Jin, R. and Ghahramani, Z. Learning with multiple labels. In *Neural Information Processing Systems*, 2002.

Joulin, A., Bach, F., and Ponce, J. Discriminative clustering for image co-segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2010.

Kendall, M. A new measure of rank correlation. *Biometrika*, 1938.

Korba, A., Garcia, A., and d'Alché-Buc, F. A structured prediction approach for label ranking. In *Neural Information Processing Systems*, 2018.

Lienen, J. and Hüllermeier, E. From label smoothing to label relaxation. In *AAAI Conference on Artificial Intelligence*, 2021.

Liu, L.-P. and Dietterich, T. Learnability of the superset label learning problem. In *International Conference on Machine Learning*, 2014.

Luo, J. and Orabona, F. Learning from candidate labeling sets. In *Neural Information Processing Systems*, 2010.

Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I., and Sivic, J. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *International Conference on Computer Vision*, 2019.

Muzellec, B., Josse, J., Boyer, C., and Cuturi, M. Missing data imputation using optimal transport. In *International Conference of Machine Learning*, 2020.

Nowak-Vila, A., Bach, F., and Rudi, A. Sharp analysis of learning with discrete losses. In *Artificial Intelligence and Statistics*, 2019.

Papandreou, G., Chen, L.-C., Murphy, K., and Yuille, A. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *International Conference on Computer Vision*, 2015.

Perchet, V. and Quincampoix, M. On a unified framework for approachability with full or partial monitoring. *Mathematics of Operations Research*, 2015.

Quadrianto, N., Smola, A., Caetano, T., and Le, Q. V. Estimating labels from label proportions. *Journal of Machine Learning Research*, 2009.

Rigollet, P. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 2007.

Rubin, D. Inference and missing data. *Biometrika*, 1976.

Sheppard, W. On the calculation of the most probable values of frequency constants, for data arranged according to equidistant division of a scale. *Proceedings of the London Mathematical Society*, 1897.

Steinwart, I. and Christmann, A. *Support vector machines*. Springer Science & Business Media, 2008.

Stone, C. Consistent nonparametric regression. *The Annals of Statistics*, 1977.

Tobin, J. Estimation of relationships for limited dependent variables. *Econometrica*, 1958.

van Rooyen, B. and Williamson, R. C. A theory of learning with corrupted labels. *Journal of Machine Learning Research*, 2017.

Vapnik, V. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., and Ruan, X. Learning to detect salient objects with image-level supervision. In *Conference on Computer Vision and Pattern Recognition*, 2017.

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2021.

Xu, L., Neufeld, J., Larson, B., and Schuurmans, D. Maximum margin clustering. *Neural Information Processing Systems*, 2004.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. Learning with local and global consistency. In *Neural Information Processing Systems*, 2003.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. Semi-supervised learning using Gaussian fields and harmonic functions. In *International Conference of Machine Learning*, 2003.

A. Proofs

Mathematical assumptions. To make formal what should be seen as implicit assumptions heretofore, we consider $\mathcal{X}$ and $\mathcal{Y}$ Polish spaces, $\mathcal{Y}$ compact, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ continuous, $\mathcal{H}$ a separable Hilbert space, φ measurable, and ψ continuous. We also assume that for ν_x-almost every $x \in \mathcal{X}$, and any $\mu \vdash \nu$, that the pushforward measure $\varphi_* \mu|_x$ has a second moment. This is the sufficient setup in order to be able to define formally objects and solutions considered all along the paper.

Notations. Beside standard notations, we use $\#\mathcal{Y}$ to design the cardinality of $\mathcal{Y}$, and $2^{\mathcal{Y}}$ to design the set of subsets of $\mathcal{Y}$. Regarding measures, we use $\mu_{\mathcal{X}}$ and $\mu|_x$ respectively the marginal over $\mathcal{X}$ and the conditional accordingly to x of $\mu \in \Delta_{\mathcal{X}\times\mathcal{Y}}$. We denote by $\mu^{\otimes n}$ the distribution of the random variable $(Z_1, \cdots, Z_n)$, where the Z_i are sampled independently according to μ. For A a Polish space, we consider Δ_A the set of Borel probability measures on this space. For $\varphi : \mathcal{Y} \to \mathcal{H}$ and $S \subset \mathcal{Y}$, we denote by $\varphi(S)$ the set $\{\varphi(y) \mid y \in S\}$. For a family of sets (S_i), we denote by $\prod S_i$ the Cartesian product $S_1 \times S_2 \times \cdots$, also defined as the set of points (y_i) such that $y_i \in S_i$ for all index i, and by $\mathcal{Y}^n$ the Cartesian product $\prod_{i \leq n} \mathcal{Y}$. Finally, for E a subset of a vector space E', $\operatorname{Conv} E$ denotes the convex hull of E and $\operatorname{Span}(E)$ its span.

Abuse of notations. For readability sake, we have abused notations. For a signed measure μ, we denote by $\mathbb{E}_\mu[X]$ the integral $\int x \,\mathrm{d}\mu(x)$, extending this notation usually reserved to probability measure. More importantly, when considering $2^{\mathcal{Y}}$, we should actually restrict ourselves to the subspace $\mathcal{S} \subset 2^{\mathcal{Y}}$ of closed subsets of $\mathcal{Y}$, as $\mathcal{S}$ is a Polish space (metrizable by the Hausdorff distance) while $2^{\mathcal{Y}}$ is not always. However, when $\mathcal{Y}$ is finite, those two spaces are equals, $2^{\mathcal{Y}} = \mathcal{S}$.

A.1. Proof of Lemma 3

From Lemma 3 in Cabannes et al. (2021), we pulled the calibration inequality

$$\mathcal{R}(f_n) - \mathcal{R}(f^*) \leq 2c_\psi \, \mathbb{E}\left[\mathbf{1}_{\|g_n(X) - g^*(X)\| > d(g^*(X), F)} \|g_n(X) - g^*(X)\|\right].$$

Where F is defined as the set of points $\xi \in \operatorname{Conv} \varphi(\mathcal{Y})$ leading to two decodings

$$F = \left\{\xi \in \operatorname{Conv} \varphi(\mathcal{Y}) \,\middle|\, \# \operatorname*{arg\,min}_{z \in \mathcal{Y}} \langle \psi(z), \xi \rangle > 1\right\},$$

and d is defined as the extension of the norm distance to sets, for $\xi \in \mathcal{H}$

$$d(\xi, F) = \inf_{\xi' \in F} \|\xi - \xi'\|_{\mathcal{H}}.$$

Using that $\|g_n(X) - g^*(X)\| \leq \left\|g_n(X) - g_n^*(X)\right\| + \left\|g_n^*(X) - g^*(X)\right\|$ and that, if $a \leq b + c$,

$$\mathbf{1}_{a > \delta} a \leq \mathbf{1}_{b + c > \delta} b + c \leq \mathbf{1}_{2 \sup(b,c) > \delta} 2 \sup b, c = 2 \sup_{e \in b, c} \mathbf{1}_{e > \delta} e \leq 2\mathbf{1}_{b > \delta} b + 2\mathbf{1}_{c > \delta} c.$$

