Consistent and Tractable Algorithm for Markov Network learning

Vojtech Franc✉[0000−0001−7189−1224], Daniel Prusa[0000−0003−4866−5709], and Andrii Yermakov[0000−0002−2173−5095]

Czech Technical University in Prague, Czech Republic
{xfrancv,prusa,yermaand}@fel.cvut.cz

Abstract. Markov network (MN) structured output classifiers provide a transparent and powerful way to model dependencies between output labels. The MN classifiers can be learned using the M3N algorithm, which, however, is not statistically consistent and requires expensive fully annotated examples. We propose an algorithm to learn MN classifiers that is based on Fisher-consistent adversarial loss minimization. Learning is transformed into a tractable convex optimization that is amenable to standard gradient methods. We also extend the algorithm to learn from examples with missing labels. We show that the extended algorithm remains convex, tractable, and statistically consistent.

1 Introduction

Structured output classification aims at the prediction of a set of statistically interdependent labels. A transparent way to model dependencies between the labels provides the Markov Network (MN) classifier, formally defined as follows. Let $\mathcal{X}$ be a set of observations. Let $\mathcal{V}$ be a finite set of objects, and let $\mathcal{E} \subseteq \binom{\mathcal{V}}{2}$ be a set of interacting objects. An object $v \in \mathcal{V}$ is characterized by a label $y \in \mathcal{Y}_v$ of a finite set $\mathcal{Y}_v$. Let $\mathcal{Y} = \times_{v \in \mathcal{V}} \mathcal{Y}_v$ be the structured output space, and let $\boldsymbol{y} = (y_y \in \mathcal{Y}_v \mid v \in \mathcal{V}) \in \mathcal{Y}$ denote the labeling of all objects in $\mathcal{V}$. The match between observation x and a label $y_v \in \mathcal{Y}$ assigned to object $v \in \mathcal{V}$ is scored by a function $f_v \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. The match between the labels $(y_v, y_{v'})$ assigned to the interacting objects $\{v, v'\} \in \mathcal{E}$ is scored by a function $f_{vv'} \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Given an observation $x \in \mathcal{X}$, the MN classifier $\boldsymbol{h} \colon \mathcal{X} \to \mathcal{Y}$ returns the labeling $\boldsymbol{y} \in \mathcal{Y}$ with the maximum total score:

$$\boldsymbol{h}(x) \in \operatorname*{Argmax}_{\boldsymbol{y} \in \mathcal{Y}} \Big[\sum_{v \in \mathcal{V}} f_v(x, y_v) + \sum_{v, v' \in \mathcal{E}} f_{vv'}(y_v, y_{v'}) \Big] \tag{1}$$

Inference (1) requires solving a valued constrained satisfaction problem which is an NP-hard in general. There are subclasses solvable efficiently; e.g., when $(\mathcal{V}, \mathcal{E})$ is acyclic, it can be solved by dynamic programming. In a general setup, the problem can be addressed using linear programming (LP) relaxation [19].

Linearly parameterized score functions can be learned efficiently by Maximum Margin Markov Network (M3N) algorithms [14, 15, 17] even if the graph

$(\mathcal{V}, \mathcal{E})$ is generic and the inference is not tractable [5, 4]. The original M3N algorithm requires fully annotated examples; however, an extension for learning from partially annotated examples, when some labels can be missing, was proposed in [6]. M3N algorithms translate learning into convex and tractable minimization of the margin rescaling loss and its variants, which serve as a surrogate of the target loss we would like to actually optimize. An unsettling issue is the statistical properties. Namely, algorithms based on margin rescaling loss minimization are not statistically consistent [10]; that is, they are not guaranteed to learn the optimal Bayes classifier even if fed in with an unlimited amount of data.

Recently, [2, 3] proposed an adversarial loss whose minimization yields a statistically consistent algorithm. Unfortunately, the evaluation of the adversarial loss requires solving a Min-Max problem whose size scales with the number of labels, and thus it is not tractable for structured prediction. In [12] an algorithm was proposed that minimizes adversarial loss instantiated for structured predictors. However, the algorithm relies on an oracle that solves a Min-Max problem of the same complexity as the one in the definition of adversarial loss. Therefore, the algorithm is applicable only for MN classifiers when the neighborhood graph $(\mathcal{V}, \mathcal{E})$ is restricted to be acyclic, and even then the oracle is not guaranteed to solve the problem optimally.

In this paper, we contribute to the problem of learning MN classifiers by:

1. We propose a novel surrogate loss, named MArkov Network Adversarial (MANA) loss, for learning MN classifiers. The MANA loss is defined by a convex optimization which is tractable for general neighborhood graph $(\mathcal{V}, \mathcal{E})$. In Theorem 3 we prove that the MANA loss is equivalent to the adversarial loss. The MANA loss is, to our knowledge, the first surrogate for learning generic MN classifiers which is simultaneously statistically consistent, convex and tractable. Minimization of the MANA loss is amenable to standard gradient methods.
2. We extend the MANA loss for learning MN classifiers on partially annotated examples when the labels are missing at random. The extended loss, named partial MANA loss, has the same computational complexity as its supervised counterpart. In Theorem 5 we prove that the partial MANA loss is Fisher consistent.
3. We evaluate the algorithms minimizing margin-rescaling loss and the proposed MANA loss using both fully annotated and partially annotated data sets. We show that the empirical performance of both losses is similar. This find is not that surprising because we also show that the margin rescaling loss is a close approximation of the consistent MANA loss, although both surrogates were developed from completely different principles.

