.68	**-9.04**
msweb	294	-13.47	**-11.46**	-14.04	**-11.98**	-12.14	**-11.37**
book	500	-26.51	**-23.10**	-42.92	**-21.25**	-32.43	**-20.19**
movie	500	-86.17	**-66.97**	-67.89	**-59.64**	-59.34	**-41.97**
webkb	839	-182.77	**-155.29**	-173.35	**-144.99**	-162.70	**-131.96**
reuters	889	-124.07	**-99.43**	-132.16	**-105.33**	-128.97	**-102.87**
20newsg	910	-182.35	**-151.92**	-167.36	**-152.8**	-163.80	**-142.50**
bbc	1056	-308.09	**-247.37**	-250.69	**-193.32**	-240.87	**-195.05**
ad	1558	-98.00	**-37.23**	-74.93	**-36.43**	-77.56	**-36.68**
Total AVG		-70.34	**-57.85**	-65.66	**-53.86**	-61.26	**-50.21**

Table 4: Discriminative performance (50% evidence) measured using negative cross entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$ on three models: CLTs, CNs, MCNs.

		Negative cross-entropy with **50% Evidence**					
Datasets	**#var**	**CLTs**		**CNs**		**MCNs**	
		$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$
nltcs	16	-4.14	**-3.53**	-4.03	**-3.86**	-3.53	**-3.18**
msnbc	17	-5.14	**-4.68**	-5.34	**-5.30**	-5.03	**-2.14**
kdd	64	-4.28	**-2.97**	-5.12	**-2.89**	-3.82	**-2.19**
plants	69	-16.07	**-8.75**	-16.32	**-9.72**	-15.15	**-8.28**
audio	100	-32.14	**-26.68**	-31.59	**-30.76**	-28.83	**-25.93**
jester	100	-49.55	**-45.73**	-49.10	**-48.08**	-44.55	**-42.27**
netflix	100	-56.43	**-56.14**	-51.97	**-48.84**	-48.28	**-46.68**
accidents	111	-34.27	**-31.73**	-33.03	**-29.3**	-30.1	**-28.34**
retail	135	-11.75	**-9.02**	-12.44	**-11.27**	-11.07	**-9.32**
pumsb*	163	-29.69	**-25.78**	-25.57	**-21.92**	-25.18	**-22.10**
dna	180	-90.48	**-82.46**	-107.81	**-88.33**	-85.73	**-74.24**
kosarek	190	-12.62	**-9.00**	-11.95	**-5.73**	-9.23	**-4.17**
msweb	294	-12.28	**-8.18**	-11.97	**-9.89**	-10.27	**-9.35**
book	500	-23.42	**-15.05**	-20.42	**-18.05**	-20.49	**-13.96**
movie	500	-77.15	**-59.46**	-62.22	**-54.47**	-55.20	**-36.89**
webkb	839	-167.71	**-152.22**	-155.01	**-139.86**	-134.67	**-123.38**
reuters	889	-113.17	**-91.79**	-105.21	**-85.56**	-110.26	**-89.12**
20newsg	910	-157.03	**-128.89**	-143.25	**-139.89**	-144.61	**-121.72**
bbc	1056	-247.79	**-209.23**	-243.75	**-174.1**	-241.93	**-184.89**
ad	1558	-88.33	**-34.38**	-65.31	**-34.53**	-72.13	**-34.78**
Total AVG		-61.67	**-50.28**	-58.07	**-48.12**	-55.00	**-44.15**

D.2 PERFORMANCE WITH CHANGING PERTURBATION RATE

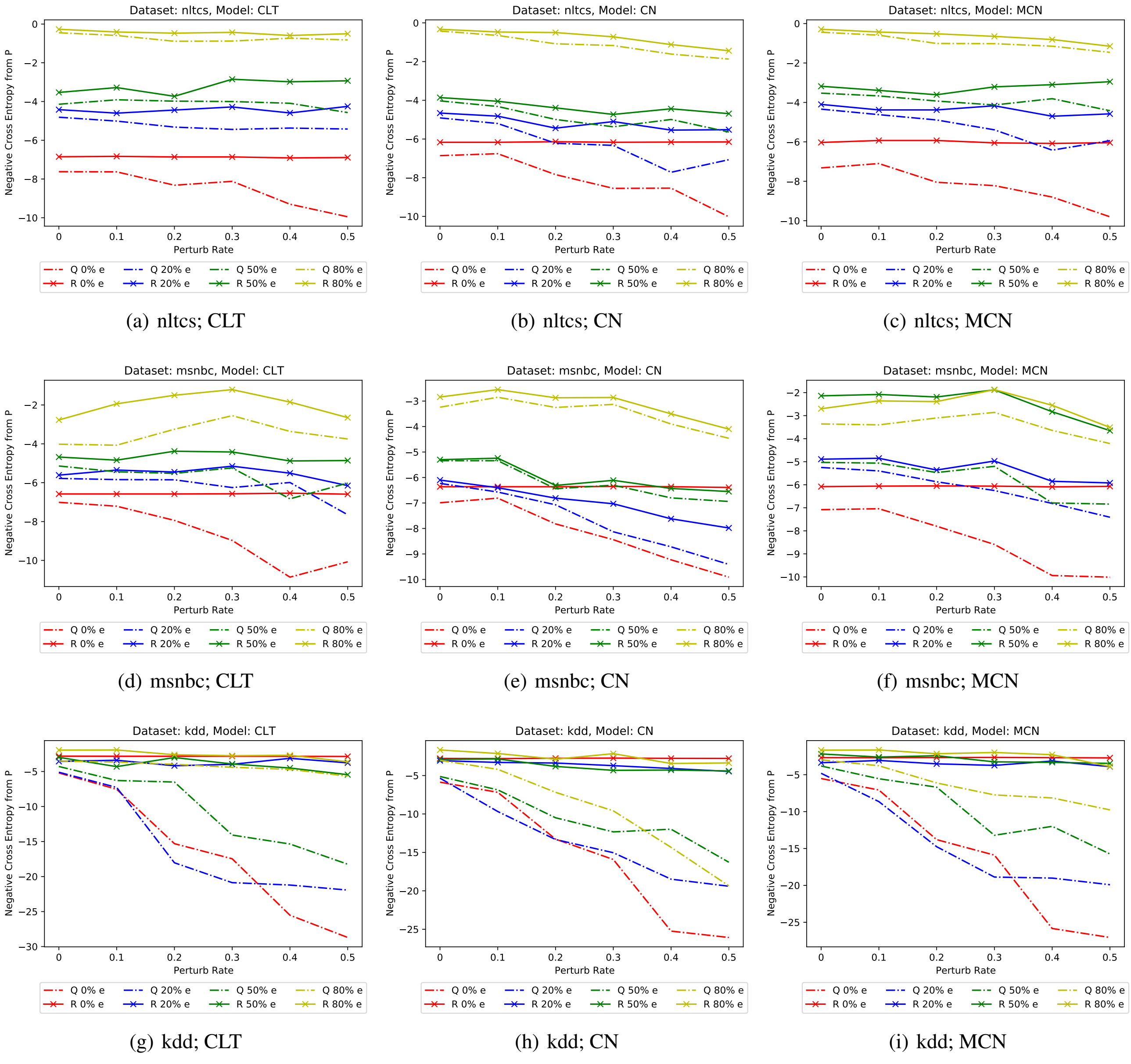
Perturb rate h (between 0 and 100, is used to control the quality of $\mathcal{Q}$ by replacing $h\%$ of the parameters in $\mathcal{Q}$ with a random number). We show these results in Figures 3 – 9.

D.3 EFFECT OF VARYING THE STANDARD DEVIATION

Standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$). We present these results in Figures 10 – 16.

Table 5: Discriminative performance (80% evidence) measured using negative cross entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$ on three models: CLTs, CNs, MCNs.