We get the refined inequality

$$\mathcal{R}(f_n) - \mathcal{R}(g^*) \leq 4c_\psi \, \mathbb{E}\left[\mathbf{1}_{2\|g_n(X) - g_n^*(X)\| > d(g^*(X), F)} \left\|g_n(X) - g_n^*(X)\right\| + \mathbf{1}_{2\|g_n^*(X) - g^*(X)\| > d(g^*(X), F)} \left\|g_n^*(X) - g^*(X)\right\|\right].$$

The first term is bounded with

$$\mathbb{E}\left[\mathbf{1}_{2\|g_n(X) - g_n^*(X)\| > d(g^*(X), F)} \left\|g_n(X) - g_n^*(X)\right\|\right] \leq \left\|g_n - g_n^*\right\|_{L^1}.$$

While for the second term, we proceed with

$$\mathbb{E}\left[\mathbf{1}_{2\|g_n^*(X) - g^*(X)\| > d(g^*(X), F)} \left\|g_n^*(X) - g^*(X)\right\|\right] \leq \left\|g_n^* - g^*\right\|_{L^\infty} \mathbb{P}_X\left(2\left\|g_n^*(X) - g^*(X)\right\| > \inf_{x \in \operatorname{supp} \nu_{\mathcal{X}}} d(g^*(X), F)\right).$$

When weights sum to one, that is $\sum_{i=1}^n \alpha_i(X) = 1$, both $g_n^*(X)$ and $g^*(X)$ are averaging of $\varphi(y)$ for $y \in \mathcal{Y}$, therefore

$$\left\|g_n^* - g^*\right\|_{L^\infty} \leq 2c_\varphi.$$

Finally, when the labels are a deterministic function of the input, $g^*(X) = \varphi(f^*(X))$, and $d(g^*(X), F) \leq \sup_{y \in \mathcal{Y}} d(\varphi(y), F)$. Defining $\delta := \sup_{y \in \mathcal{Y}} d(\varphi(y), F)/2$, and adding everything together leads to Lemma 3.

A.2. Implication of Assumptions 2 and 3

Assume that Assumption 2 holds, consider $x \in \operatorname{supp} \nu_{\mathcal{X}}$, let us show that $f^*(x) = y_x$ and $\mu^*|_x = \delta_{y_x}$. First of all, notice that $\bigcap_{S; S \in \operatorname{supp} \nu|_x} = \{y_x\}$; that $\delta_{y_x} \vdash \nu|_x$, as it corresponds to $\pi|_{x,S} = \delta_{y_x} \in \Delta_S$, for all S in the support of $\nu|_x$; and that, because ℓ is well-behaved,

$$\inf_{z \in \mathcal{Y}} \ell(z, y_x) = \ell(y_x, y_x) = 0.$$

This infimum is only achieved for $z = y_x$, hence if we prove that $\mu^*|_x = \delta_{y_x}$, we directly have that $f^*(x) = y_x$. Finally, suppose that $\mu|_x \vdash \nu|_x$ charges $y \neq y_x$. Because y does not belong to all sets charged by $\nu|_x$, $\mu|_x$ should charge an other $y' \in \mathcal{Y}$, and therefore

$$\inf_{z \in \mathcal{Y}} \mathbb{E}_{Y \sim \mu|_x} [\ell(z, y)] \geq \inf_{z \in \mathcal{Y}} \mu|_x(y)\ell(z, y) + \mu|_x(y')\ell(z, y') > 0.$$

Which shows that $\mu^*|_x = \delta_{y_x}$. We deduce that $g^*(x) = y_x$.

Now suppose that Assumption 3 holds too, and consider two $x, x' \in \operatorname{supp} \nu_{\mathcal{X}}$ belonging to two different classes $f(x) = y$ and $f(x') = y'$. We have that $g^*(x) = \varphi(y)$ and $g^*(x') = \varphi(y')$, therefore,

$$d(x, x') \geq c^{-1} \left\| \varphi(y) - \varphi(y') \right\|_{\mathcal{H}}.$$

Define $h_2 = \inf_{y \neq y'} c^{-1} \left\| \varphi(y) - \varphi(y') \right\|_{\mathcal{H}}$. Let us now show that $h_2 > 0$. When $\mathcal{Y}$ is finite, this infimum is a minimum, therefore, $h_2 = 0$, only if there exists a $y \neq y'$, such that $\varphi(y) = \varphi(y')$, which would implies that $\ell(\cdot, y) = \ell(\cdot, y')$ and therefore $\ell(y, y') = \ell(y, y)$ which is impossible when ℓ is proper.

A.3. Proof of Theorem 4

Reusing Lemma 3, we have

$$\mathcal{E}(f_n) \leq 4c_\psi \, \mathbb{E}_{\mathcal{D}_n, X} \left[\left\| g_n^*(X) - g_n(X) \right\|_{\mathcal{H}} \right] + 8c_\psi c_\varphi \, \mathbb{E}_{\mathcal{D}_n, X} \left[\mathbf{1}_{\| g_n^*(X) - g^*(X) \| > \delta} \right].$$

We will first prove that

$$\mathbb{E}_{\mathcal{D}_n} \left[\mathbf{1}_{\| g_n^*(X) - g^*(X) \| > \delta} \right] \leq \exp\left(-\frac{np}{8} \right)$$

as long as $k < np/2$. The error between g^* and g_n relates to classical supervised learning of g^* from samples $(X_i, Y_i) \sim \mu^*$. We invite the reader who would like more insights on this fully supervised part of the proof to refer to the several monographs written on local averaging methods and, in particular, nearest neighbors, such as Biau & Devroye (2015). Because of class separation, we know that, if k points fall at distance at most h of $x \in \operatorname{supp} \nu_{\mathcal{X}}$, $g_n^*(x) = k^{-1} \sum_{i; X_i \in \mathcal{N}(x)} \varphi(y_i) = \varphi(y_x) = g^*(x)$, where $\mathcal{N}(x)$ designs the k-nearest neighbors of x in (X_i). Because the probability of falling at distance h of x for each X_i is lower bounded by p, we have that

$$\mathbb{P}_{\mathcal{D}_n}(g_n^*(x) \neq g^*(x)) \leq \mathbb{P}(\text{Bernouilli}(n, p) < k).$$

This can be upper bound by $\exp(-np/8)$ as soon as $k < np/2$, based on Chernoff multiplicative bound (see Biau & Devroye, 2015, for a reference), meaning

$$\mathbb{E}_{\mathcal{D}_n, X} \left[\mathbf{1}_{\| g_n^*(X) - g^*(X) \| \geq \delta} \right] \leq \exp(-np/8).$$