The necessary background and state-of-the-art is given in Section 2. The contributions of this paper are presented in Section 3. Section 4 provides an empirical evaluation of the proposed and existing methods, and Section 5 concludes the paper. Proofs of the novel Theorems 3, 4 and 5 are deferred to the supplementary material.

2 State-of-the-art

In this section, we describe the state-of-the-art in risk minimization approaches applicable to learning MN classifiers. We survey existing surrogate losses and describe which are Fisher-consistent and which are tractable when applied for learning the MN classifier. Section 2.1 focuses on supervised learning, and Section 2.2 on learning from partially annotated examples.

2.1 Supervised learning

Assume that instances $(x, \boldsymbol{y})$ are generated from a distribution $p_{XY}(x, \boldsymbol{y})$ on $\mathcal{X} \times \mathcal{Y}$. Let $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a target loss penalizing the predictions of the labeling. In this paper, we focus on additive losses, that is,

$$\ell(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{v \in \mathcal{V}} \ell_v(y_v, \hat{y}_{v'}) \tag{2}$$

where $\ell_v\colon \mathcal{Y}_v \times \mathcal{Y}_v \to \mathbb{R}$ are some single-label losses. The goal is to find a classifier $\boldsymbol{h}\colon \mathcal{X} \to \mathcal{Y}$ that minimizes the expected risk [1]:

$$R_\ell(\boldsymbol{h}, p_{XY}) = \mathbb{E}_{x,\boldsymbol{y} \sim p_{XY}} \ell(\boldsymbol{y}, \boldsymbol{h}(x))\,.$$

At best we achieve the Bayes risk $R^*_\ell(p_{XY}) = \inf_{\boldsymbol{h}\colon \mathcal{X} \to \mathcal{Y}} R_\ell(\boldsymbol{h}, p_{XY})$. The classifier is usually modeled as a composed function $\boldsymbol{h}(x) = \boldsymbol{T} \circ \boldsymbol{f}(x)$, where $\boldsymbol{f}\colon \mathcal{X} \to \mathbb{R}^d$ is a score map, and $\boldsymbol{T}\colon \mathbb{R}^d \to \mathcal{Y}$ is a fixed label decoding. For example, the most common prediction model, also considered in this paper, assigns labels based on maximization of a score function $f\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, i.e., $\boldsymbol{h}(x) \in \operatorname{Argmax}_{\boldsymbol{y} \in \mathcal{Y}} f(x, \boldsymbol{y})$. This corresponds to $d = |\mathcal{Y}|$ and $\boldsymbol{T}(\boldsymbol{f}) \in \operatorname{Argmax}_{\boldsymbol{y} \in \mathcal{Y}} f_{\boldsymbol{y}}$ where $\boldsymbol{f}(x) = (f(x, \boldsymbol{y}), \boldsymbol{y} \in \mathcal{Y}) \in \mathbb{R}^{|\mathcal{Y}|}$. The MN classifier (1) is obtained when $f(x, \boldsymbol{y})$ decomposes over objects $\mathcal{V}$ and edges $\mathcal{E}$, i.e.,

$$f(x, \boldsymbol{y}) = \sum_{v \in \mathcal{V}} f_v(x, y_v) + \sum_{v,v' \in \mathcal{E}} f_{vv'}(y_v, y_{v'})\,. \tag{3}$$

The finding of the classifier can be posed as a minimization of $R_\ell(\boldsymbol{T} \circ \boldsymbol{f})$ w.r.t. $\boldsymbol{f}$. However, direct minimization of the ℓ-risk is difficult due to the discrete nature of the commonly used losses ℓ. Therefore, ℓ is replaced by a surrogate loss $\psi\colon \mathbb{R}^d \times \mathcal{Y} \to \mathbb{R}$, which evaluates the score map $\boldsymbol{f}$ on $p_{XY}(x, \boldsymbol{y})$ using a ψ-risk $R_\psi(\boldsymbol{f}, p_{XY}) = \mathbb{E}_{x,\boldsymbol{y} \sim p_{XY}} \psi(\boldsymbol{f}(x), \boldsymbol{y})$. The optimal score w.r.t. the ψ-risk is then $\boldsymbol{f}_\psi \in \operatorname{Argmin}_{\boldsymbol{f}\colon \mathcal{X} \to \mathbb{R}^d} R_\psi(\boldsymbol{f}, p_{XY})$. There are two requirements on the surrogate loss ψ. First, optimization of the surrogate should be tractable, hence, $\psi(\boldsymbol{f}, \boldsymbol{y})$ is designed to be convex in $\boldsymbol{f}$ and cheap to evaluate. Second, the resulting classifier $\boldsymbol{h}(x) = \boldsymbol{T} \circ \boldsymbol{f}_\psi(x)$ should achieve low ℓ-risk, being our true objective. Ideally, we require the surrogate ψ to be Fisher consistent [9, 20]:

$$R^*_\ell(p_{XY}) = R_\ell(\boldsymbol{T} \circ \boldsymbol{f}_\psi, p_{XY})\,, \tag{4}$$

[1] We refer to the expectation of a loss LOSS as the LOSS-risk.

i.e., the classifier found by minimizing the ψ-risk achieves the Bayes ℓ-risk.