		80% Evidence					
Datasets	**#var**	**CLTs**		**CNs**		**MCNs**	
		$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{Q}$	$\mathcal{R}$
nltcs	16	-0.45	**-0.27**	-0.42	**-0.33**	-0.44	**-0.29**
msnbc	17	-4.02	**-2.77**	-3.24	**-2.84**	-3.36	**-2.70**
kdd	64	-3.51	**-1.94**	-3.02	**-1.67**	-2.99	**-1.68**
plants	69	-10.27	**-7.20**	-10.23	**-7.70**	-9.35	**-6.64**
audio	100	-27.23	**-25.88**	-21.55	**-17.76**	-21.02	**-18.20**
jester	100	-44.68	**-29.78**	-35.07	**-34.24**	-35.51	**-27.13**
netflix	100	-55.29	**-53.86**	-47.55	**-46.98**	-46.38	**-45.28**
accidents	111	-34.19	**-28.96**	-32.36	**-27.84**	-29.53	**-25.56**
retail	135	-8.86	**-6.08**	-8.09	**-5.95**	-7.38	**-4.98**
pumsb*	163	-16.75	**-15.62**	-13.88	**-11.22**	-12.31	**-11.32**
dna	180	-79.33	**-65.40**	-99.28	**-82.15**	-75.45	**-60.68**
kosarek	190	-10.38	**-8.63**	-8.82	**-4.40**	-7.87	**-5.22**
msweb	294	-11.66	**-5.66**	-7.72	**-6.87**	-7.72	**-6.31**
book	500	-17.36	**-15.17**	-15.83	**-13.05**	-14.54	**-11.72**
movie	500	-59.20	**-46.65**	-52.70	**-39.36**	-39.65	**-21.40**
webkb	839	-105.83	**-72.08**	-91.48	**-61.68**	-90.19	**-55.65**
reuters	889	-83.56	**-68.86**	-74.57	**-62.59**	-77.75	**-65.24**
20newsg	910	-93.59	**-85.91**	-87.60	**-74.44**	-79.68	**-48.44**
bbc	1056	-171.31	**-163.04**	-164.16	**-148.02**	-167.02	**-158.23**
ad	1558	-80.70	**-26.19**	-57.95	**-30.09**	-55.60	**-27.13**
Total AVG		-45.91	**-36.5**	-41.78	**-33.56**	-39.19	**-30.19**



(a) nltcs; CLT

(b) nltcs; CN

(c) nltcs; MCN

(d) msnbc; CLT

(e) msnbc; CN

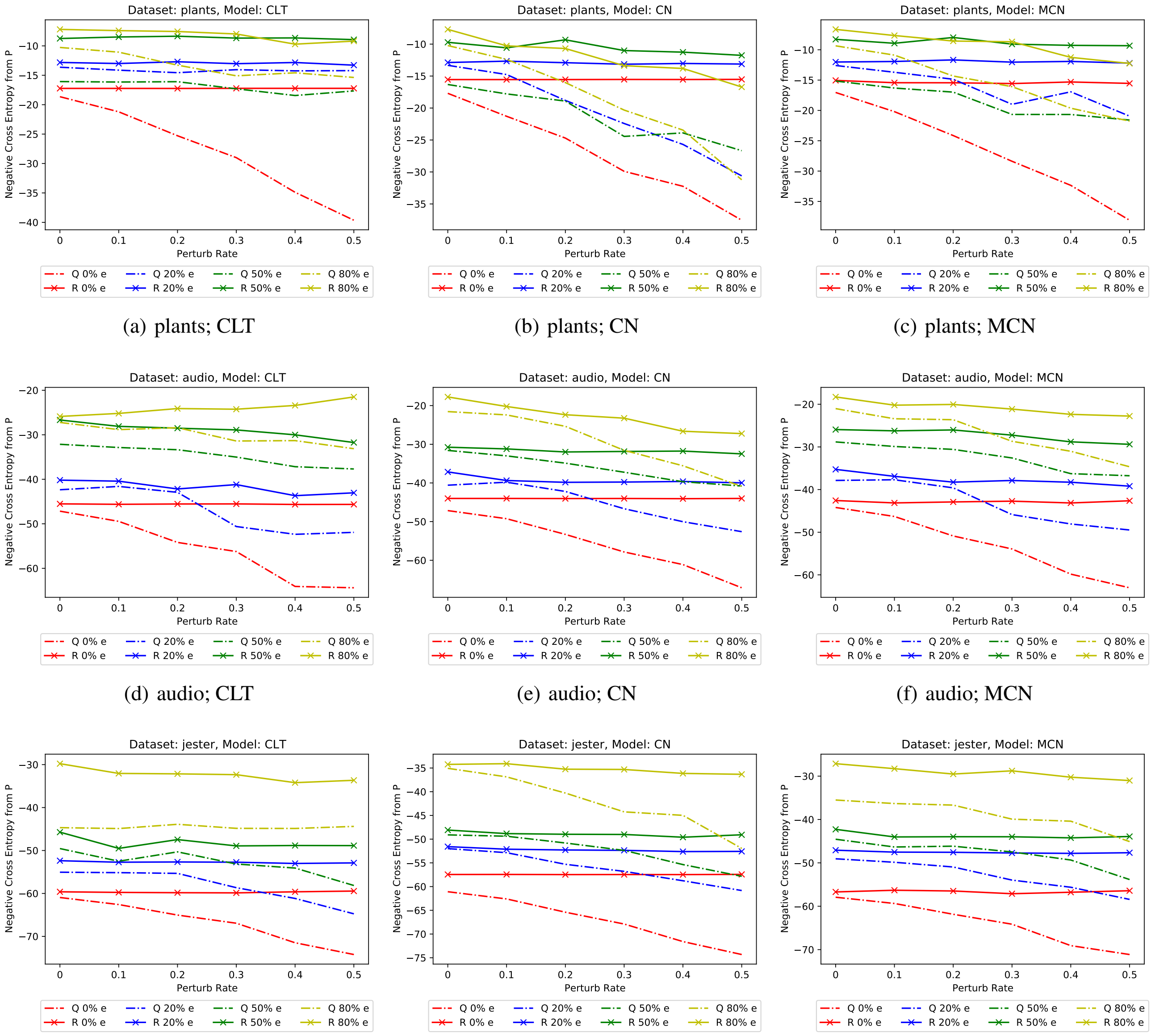
(f) msnbc; MCN

(g) kdd; CLT

(h) kdd; CN

(i) kdd; MCN

Figure 3: Negative Cross Entropy of $\mathcal{P}$ and $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of perturb rate for datasets: nltcs, msnbc, kdd.