For the disambiguation part in $\left\| g_n - g_n^* \right\|_{L^1}$, we distinguish two types of datasets, the ones where for any input X_i its k-neighbors at are distance at least h, ensuring that disambiguation can be done by clusters, and datasets that does not verify this property. Consider the event

$$\mathbb{D} = \left\{ (X_i)_{i \leq n} \middle| \sup_i d(X_i, X_{(k)}(X_i)) < h \right\}$$

where $X_{(k)}(x)$ design the k-th nearest neighbor of x in $(X_i)_{i \leq n}$. We proceed with

$$\mathbb{E}_{\mathcal{D}_n, X} \left[\left\| g_n^*(X) - g_n(X) \right\|_{\mathcal{H}} \right] \leq \sup_{X \in \mathcal{X}} \left\| g_n^* - g_n \right\|_\infty \mathbb{P}_{\mathcal{D}_n}((X_i) \notin \mathbb{D}) + \mathbb{E}_{\mathcal{D}_n, X} \left[\left\| g_n^*(X) - g_n(X) \right\|_{\mathcal{H}} \middle| (X_i) \in \mathbb{D} \right],$$

Which is based on $E[Z] = \mathbb{P}(Z \in A) \, \mathbb{E}[Z|A] + \mathbb{P}(Z \notin A) \, \mathbb{E}[Z|^c A]$. For the term corresponding to bad datasets, we can bound the disambiguation error with the maximum error. Similarly to the derivation for Lemma 3, because $g_n^*(x)$ and $g_n^*(X)$, are averaging of $\varphi(y)$, we have that

$$\sup_{x \in \operatorname{supp} \nu_{\mathcal{X}}} \left\| g_n(x) - g_n^*(x) \right\| \leq 2c_\varphi.$$

Indeed, we allow ourselves to pay the worst error on those datasets as their probability is really small, which can be proved based on the following derivation.

$$\begin{aligned}\mathbb{P}_{\mathcal{D}_n}((X_i)_{i\leq n}\notin\mathbb{D}) &= \mathbb{P}_{(X_i)}(\sup_i d(X_i, X_{(k)}(X_i)) \geq h) = \mathbb{P}_{(X_i)}\left(\cup_{i\leq n}\left\{d(X_i, X_{(k)}(X_i))\geq h\right\}\right)\\ &\leq \sum_{i=1}^n \mathbb{P}_{(X_i)}\left(d(X_i,X_{(k)}(X_i))\geq h\right) = n\,\mathbb{P}_{X,\mathcal{D}_{n-1}}\left(d(X,X_{(k)}(X))\geq h\right).\end{aligned}$$

This last probability has already been work out when dealing with the fully supervised part, and was bounded as

$$\mathbb{P}_{X,\mathcal{D}_{n-1}}\left(d(X,X_{(k)}(X))\geq h\right) \leq \exp\left(-(n-1)p/8\right).$$

as long as $k < (n-1)p/2$. Finally we have

$$\sup_{X\in\mathcal{X}}\left\|g_n^* - g_n\right\|_\infty \mathbb{P}_{\mathcal{D}_n}((X_i)_{i\leq n}\notin\mathbb{D}) \leq 2c_\varphi n\exp\left(-(n-1)p/8\right).$$

For the expectation term, corresponding to datasets, $\mathcal{D}_n\in\mathbb{D}$, that cluster data accordingly to classes, we have to make sure that $\hat{y}_i = y_i^*$ is the only acceptable solution of Eq. (4), which is true as soon as the intersection of S_j, for x_j the neighbors of x_i, only contained y_i^*. To work out the disambiguation algorithm, notice that

$$\begin{aligned}\left\|g_n - g_n^*\right\|_{L^1} &= \int_{\mathcal{X}}\left\|\sum_{i=1}^n \alpha_i(x)\varphi(\hat{y}_i) - \varphi(y_i^*)\right\|\mathrm{d}\nu_{\mathcal{X}}(x) \leq \int_{\mathcal{X}} k^{-1}\sum_{i=1}^n \mathbf{1}_{X_i\in\mathcal{N}(x)}\left\|\varphi(\hat{y}_i) - \varphi(y_i^*)\right\|\mathrm{d}\nu_{\mathcal{X}}(x)\\ &= k^{-1}\sum_{i=1}^n \mathbb{P}_X\left(X_i\in\mathcal{N}(X)\right)\left\|\varphi(\hat{y}_i) - \varphi(y_i^*)\right\| \leq 2c_\varphi k^{-1}\sum_{i=1}^n \mathbb{P}_X\left(X_i\in\mathcal{N}(X)\right)\mathbf{1}_{\varphi(\hat{y}_i)\neq\varphi(y_i^*)}.\end{aligned}$$

Finally we have, after proper conditionning, considering the variability in S_i while fixing X_i first,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n,X}\left[\left\|g_n^*(X) - g_n(X)\right\|_{\mathcal{H}}\middle|(X_i)\in\mathbb{D}\right] &= 2c_\varphi k^{-1}\,\mathbb{E}_{(X_i)}\left[\sum_{i=1}^n \mathbb{P}_X\left(X_i\in\mathcal{N}(X)\right)\mathbb{E}_{(S_i)}\left[\mathbf{1}_{\varphi(\hat{y}_i)\neq\varphi(y_i^*)}\middle|(X_i)\right]\middle|(X_i)\in\mathbb{D}\right]\\ &= 2c_\varphi k^{-1}\,\mathbb{E}_{(X_i),X}\left[\sum_{i=1}^n \mathbf{1}_{X_i\in\mathcal{N}(X)}\,\mathbb{P}_{(S_i)}\left(\varphi(\hat{y}_i)\neq\varphi(y_i^*)\middle|(X_i)\right)\middle|(X_i)\in\mathbb{D}\right].\end{aligned}$$

We design $\mathbb{D}$, because when this event holds, we know that the k-th nearest neighbor of any input X_i is at distance at most h of X_i, meaning the because of class separation, $y_{x_i}\in S_j$ for any $X_j\in\mathcal{N}(X_i)$. This mean that outputting $(\hat{y}_i) = (y_i^*)$ and $z_j = y_j$, will lead to an optimal error in Eq. (4). Now suppose that there is an other solution for Eq. (4) such that $\hat{y}_i\neq y_i^*$, it should also achieve an optimal error, therefore it should verify $z_j = \hat{y}_j$ for all j as well as $\hat{y}_j = \hat{y}_i$ for all j such that X_j is one of the k nearest neighbors of X_i. This implies that $\hat{y}_i\in\cap_{j;X_j\in\mathcal{N}(X_i)}S_j$, which happen with probability

$$\mathbb{P}_{(S_j)_{j;X_j\in\mathcal{N}(X_i)}}(\exists z\neq y_i, z\in\cap_j S_j) \leq m\,\mathbb{P}_{S_j}(z\in S_j)^k \leq m\eta^k = m\exp(-k\left|\log(\eta)\right|).$$

With $m = \#\mathcal{Y}$ the number of element in $\mathcal{Y}$. We deduce that

$$\mathbb{P}_{(S_i)}\left(\varphi(\hat{y}_i)\neq\varphi(y_i^*)\middle|(X_i)\right) \leq m\exp(-k\left|\log(\eta)\right|).$$

And because $\sum_{i=1}^n \mathbf{1}_{X_i\in\mathcal{N}(X)} = k$, we conclude that

$$\mathbb{E}_{\mathcal{D}_n,X}\left[\left\|g_n^*(X) - g_n(X)\right\|_{\mathcal{H}}\middle|(X_i)\in\mathbb{D}\right] \leq 2c_\varphi m\exp(-k\left|\log(\eta)\right|).$$

Finally, adding everything together we get

$$\mathcal{E}(f_n) \leq 8c_\varphi c_\psi\exp\left(-\frac{np}{8}\right) + 8c_\varphi c_\psi n\exp\left(-\frac{(n-1)p}{8}\right) + 8c_\varphi c_\psi m\exp\left(-k\left|\log(\eta)\right|\right).$$

as long as $k < (n-1)p/2$, which implies Theorem 4 as long as $n\geq 2$.

Remark 9 (Other approaches)**.** *While we have proceed with analysis based on local averaging methods, other paths could be explored to prove convergence results of the algorithm provided Eq.* (4) *and* (5)*. For example, one could prove Wasserstein convergence of* $\sum_{i=1}^n \delta_{(x_i, \hat{y}_i)}$ *towards* $\sum_{i=1}^n \delta_{(x_i, \hat{y}_i^*)}$*, together with some continuity of the learning algorithm as a function of those distributions.*[4] *This analysis could be understood as tripartite:*

- *A disambiguation error, comparing* $\hat{y}_i$ *to* y_i^**.*
- *A stability / robustness measure of the algorithm to learn* f_n *from data when substituting* y_i^* *by* $\hat{y}_i$*.*
- *A consistency result regarding* f_n^* *learnt on* (x_i, y_i^*)*.*

Our analysis followed a similar path, yet with the first two parts tackled jointly.

A.4. Proof of Proposition 6

Under the non-ambiguity hypothesis (Assumption 2), the solution of Eq. (3) is characterized pointwise by $f^*(x) = y_x$ for all $x \in \operatorname{supp} \nu_{\mathcal{X}}$. Similarly under Assumption 2, we have the characterization $f^*(x) \in \cap_{S \in \operatorname{supp} \nu|_x} S$. With the notation of Definition 5, since $f^*(x)$ minimizes $z \to \mathbb{E}_{Y \sim \mu_S}[\ell(z, Y)]$ for all $S \in \operatorname{supp} \nu|_x$, it also minimizes $z \to \mathbb{E}_{S \sim \nu|_x} \mathbb{E}_{Y \sim \mu_S}[\ell(z, Y)]$.

For the second part of the proposition, we use the structured prediction framework of Ciliberto et al. (2020). Define the signed measure μ° defined as $\mu^\circ_{\mathcal{X}} := \nu_{\mathcal{X}}$ and $\mu^\circ|_x := \mathbb{E}_{S \sim \nu|_x} \mathbb{E}_{Y \sim \mu_S}[\delta_Y]$, and $f^\circ : \mathcal{X} \to \mathcal{Y}$ the solution $f^\circ \in \arg\min_{f : \mathcal{X} \to \mathcal{Y}} \mathbb{E}_{(X,Y) \sim \mu^\circ}[\ell(f(X), Y)] = \arg\min_{f : \mathcal{X} \to \mathcal{Y}} \mathbb{E}_{(X,Y) \sim \nu}\left[\mathbb{E}_{Y \sim \mu_S}[\ell(f(X), Y)]\right]$. The first part of the proposition tells us that $f^\circ = f^*$ under Assumption 2. The framework of Ciliberto et al. (2020), tells us that f° is obtained after decoding, Eq. (9), of $g^\circ : \mathcal{X} \to \mathcal{H}$, and that if g_n° converges to g° with the L^1 norm, f_n° converges to f° in term of the μ°-risk. Under Assumption 2 and mild hypothesis on μ°, it is possible to prove that convergence in term of the μ°-risk implies convergence in term of the μ-risk (for example through calibration inequality similar to Proposition 2 of Cabannes et al. (2020)).

A.5. Ranking with Partial ordering is a well behaved problem

Here, we discuss about building directly ξ_S to initialize our alternative minimization scheme or considering μ_S given by the definition of well-behaved problem (Definition 5). Since the existence of μ_S implying ξ_S defined as $\mathbb{E}_{Y \sim \mu_S}[\varphi(Y)]$, we will only study when ξ_S can be cast as a μ_S.

In ranking, we have that $\psi = -\varphi$, which corresponds to "correlation losses". In this setting, we have that $\operatorname{Span}(\varphi(\mathcal{Y})) = \operatorname{Span}(\psi(\mathcal{Y}))$. More generally, looking at a "minimal" representation of ℓ, one can always assume the equality of those spans, as what happens on the orthogonal of the intersection of those spans, does not modify the scalar product $\varphi(y)^\top \psi(z)$. Similarly, ξ_S can be restricted to $\operatorname{Span}(\psi(\mathcal{Y}))$, and therefore $\operatorname{Span}(\varphi(\mathcal{Y}))$, which exactly the image by $\mu \to \mathbb{E}_{Y \sim \mu}[\varphi(Y)]$ of the set of signed measures, showing the existence of a μ_S matching Definition 5.

B. IQP implementation for Eq. (4)

In this section, we introduce an IQP implementation to solve for Eq. (4). We first mention that our alternative minimization scheme is not restricted to well-behaved problem, before motivating the introduction of the IQP algorithm in two different ways, and finally describing its implementation.

B.1. Initialization of alternative minimization for non well-behaved problem

Before describing the IQP implementation to solve Eq. (12), we would like to stress that, even for non well-behaved partial labelling problems, it is possible to search for smart ways to initialize variables of the alternative minimization scheme. For example, one could look at $z_i^{(0)} \in \cap_{j; x_j \in \mathcal{N}_{k_i}} S_j$, where $\mathcal{N}_k$ designs the k nearest neighbors of x_i in $(x_j)_{j \leq n}$, and k_i is chosen such that this intersection is a singleton.

[4] The Wasserstein metric is useful to think in term of distributions, which is natural when considering partial supervision that can be cast as a set of admissible fully supervised distributions. This approach has been successfully followed by Perchet & Quincampoix (2015) to deal with partial monitoring in games.