(a) plants; CLT

(b) plants; CN

(c) plants; MCN

(d) audio; CLT

(e) audio; CN

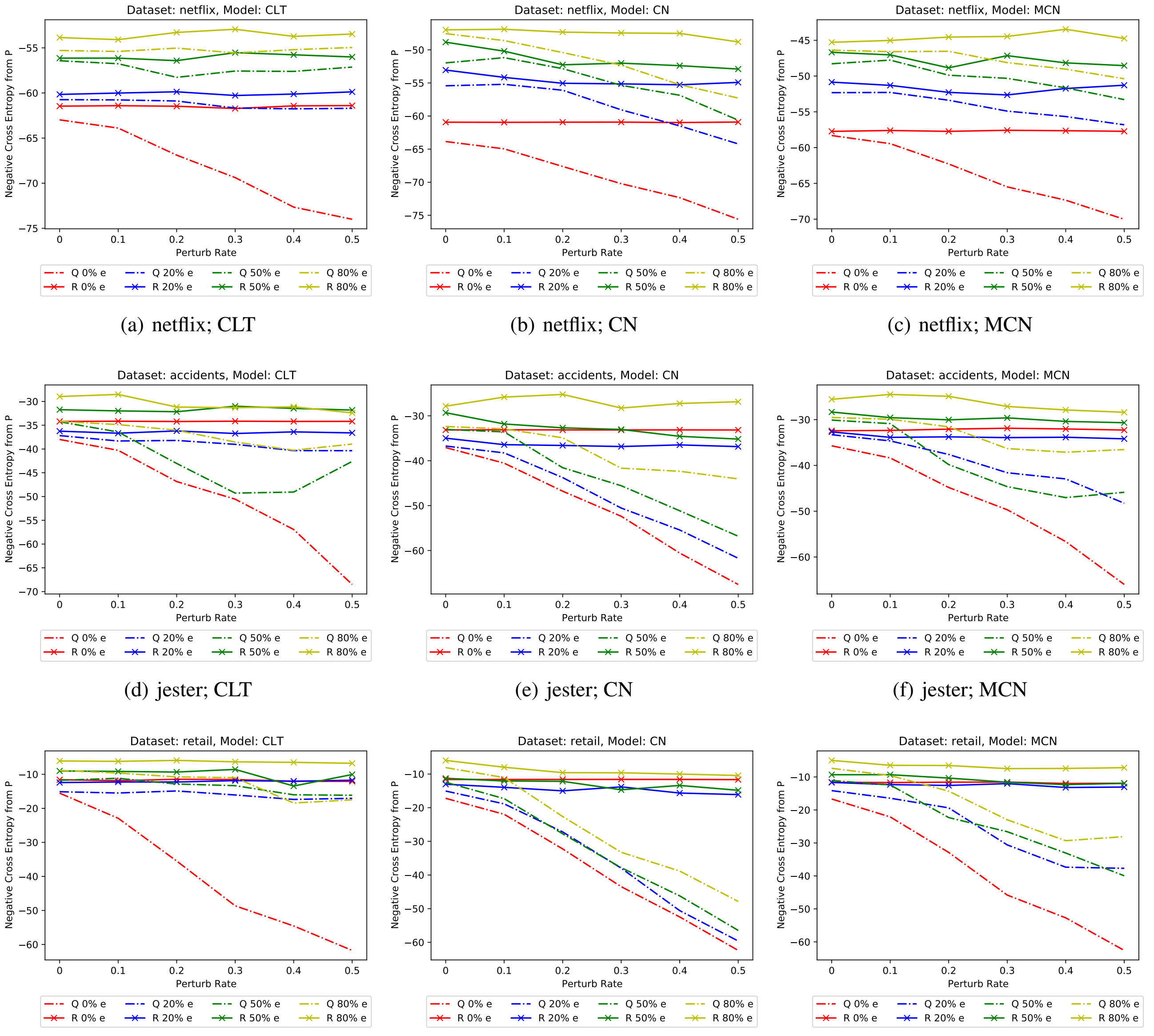
(f) audio; MCN

(g) jester; CLT

(h) jester; CN

(i) jester; MCN

Figure 4: Negative Cross Entropy of $\mathcal{P}$ and $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of perturb rate for datasets: plants, audio, jester.



(a) netflix; CLT

(b) netflix; CN

(c) netflix; MCN

(d) jester; CLT

(e) jester; CN

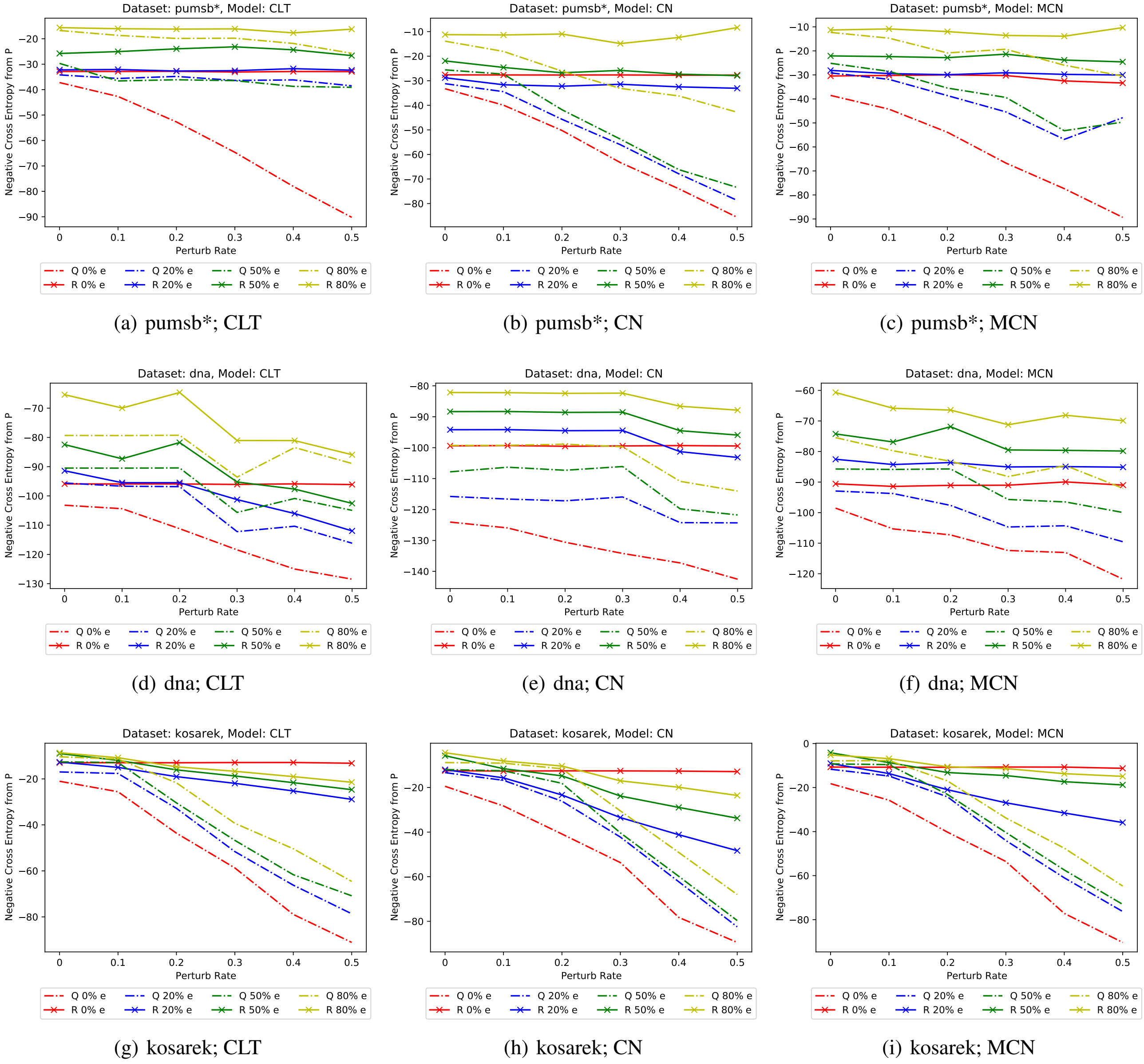
(f) jester; MCN

(g) retail; CLT

(h) retail; CN

(i) retail; MCN

Figure 5: Negative Cross Entropy of $\mathcal{P}$ and $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of perturb rate for datasets: netflix, accidents, tretail.