B.2. Link with Diffrac and empirical risk minimization

Our IQP algorithm is similar to an existing disambiguation algorithm known as the Diffrac algorithm (Bach & Harchaoui, 2007; Joulin et al., 2010).[5] This algorithm was derived by implicitly following empirical risk minimization of Eq. (2). This approach leads to algorithms written as

$$(y_i) \in \operatorname*{arg\,min}_{(y_i)\in C_n} \inf_{f\in\mathcal{F}} \sum_{i=1}^n \ell(f(x_i), y_i) + \lambda\Omega(f),$$

for $\mathcal{F}$ a space of functions, and $\Omega : \mathcal{F} \to \mathbb{R}_+$ a measure of complexity. Under some conditions, it is possible to simplify the dependency in f (*e.g.,* *Xu et al., 2004; Bach & Harchaoui, 2007*). For example, if $\ell(y, z)$ can be written as $\|\varphi(y) - \varphi(z)\|^2$ for a mapping $\varphi : \mathcal{X} \to \mathcal{Y}$, *e.g.* the Kendall loss detailed in Section 5.4,[6] and the search of $\varphi(f) : \mathcal{X} \to \varphi(\mathcal{Y})$ is relaxed as a $g : \mathcal{X} \to \mathcal{H}$. With Ω and $\mathcal{F}$ linked with kernel regression on the surrogate functional space $\mathcal{X} \to \mathcal{H}$, it is possible to solve the minimization with respect to g as $g(x_i) = \sum_{j=1}^n \alpha_j(x_i)\varphi(y_i)$, with α given by kernel ridge regression (Ciliberto et al., 2016), and to obtain a disambiguation algorithm written as

$$\operatorname*{arg\,min}_{y_i\in S_i} \sum_{i=1}^n \Big\| \sum_{j=1}^n \alpha_j(x_i)\varphi(y_j) - \varphi(y_i) \Big\|^2.$$

This IQP is a special case of the one we will detail. As such, our IQP is a generalization of the Diffrac algorithm, and this paper provides, to our knowledge, *the first consistency result for Diffrac*.

B.3. Link with an other determinism measure

While we have considered the measure of determinism given by Eq. (2), we could have considered its quadratic variant

$$\mu^\star \in \operatorname*{arg\,min}_{\mu\vdash\nu} \inf_{f:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{X\sim\nu_\mathcal{X}} \left[\mathbb{E}_{Y,Y'\sim\mu|_x} \left[\ell(Y, Y')\right]\right].$$

This correspond to the right drawing of Figure 4. We could arguably translate it experimentally as

$$(\hat{y}_i) \in \operatorname*{arg\,min}_{(y_i)\in C_n} \sum_{i,j=1}^n \alpha_i(x_j)\ell(y_i, y_j), \tag{14}$$

and still derive Theorem 4 when substituting Eq. (4) by Eq. (14). When the loss is a correlation loss $\ell(y, z) = -\varphi(y)^\top\varphi(z)$. This leads to the quadratic problem

$$(\hat{y}_i) \in \operatorname*{arg\,min}_{(y_i)\in C_n} - \sum_{i,j=1}^n \alpha_i(x_j)\varphi(y_i)^\top\varphi(y_j).$$

B.4. IQP Implementation

In order to make our implementation possible for any symmetric loss $\ell : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}$, on a finite space $\mathcal{Y}$, we introduce the following decomposition.

Proposition 10 (Quadratic decomposition)**.** *When $\mathcal{Y}$ is finite, any proper symmetric loss ℓ admits a decomposition with two mappings $\varphi : \mathcal{Y} \to \mathbb{R}^m$, $\psi : \mathcal{Y} \to \mathbb{R}^m$, for a $m \in \mathbb{N}$ and a $c \in \mathbb{R}$, reading*

$$\forall\, y, z \in \mathcal{Y}, \quad \ell(y, z) = \psi(y)^\top\psi(z) - \varphi(y)^\top\varphi(z) \qquad \textit{with} \qquad \|\varphi(y)\| = \|\psi(y)\| = c \tag{15}$$

Proof. Consider $\mathcal{Y} = y_1, \cdot, y_m$ and $L = (\ell(y_i, y_j))_{i,j\leq m} \in \mathbb{R}^{m\times m}$. L is a symmetric matrix, diagonalizable as $L = \sum_{i=1}^m \lambda_i u_i \otimes u_i$, with (u_i) a orthonormal basis of $\mathbb{R}^m$, and $\lambda_i \in \mathbb{R}$ its eigen values. We have, with (e_i) the Cartesian basis of $\mathbb{R}^m$,

$$\ell(y_j, y_k) = L_{jk} = \left\langle e_j, Le_k \right\rangle = \sum_{i=1}^m (\lambda_i)_+ \left\langle e_j, u_i\right\rangle \left\langle e_k, u_i\right\rangle - \sum_{i=1}^m (\lambda_i)_- \left\langle e_j, u_i\right\rangle \left\langle e_k, u_i\right\rangle.$$

[5] The Diffrac algorithm was first introduced for clustering, which is a classical approach to unsupervised learning. In practice, it consists to change the constraint set $C_n = \prod S_i$ by a set of the type $C_n = \arg\max_{(y_i)\in\mathcal{Y}^n} \sum_{i,j=1}^n \mathbf{1}_{y_i\neq y_j}$ in Eqs. (4) and (14), meaning that (y_i) should be disambiguated into different classes.

[6] Since $\|\varphi(y)\|$ is constant.

We build the decomposition

$$\tilde\psi(y_k) = \left(\sqrt{(\lambda_i)_+}\left\langle e_k, u_i\right\rangle\right)_{i\leq m}, \qquad\text{and}\qquad \tilde\varphi(y_k) = \left(\sqrt{(\lambda_i)_-}\left\langle e_k, u_i\right\rangle\right)_{i\leq m}.$$

It satisfies $\ell(y_j, y_k) = \tilde\psi(y)^\top\tilde\psi(z) - \tilde\varphi(y)^\top\tilde\varphi(z)$. We only need to show that we can consider φ of constant norm. For this, first consider $C = \max_i\left|\lambda_i\right|$, we have $\left\|\tilde\psi(y_k)\right\|^2 = \sum_{i=1}^m (\lambda_i)_+\left\langle u_i, e_k\right\rangle^2 \leq C\sum_{i=1}^m\left\langle u_i, e_k\right\rangle^2 = C\left\|e_k\right\|^2 = C$ The last equalities being due to the fact that (u_i) is orthonormal. Now, introduce the correction vector $\xi : \mathcal{Y}\to\mathbb{R}^m$, $\xi(y_i) = \sqrt{C - \left\|\tilde\psi(y)\right\|^2}e_i$. And consider $\varphi = \binom{\tilde\varphi}{\xi}$, $\psi = \binom{\tilde\psi}{\xi}$. By construction, ψ is of constant norm being equal to C and that $\ell(y, z) = \psi(y)^T\psi(z) - \varphi(y)^T\varphi(z)$. Finally, because $\ell(y, z) = 0$, we also have φ of constant norm. □

Using the decomposition Eq. (15), Eq. (14) reads, with $\mathbf{y} = (y_i)$

$$\hat{\mathbf{y}} \in \operatorname*{arg\,min}_{\mathbf{y}\in C_n}\sum_{i=1}^n\alpha_i(x_j)\psi(y_i)\psi(y_j) - \sum_{i=1}^n\alpha_i(x_j)\varphi(y_i)\varphi(y_j).$$