(a) pumsb*; CLT

(b) pumsb*; CN

(c) pumsb*; MCN

(d) dna; CLT

(e) dna; CN

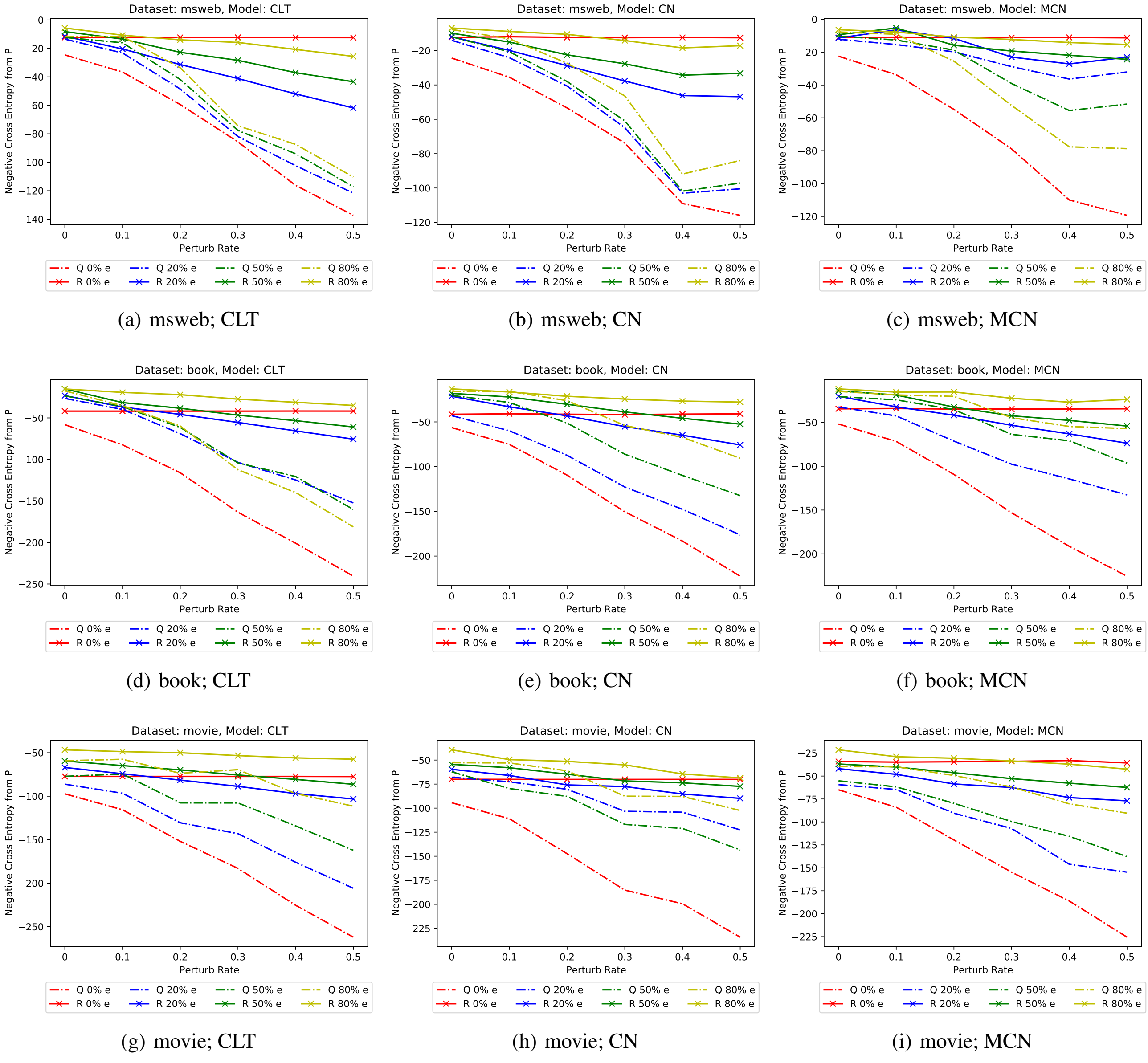
(f) dna; MCN

(g) kosarek; CLT

(h) kosarek; CN

(i) kosarek; MCN

Figure 6: Negative Cross Entropy of $\mathcal{P}$ and $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of perturb rate for datasets: pumsb*, dna, kosarek



(a) msweb; CLT

(b) msweb; CN

(c) msweb; MCN

(d) book; CLT

(e) book; CN

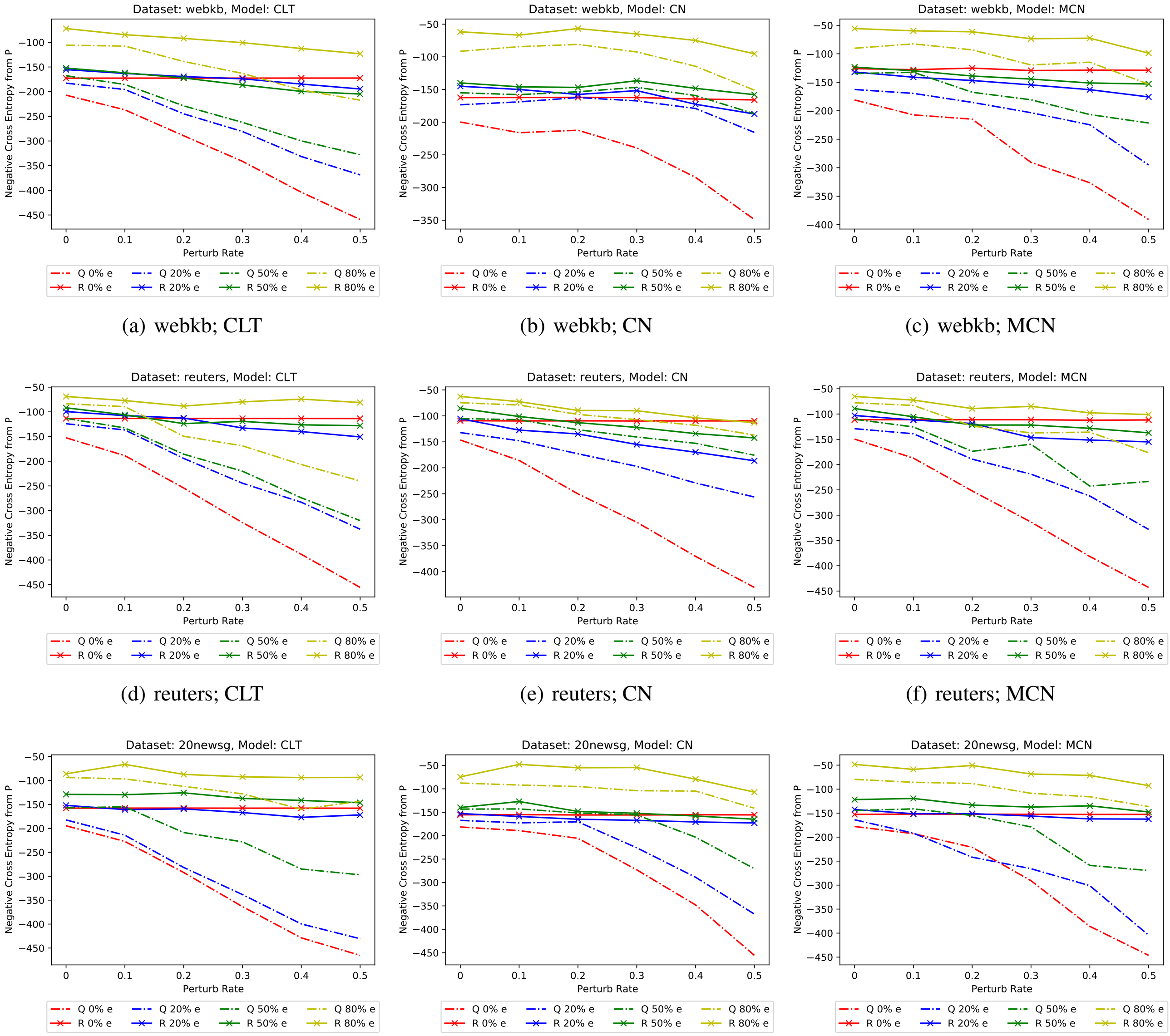
(f) book; MCN

(g) movie; CLT

(h) movie; CN

(i) movie; MCN

Figure 7: Negative Cross Entropy of $\mathcal{P}$ and $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of perturb rate for: msweb, book, movie.



(a) webkb; CLT

(b) webkb; CN

(c) webkb; MCN

(d) reuters; CLT

(e) reuters; CN

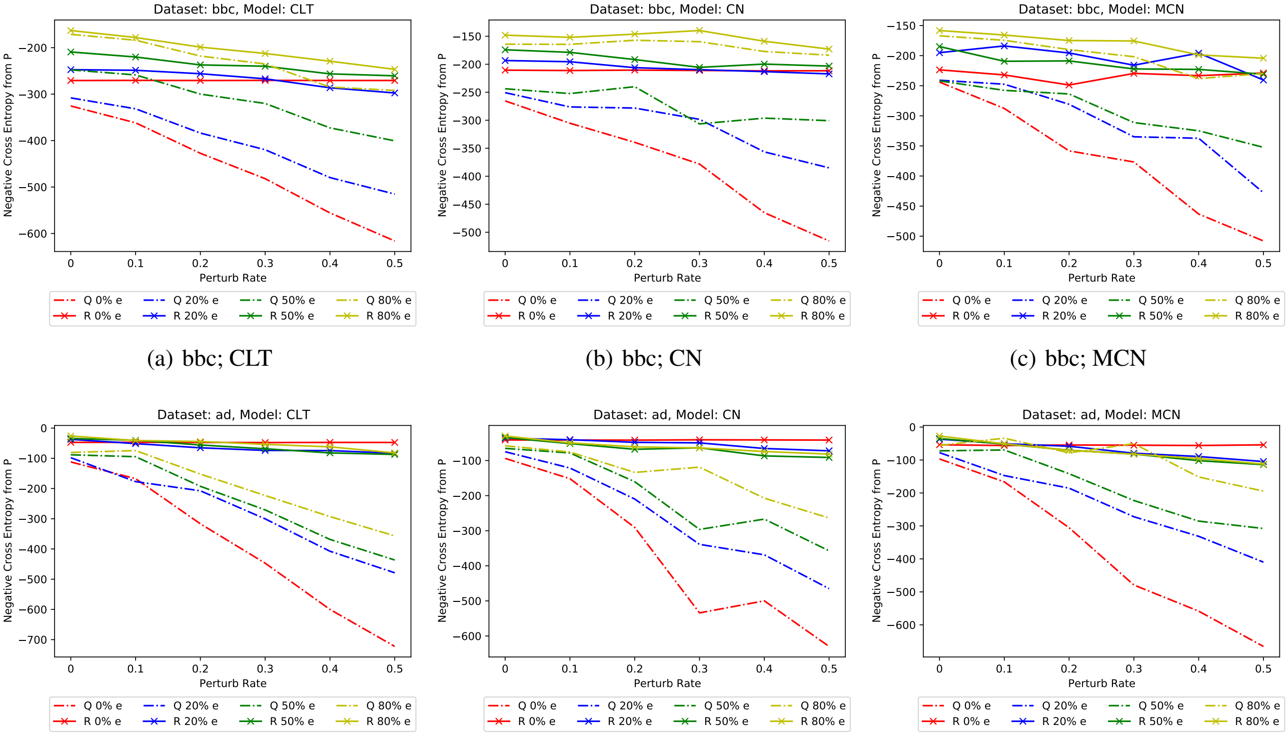
(f) reuters; MCN

(g) 20newsg; CLT

(h) 20newsg; CN

(i) 20newsg; MCN

Figure 8: Negative Cross Entropy of $\mathcal{P}$ and $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of perturb rate for datasets: webkb, reuters, 20newsg.



(a) bbc; CLT

(b) bbc; CN

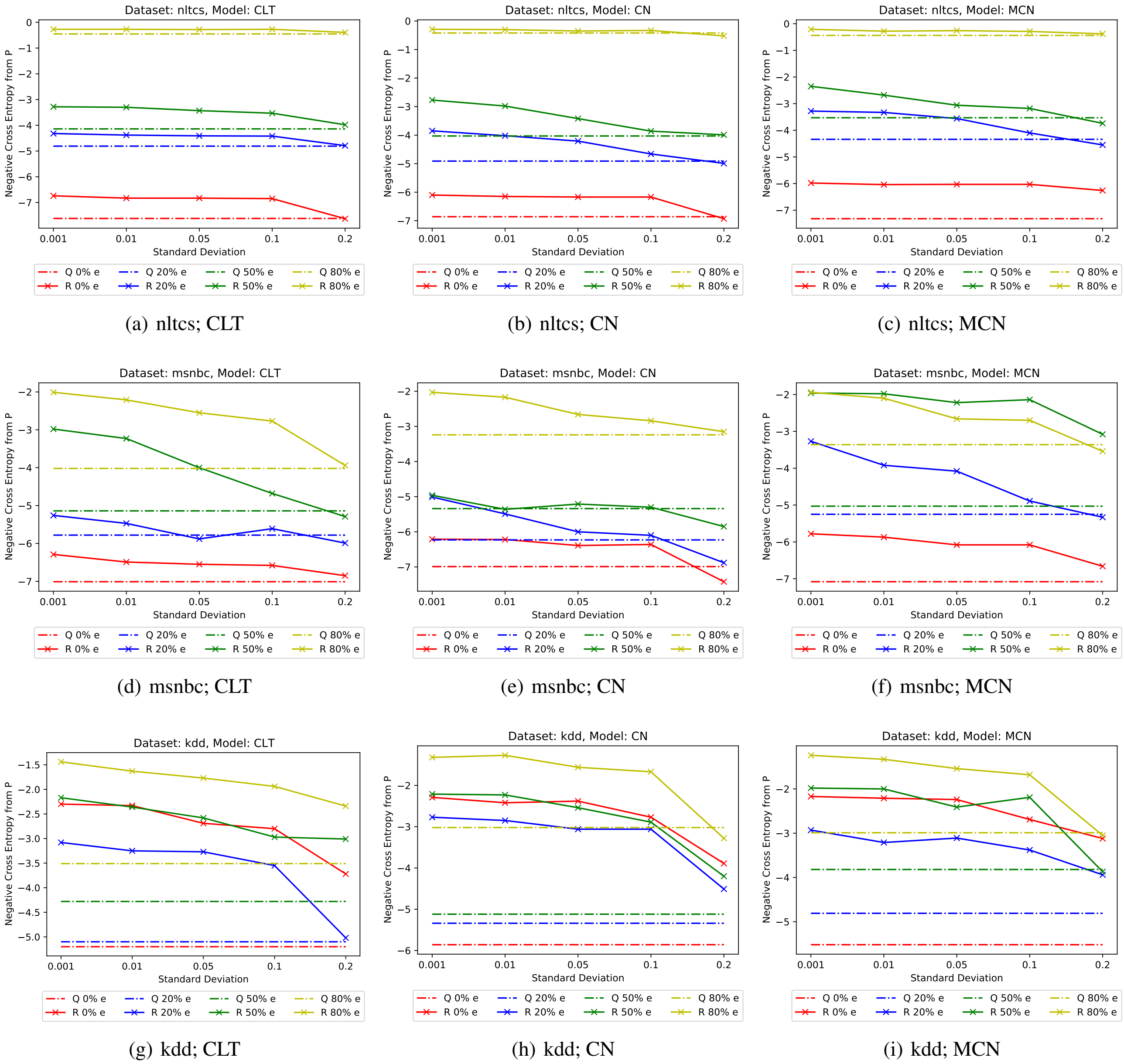
(c) bbc; MCN

(d) ad; CLT

(e) ad; CN

(f) ad; MCN

Figure 9: Negative Cross Entropy of $\mathcal{P}$ and $\mathcal{Q}$, $\mathcal{P}$ and $\mathcal{R}$ with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of perturb rate for datasets: 20newsg, bbc, ad.