By defining the matrix $A = (\alpha_i(x_j))_{ij\leq n}\in\mathbb{R}^{n\times n}$, $\boldsymbol{\Psi}(\mathbf{y}) = (\psi(y_i))_{i\leq n}\in\mathbb{R}^{n\times m}$ and $\boldsymbol{\Phi}(\mathbf{y}) = (\varphi(y_i))_{i\leq n}\in\mathbb{R}^{n\times m}$, we cast it as

$$\hat{\mathbf{y}} \in \operatorname*{arg\,min}_{\mathbf{y}\in C_n}\operatorname{Tr}\left(A\boldsymbol{\Psi}(\mathbf{y})\boldsymbol{\Psi}(\mathbf{y})^\top\right) - \operatorname{Tr}\left(A\boldsymbol{\Phi}(\mathbf{y})\boldsymbol{\Phi}(\mathbf{y})^\top\right).$$

Objective convexification. As $\alpha_i(x_j)$ is a measure of similarity between x_i and x_j, A is usually symmetric positive definite, making this objective convex in $\boldsymbol{\Psi}$ and concave in $\boldsymbol{\Phi}$. However, recalling Eq. (15), we have $\operatorname{Tr}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \operatorname{Tr}\boldsymbol{\Psi}\boldsymbol{\Psi}^\top = nc$, therefore considering the spectral norm of A, we convexify the objective as

$$\hat{\mathbf{y}} \in \operatorname*{arg\,min}_{\mathbf{y}\in C_n}\operatorname{Tr}\left((\left\|A\right\|_* I + A)\boldsymbol{\Psi}(\mathbf{y})\boldsymbol{\Psi}(\mathbf{y})^\top\right) + \operatorname{Tr}\left((\left\|A\right\|_* I - A)\boldsymbol{\Phi}(\mathbf{y})\boldsymbol{\Phi}(\mathbf{y})^\top\right).$$

Considering

$$B = \begin{pmatrix} \left\|A\right\|_* I + A & 0 \\ 0 & \left\|A\right\|_* I - A\end{pmatrix} \qquad\text{and}\qquad \Xi(\mathbf{y}) = \begin{pmatrix}\boldsymbol{\Psi}(\mathbf{y})\\ \boldsymbol{\Phi}(\mathbf{y})\end{pmatrix},$$

allow to simplify this objective as

$$\hat{\mathbf{y}} \in \operatorname*{arg\,min}_{\mathbf{y}\in C_n}\operatorname{Tr}\left(B\Xi(\mathbf{y})\Xi(\mathbf{y})^\top\right).$$

When parametrized by $\xi = \Xi(\mathbf{y})$, this is an optimization problem with a convex quadratic objective and "integer-like" constraint $\xi\in\Xi(C_n)$, identifying to an integer quadratic program (IQP).

Relaxation. IQP are known to be NP-hard, several tools exists in literature and optimization library implementing them. The most classical approach consists in relaxing the integer constraint $\xi\in\Xi(C_n)$ into the convex constraint $\xi\in\operatorname{Conv}(\Xi(C_n))$, solving the resulting convex quadratic program, and projecting back the solution towards an extreme of the convex set. Arguably, our alternative minimization approach is a better grounded heuristic to solve our specific disambiguation problem.

C. Example with graphical illustrations

To ease the understanding of the disambiguation principle (2), we provide a toy example with a graphical illustration, Figure 4. Since Eq. (2) decorrelates inputs, we will consider $\mathcal{X}$ to be a singleton, in order to remove the dependency to $\mathcal{X}$. In the following, we consider $\mathcal{Y} = \{a, b, c\}$, with the loss given by

$$L = (\ell(y, z))_{y,z\in\mathcal{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0\end{pmatrix}.$$

This problem can be represented on a triangle through the embedding of probability measures reading $\xi : \Delta_{\mathcal{Y}}\to\mathbb{R}^3; \mu\to\mu(a)e_1 + \mu(b)e_2 + \mu(c)e_3$, and onto the triangle $\left\{z\in\mathbb{R}_+^3\,\middle|\, z^\top 1 = 1\right\}$. Note that ξ can be extended from any signed measure

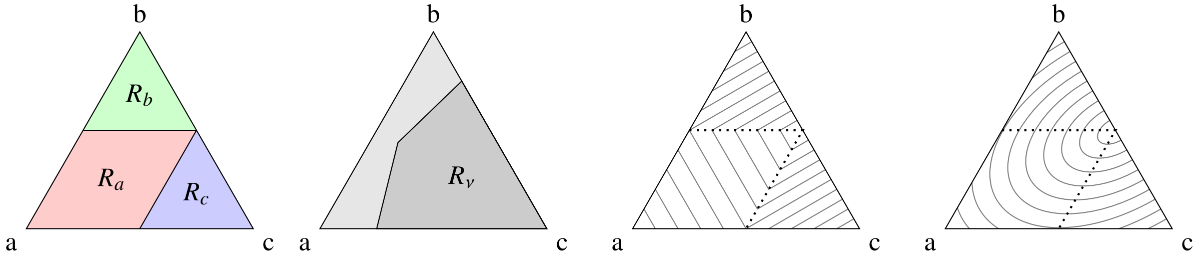

Figure 4. Exposition of a pointwise problem in the simplex $\Delta_{\mathcal{Y}}$, with $\mathcal{Y} = \{a, b, c\}$ and a proper symmetric loss defined by $\ell(a, b) = \ell(a, c) = \ell(b, c)/2$. (Left) Representation of the decision regions $R_z = \left\{\mu \in \Delta_{\mathcal{Y}} \,\middle|\, z \in \arg\min_{z' \in \mathcal{Y}} \mathbb{E}_{Y\sim\mu}[\ell(z, y)]\right\}$. for $z \in \mathcal{Y}$. (Middle Left) Representation of $R_\nu = \left\{\mu \in \Delta_{\mathcal{Y}} \,\middle|\, \mu \vdash \nu\right\}$ for $\nu = (5\delta_{\{a,b,c\}} + \delta_{\{c\}} + \delta_{\{a,c\}} + \delta_{\{b,c\}})/8$ (Middle Right) Level curves of the piecewise function $\Delta_{\mathcal{Y}} \to \mathbb{R}; \mu \to \min_{z\in\mathcal{Y}} \mathbb{E}_{Y\sim\mu}[\ell(z, Y)]$ corresponding to Eq. (2). (Right) Level curves of the quadratic function $\Delta_{\mathcal{Y}} \to \mathbb{R}; \mu \to \mathbb{E}_{Y,Y'\sim\mu}[\ell(Y, Y')]$. Our disambiguation (2) corresponds to minimizing the concave function represented on the middle right drawing on the convex domain represented on the middle left drawing.

of total mass normalized to one onto the plane $\left\{z \in \mathbb{R}^3 \,\middle|\, z^\top 1 = 1\right\}$, as well as the drawings Figure 4 can be extended onto the affine span of the represented triangles. The objective (2) reads pointwise as $\Delta_{\mathcal{Y}} \to \mathbb{R}; \mu \to \min_{i\leq 3} e_i^\top L\xi(\mu)$, while its quadratic version reads $\Delta_{\mathcal{Y}} \to \mathcal{Y}; \mu \to \xi(\mu)^\top L\xi(\mu)$. Note that while L is not definite negative, one can check that the restriction of $\mathbb{R}^3 \to \mathbb{R}; z \to z^\top Lz$ to the definition domain $\left\{z \in \mathbb{R}^3 \,\middle|\, z^\top 1 = 1\right\}$ is concave, as suggested by the right drawing of Figure 4.