(a) nltcs; CLT

(b) nltcs; CN

(c) nltcs; MCN

(d) msnbc; CLT

(e) msnbc; CN

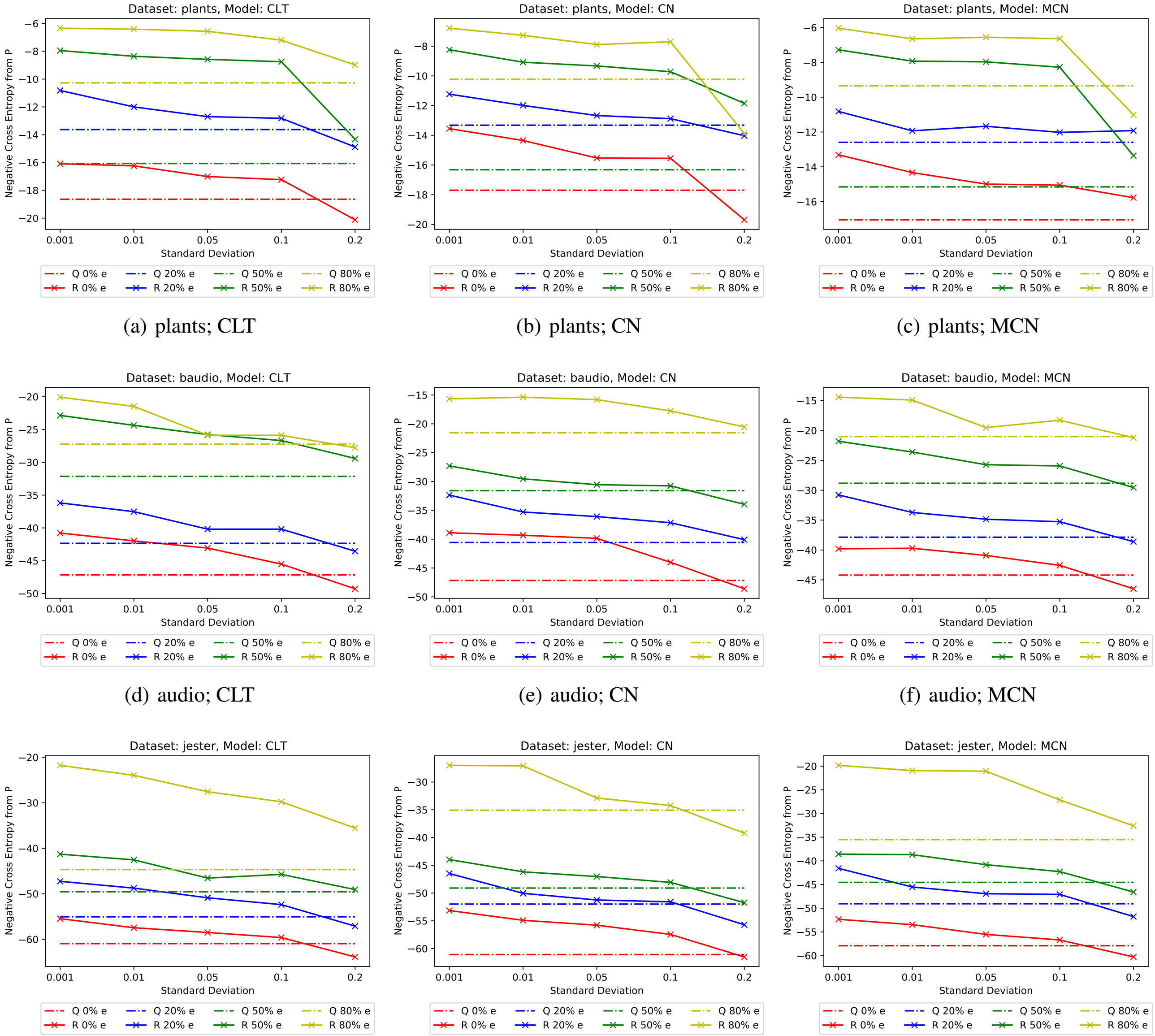
(f) msnbc; MCN

(g) kdd; CLT

(h) kdd; CN

(i) kdd; MCN

Figure 10: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$, with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$) for datasets: nltcs, msnbc, kdd.



(a) plants; CLT

(b) plants; CN

(c) plants; MCN

(d) audio; CLT

(e) audio; CN

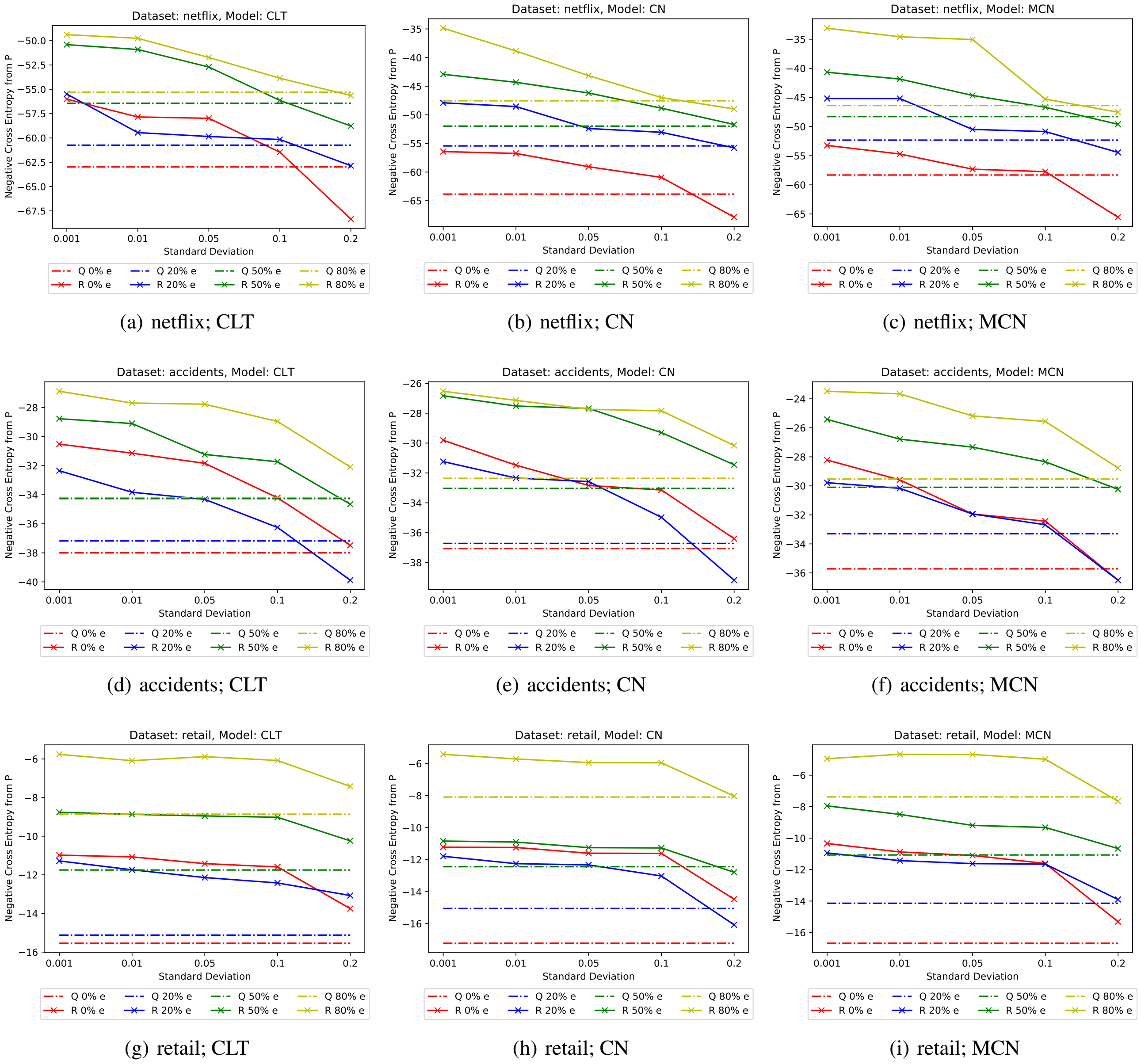
(f) audio; MCN

(g) jester; CLT

(h) jester; CN

(i) jester; MCN

Figure 11: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$, with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$) for datasets: plants, audio, jester.