It should be noted that (ℓ, ν) being a well-behaved partial labelling problem can be understood graphically, as having the intersection of the decision regions $\cap_{z\in S} R_z$ non-empty for any set S in the support of ν. As such, it is easy to see that our toy problem is well-behaved for any distribution ν. Formally, to match Definition 5, we can define $\mu_{\{e\}} = \delta_e$ for $e \in \{a, b, c\}$ and

$$\mu_{\{a,b\}} = .5\delta_a + .5\delta_b, \quad \mu_{\{a,c\}} = .5\delta_b + .5\delta_c, \quad \mu_{\{b,c\}} = \delta_b + \delta_c - \delta_a, \quad \mu_{\{a,b,c\}} = .5\delta_b + .5\delta_c.$$

Graphically $\xi(\mu_{\{a,b\}})$ can be chosen as any points on the horizontal dashed line on the middle right drawing of Figure 4 (similarly for $\xi\mu_{\{a,c\}}$), while $\xi(\mu_{\{a,b,c\}})$ has to be chosen has the intersection $.5e_2 + .5e_3$, and while $\xi(\mu_{\{b,c\}})$ has to be chosen outside the simplex on the half-line leaving $.5e_2 + .5e_3$ supported by the perpendicular bisector of $[e_2, e_3]$ and not containing e_1.

D. Experiments

While our results are much more theoretical than experimental, out of principle, as well as for reproducibility, comparison and usage sake, we detail our experiments.

D.1. Interval regression - Figure 1

Figure 1 corresponds to the regression setup consisting of learning $f^* : [0, 1] \to \mathbb{R}; x \to \sin(\omega x)$, with $\omega = 10 \approx 3\pi$. The dataset represented on Figure 1 is collected in the following way. We sample $(x_i)_{i\leq n}$ with $n = 10$, uniformly at random on $\mathcal{X} = [0, 1]$, after fixing a random seed for reproducibility. We collect $y_i = f(x_i)$. We create (s_i) by sampling u_i uniformly on $[0, 1]$, defining $r_i = r - \gamma \log(u_i)$, with $r = 1$ and $\gamma = 3^{-1}$, sampling c_i uniformly at random on $[0, r_i]$, and defining $s_i = y_i + \operatorname{sign}(y_i) \cdot c_i + [-r_i, r_i]$. The corruption is skewed on purpose to showcase disambiguation instability of the baseline (13) compared to our method. We solve Eq. (4) with alternative minimization, initialized by taking $y_i^{(0)}$ at the center of s_i, and stopping the minimization scheme when $\sum_{i\leq n} |y_i^{(t+1)} - y_i^{(t)}| < \varepsilon$ for ε a stopping criterion fixed to 10^{-6}. For $x \in \mathcal{X}$, the inference Eqs. (5) and (13) is done through grid search, considering, for $f_n(x)$, 1000 guesses dividing uniformly $[-6, 6] \subset \mathcal{Y} = \mathbb{R}$. We consider weights α given by kernel ridge regression with Gaussian kernel, defined as

$$\alpha(x) = (K + n\lambda I)^{-1}K_x \in \mathbb{R}^n, \quad K = (k(x_i, x_j))_{i,j\leq n} \in \mathbb{R}^{n\times n}, \quad K_x = (k(x_i, x))_{i\leq n} \in \mathbb{R}^n, \quad k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right),$$

with λ a regularization parameter, and σ a standard deviation parameter. In our simulation, we fix $\sigma = .1$ based on simple considerations on the data, while we consider $\lambda \in [10^{-1}, 10^{-3}, 10^{-6}]$. The evaluation of the mean square error between f_n

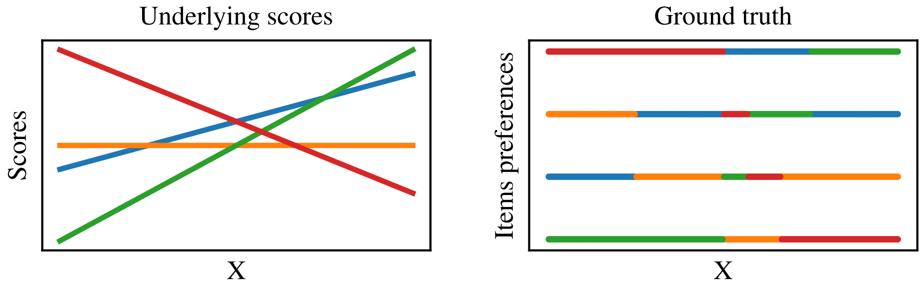

Figure 5. Ranking setting. We consider $\mathcal{X}$ an interval of $\mathbb{R}$, and $\mathcal{Y} = \mathfrak{S}_m$ with $m = 4$ on the figure. (Right) To create a ranking dataset, we sample randomly m lines in $\mathbb{R}^2$, embedding a value, or equivalently a score, associated to each items as a function of the input x. (Left) By ordering those lines, we create preferences between items as a function of x. On the figure, when x is small, the "red" item is prefered over the "orange" item, itself prefered over the "blue" item, itself prefered over the "green" item. While when x is big, "green" is prefered over "blue", prefered over "orange", prefered over "red". We create a partial labelling dataset by sampling $(x_i) \in \mathcal{X}^n$, and providing only partial ordering that the (y_i) follow. For example, for a small x, we might only give the partial information that "red" is prefered over "blue".

and f^*, which is equivalent to evaluating the risk with the regression loss $\ell(y, z) = \|y - z\|^2$, is done by considering 200 points dividing uniformly $\mathcal{X} = [0, 1]$ and evaluating f_n and f^* on it. The best hyperparameter λ is chosen by minimizing this error. It leads to $\lambda = 10^{-1}$ for the baseline (13), and $\lambda = 10^{-6}$ for our algorithm (4) and (5). This difference in λ is normal since both methods are not estimating the same surrogate quantities. The fact that λ is smaller for our algorithm is natural as our disambiguation objective (4) already has a regularization effect on the solution.[7] Note that we used the same weights α for Eq. (4) and Eq. (5), which is suboptimal, but fair to the baseline, as, consequently, both methods have the same number of hyperparameters.

D.2. Classification - Figure 2

Figure 2 corresponds to classification problems, based on real dataset from the LIBSVM datasets repository. At the time of writing, the datasets are available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html`. We present results on the "Dna" and "Svmguide2" datasets, that both have 3 classes ($m = 3$), and respectively have 4000 samples with 180 features ($n = 4000$,$d = 180$) and 391 samples with 20 features ($n = 391$, $d = 20$).