(a) netflix; CLT

(b) netflix; CN

(c) netflix; MCN

(d) accidents; CLT

(e) accidents; CN

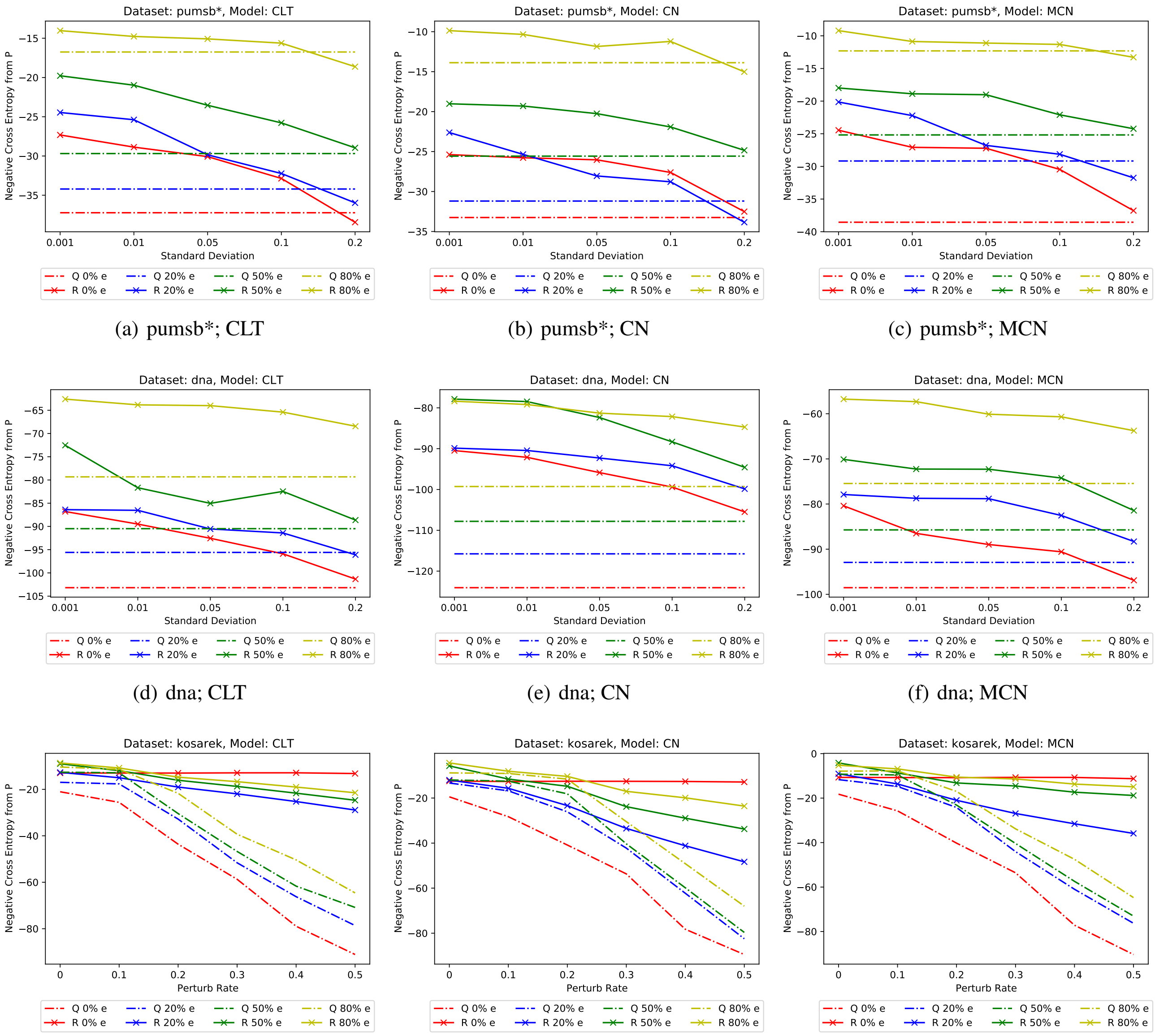
(f) accidents; MCN

(g) retail; CLT

(h) retail; CN

(i) retail; MCN

Figure 12: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$, with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$) for datasets: netflix, accidents, retail.



(a) pumsb*; CLT

(b) pumsb*; CN

(c) pumsb*; MCN

(d) dna; CLT

(e) dna; CN

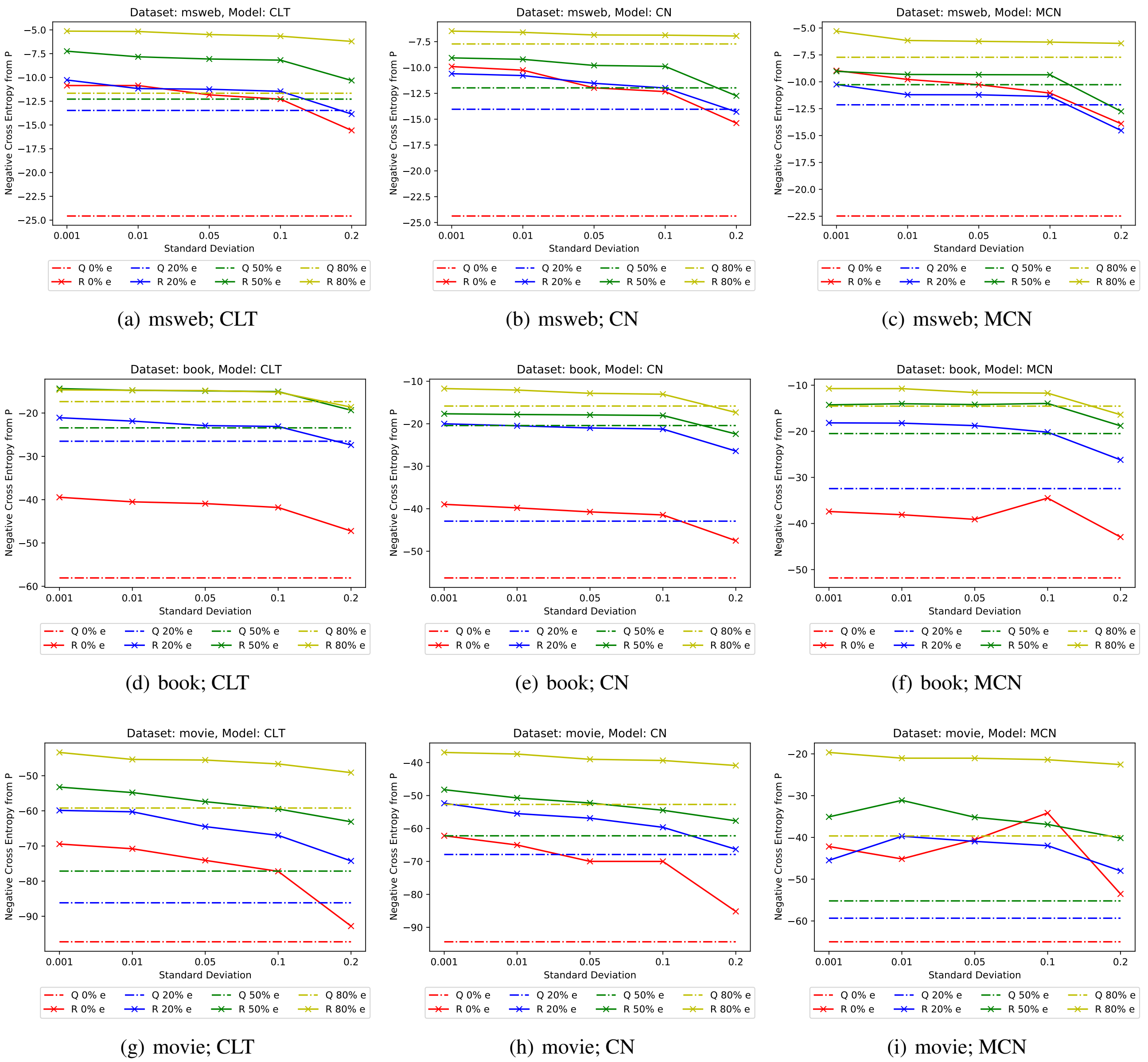
(f) dna; MCN

(g) kosarek; CLT

(h) kosarek; CN

(i) kosarek; MCN

Figure 13: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$, with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$) for datasets: pumsb*, dna, kosarek.



(a) msweb; CLT

(b) msweb; CN

(c) msweb; MCN

(d) book; CLT

(e) book; CN

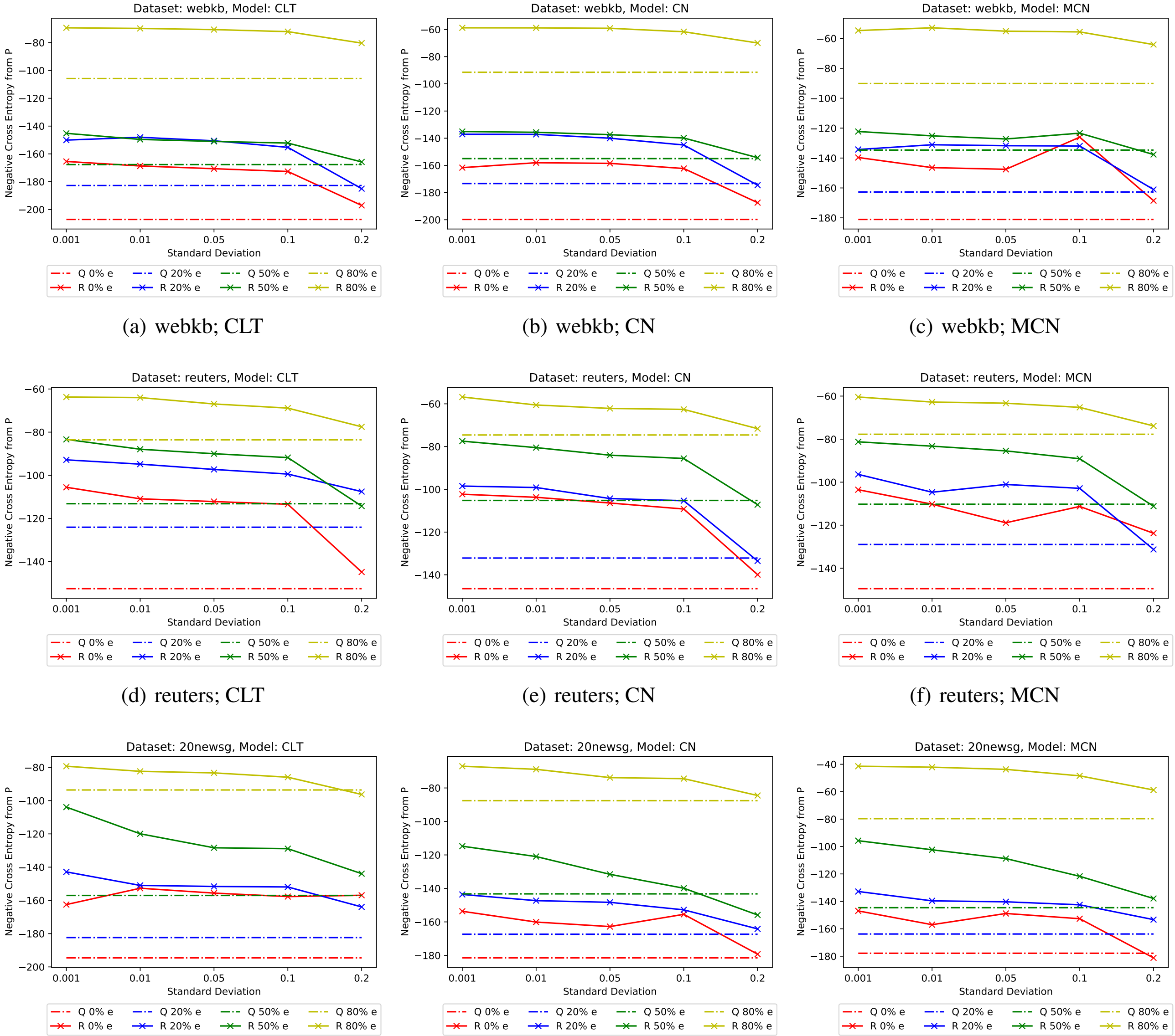
(f) book; MCN

(g) movie; CLT

(h) movie; CN

(i) movie; MCN

Figure 14: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$, with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$) for datasets: msweb, book, movie.



(a) webkb; CLT

(b) webkb; CN

(c) webkb; MCN

(d) reuters; CLT

(e) reuters; CN

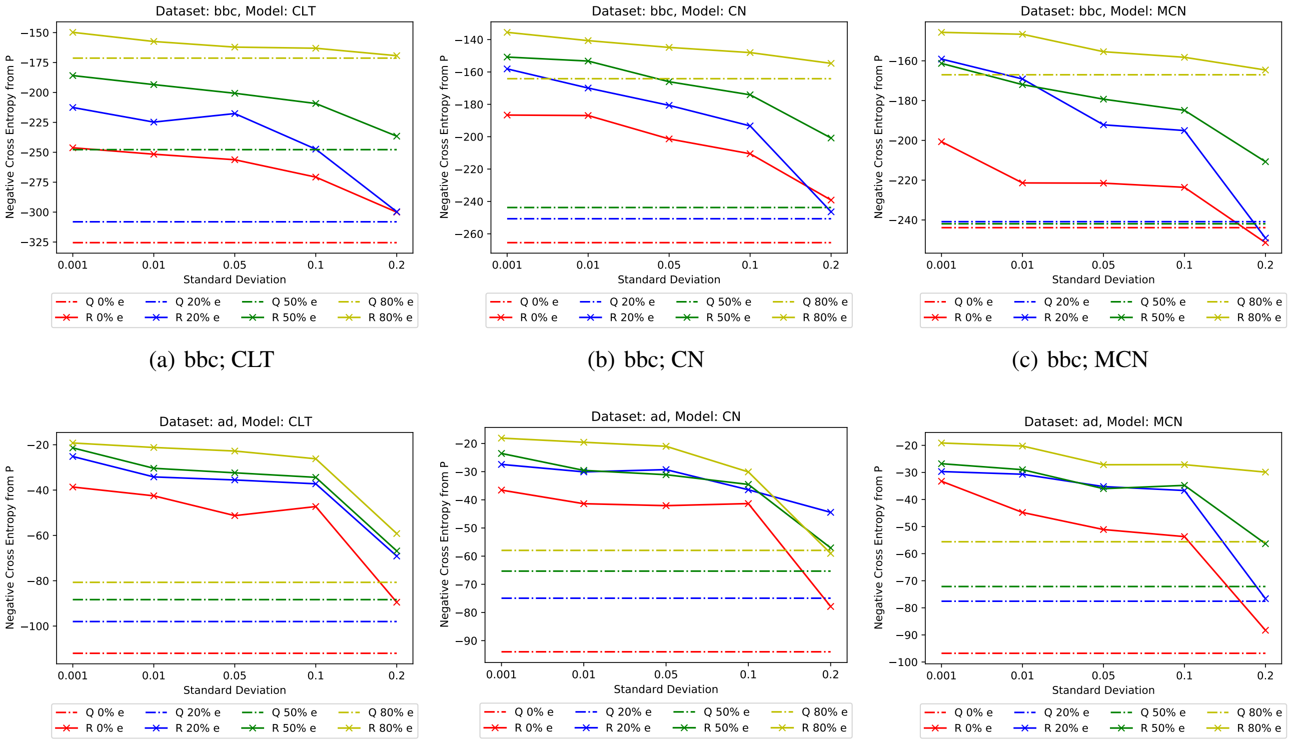
(f) reuters; MCN

(g) 20newsg; CLT

(h) 20newsg; CN

(i) 20newsg; MCN

Figure 15: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$, with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$) for datasets: webkb, reuters, 20newsg.



(a) bbc; CLT

(b) bbc; CN

(c) bbc; MCN

(d) ad; CLT

(e) ad; CN

(f) ad; MCN

Figure 16: Negative Cross Entropy between $\mathcal{P}$ and $\mathcal{Q}$, and between $\mathcal{P}$ and $\mathcal{R}$, with evidence of 0%, 20%, 50% and 80% on three different models: CLT, CN and MCN, as a function of standard deviation σ (of Gaussian noise that is applied to the local statistics $\mathcal{P}_{jk}(X_j, X_k)$) for datasets: bbc, ad.