In term of *complexity*, when $\mathcal{Y} = [\![1, m]\!] = \{1, 2, \cdots, m\}$, and weights based on kernel ridge regression with Gaussian kernel as described in the last paragraph the complexity of performing inference for Eqs. (5) and (13) can be done in $O(nm)$ in time and $O(n + m)$ in space, where n is the number of training samples (Nowak-Vila et al., 2019; Cabannes et al., 2020). The disambiguation (4) performed with alternative minimization is done in $O(cn^2m)$ in time and in $O(n(n + m))$ in space, with c the number of steps in the alternative minimization scheme. In practice, c is really small, which can be understood since we are minimizing a concave function and each step leads to a guess on the border of the constraint domain.

Based on the dataset (x_i, y_i), we create (s_i) by sampling it accordingly to $\gamma\delta_{\{y_i\}} + 1 - \gamma\delta_{\{y, y_i\}}$, with y the most present labels in the dataset (indeed we choose the two datasets because they were not too big and presenting unequal labels proportion), and $\gamma \in [0, 1]$ the corruption parameter represented in percentage on the x-axis of Figure 2. This skewed corruption allows to distinguish methods and invalidate the simple approach consisting to averaging candidate (AC) in set to recover y_i from s_i, which works well when data are *missing at random* (Heitjan & Rubin, 1991). We separate (x_i, s_i) in 8 folds, consider $\sigma \in d \cdot [1, .1, .01]$, where d is the dimension of $\mathcal{X}$, and $\lambda \in n^{-1/2} \cdot [1, 10^{-3}, 10^{-6}]$, where n is the number of data. We test the different hyperparameter setup and reported the best error for each corruption parameter on Figure 2. Those errors are measured with the 0-1 loss, computed as averaged over the 8 folds, *i.e.* cross-validated, which standard deviation represented as errorbars on the figure. The best hyperparameter generally corresponds to $\sigma = .1$ and $\lambda = 10^{-3}$ when the corruption is small and $\sigma = 1$, $\lambda = 10^{-3}$ when the corruption is big. Differences between cross-validated error and testing error were small, and we presented the first one out of simplicity.

In term of *energy cost*, the experiments were run on a personal laptop that has two processors, each of them running

[7] Moreover, the analysis in Cabannes et al. (2020) suggests that the baseline is estimating a surrogate function in $\mathcal{X} \to 2^{\mathbb{R}}$, while our method is estimating a function in $\mathcal{X} \to \mathbb{R}$, which is a much smaller function space, hence needing less regularization. However, those reflections are based on upper bounds, that might be sub-optimal, which could invalidate those considerations.

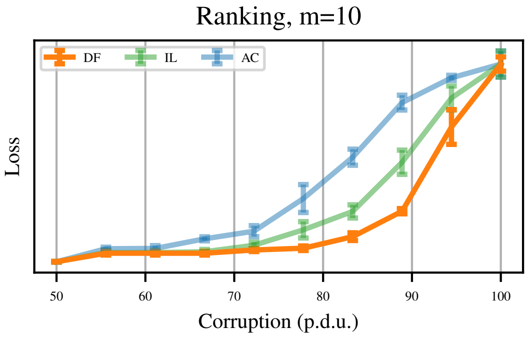

Figure 6. Performance of our algorithm for ranking with partial ordering. This figure is similar to Figure 2, but is based on the ranking problem illustrated on Figure 5. For this figure, we consider $m = 10$, as it is arguably the limit where the LP relaxation provided by Cabannes et al. (2020) of the NP-hard minimum feedback arcset problem still performs well. The corruption parameter corresponds to the proportion of coordinates lost in the Kendall embedding when creating s_i from y_i. Because the Kendall embedding satisfies transitivity constraints, a corruption smaller than 50% is almost ineffective to remove any information. In this figure, we observe a similar behavior for ranking to the one observed for classification on Figure 2, suggesting that those empirical findings are not spurious.

2.3 billion instructions per second. During experiments, all the data were stored on the random access memory of 8GB. Experiments were run on Python, extensively relying on the NumPy library (Harris et al., 2020). The heaviest computation is Figure 2. Its total runtime, cross-validation included, was around 70 seconds. This paper is the results of experimentations, we evaluate the total cost of our experimentations to be three orders of magnitude higher than the cost of reproducing the final computations presented on Figure 1, 2 and 3. The total computational energy cost is very negligible.

D.3. Semi-supervised learning - Figure 3

On Figure 3, we review a semi-supervised classification problem with $\mathcal{Y} = \llbracket 1, 4 \rrbracket$, $\mathcal{X} = [-4.5, 4.5]^2$, $\mu_{\mathcal{X}}$ only charging $\left\{x = (x_1, x_2) \in \mathbb{R}^2 \,\middle|\, x_1^2 + x_2^2 \in \mathbb{N}^*\right\}$ and the solution $f^* : \mathcal{X} \to \mathcal{Y}$ being defined almost everywhere as $f^*(x) = x_1^2 + x_2^2$. We collect a dataset (x_i, s_i), by sampling 2000 points θ_i uniformly at random on $[0, 1]$, as well as r_i uniformly at random in $\llbracket 1, 4 \rrbracket = \{1, 2, 3, 4\}$, before building $x_i = r_i \cdot (\cos(2\pi\theta_i), \sin(2\pi\theta_i)) \in \mathcal{X}$, and $s_i = \mathcal{Y}$. We add four labelled points to this dataset $x_{2001} = (-2\sqrt{3}, 2)$ with $s_{2001} = \{4\}$, $x_{2001} = (1, -2\sqrt{2})$ with $s_{2002} = \{3\}$, $x_{2001} = (-\sqrt{3}, -1)$ with $s_{2003} = \{2\}$ and $x_{2001} = (-1, 0)$ with $s_{2004} = \{1\}$. We designed the weights α in Eq. (4) with k-nearest neighbors, with $k = 20$, and solve this equation with a variant of alternative minimization, leading to the optimal solution $\tilde{y}_i = y_i^*$. In order to be able to compute the baseline (13), we design weights α for the inference task based on Nadaraya-Watson estimators with Gaussian kernel, defined as $\alpha_i(x) = \exp\left(\|x - x_i\|^2 / h\right)$, with $h = .08$. We solve the inference task on a grid of $\mathcal{X}$ composed of 2500 points, and artificially recreate the observation to make them neat and reduce the resulting pdf size. Note that it is possible to design weights α that capture the cluster structure of the data, which, in this case, will lead to a nice behavior of the baseline as well as our algorithm. Arguably, this experiment showcase a regularization property of our algorithm (4).

D.4. Ranking with partial ordering

To conclude this experiment section, we look at ranking with partial ordering. We refer to Section 5.4 for a clear description of this instance of partial labelling. We provide to the reader eager to use our method, an implementation of our algorithm, available online at `https://github.com/VivienCabannes/partial_labelling`. It is based on LP relaxation of the NP-hard minimum feedback arcset problem. This relaxation was proven exact when $m \leq 6$ by Cabannes et al. (2020). The LP implementation relies on CPLEX (IBM, 2017). As complementary experiments, we will not provide much reproducibility details, those details would be really similar to the previous paragraphs, and the curious reader could run our code instead. We present our ranking setup on Figure 5 and our results on Figure 6.