In the common ML setup, the distribution $p_{XY}(x, \boldsymbol{y})$ is unknown; however, we have a training sequence $\mathcal{T}_{XY} = ((x^i, \boldsymbol{y}^i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m)$ drawn from i.i.d. random variables with distribution $p_{XY}(x, \boldsymbol{y})$. Training data $\mathcal{T}_{XY}$ are used to approximate $p_{XY}(x, \boldsymbol{y})$ by the empirical distribution $\hat{p}^m_{XY}(x, \boldsymbol{y}) = \frac{1}{m} \sum_{i=1}^m [\![x = x^i \wedge \boldsymbol{y} = \boldsymbol{y}^i]\!]$, and the classifier is found using the empirical risk minimization (ERM)

$$\boldsymbol{f}^m_\psi \in \operatorname*{Argmin}_{\boldsymbol{f} \in \mathcal{F}} R_\psi(\boldsymbol{f}, \hat{p}^m_{XY}) \,, \tag{5}$$

where $\mathcal{F} \subseteq \{\boldsymbol{f} \colon \mathcal{X} \to \mathbb{R}^d\}$ is an a priori chosen class of functions. Under suitable conditions, with an increasing number of examples m, the population ψ-risk converges in probability to the minimal attainable ψ-risk, i.e., $R_\psi(\boldsymbol{f}^m_\psi, p_{XY}) \xrightarrow{p} R_\psi(\boldsymbol{f}^*_\psi, p_{XY})$. In this case, a Fisher-consistent surrogate ψ (i.e., the surrogate satisfying (4)), which is also continuous and bounded from below, guarantees the convergence of the ℓ-risk to the Bayes ℓ-risk, i.e. $R_\ell(\boldsymbol{T} \circ \boldsymbol{f}^m_\psi, p_{XY}) \xrightarrow{p} R^*_\ell(p_{XY})$ [20].

Structured Output Support Vector Machines [16, 15, 17] (SO-SVM) is an instance of the ERM, that is designed to learn the linear classifier and the surrogate is a certain convex piecewise linear function.

Let $\boldsymbol{\phi} \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ be an input-output feature map that embeds $\mathcal{X} \times \mathcal{Y}$ in a parameter space $\mathbb{R}^n$. Let $f(x, \boldsymbol{y}) = \boldsymbol{\phi}(x)^T \boldsymbol{\theta}$ be the score function parameterized by $\boldsymbol{\theta} \in \mathbb{R}^n$. We will use $\boldsymbol{\Phi}(x) = (\boldsymbol{\phi}(x, \boldsymbol{y}), \boldsymbol{y} \in \mathcal{Y}) \in \mathbb{R}^{n \times |\mathcal{Y}|}$ to denote a matrix that for a given $x \in \mathcal{X}$ contains the feature maps of all labelings $\boldsymbol{y} \in \mathcal{Y}$. Let $\boldsymbol{T}(\boldsymbol{f}) \in \operatorname{Argmax}_{\boldsymbol{y} \in \mathcal{Y}} f_{\boldsymbol{y}}$ be the label decoding and $\boldsymbol{f}(x) = (f(x, \boldsymbol{y}), \boldsymbol{y} \in \mathcal{Y}) \in \mathbb{R}^{|\mathcal{Y}|}$ be the score map. The linear classifier can be written in a compact way as

$$\boldsymbol{h}(x) \in \operatorname*{Argmax}_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{\phi}(x, \boldsymbol{y})^T \boldsymbol{\theta} = \boldsymbol{T} \circ \boldsymbol{f}(x) = \boldsymbol{T} \circ \boldsymbol{\Phi}(x)^T \boldsymbol{\theta} \,. \tag{6}$$

In this paper, we concentrate on a linear MN classifier, obtained when the input-output feature map decomposes over objects $\mathcal{V}$ and edges $\mathcal{E}$ as

$$\boldsymbol{\phi}(x, \boldsymbol{y}) = \sum_{v \in \mathcal{V}} \boldsymbol{\phi}_v(x, y_v) + \sum_{v, v' \in \mathcal{E}} \boldsymbol{\phi}_{vv'}(y_v, y_{v'}) \,, \tag{7}$$

where $\boldsymbol{\phi}_v \colon \mathcal{X} \times \mathcal{Y}_v \to \mathbb{R}^n$, $v \in \mathcal{V}$, and $\boldsymbol{\phi}_{vv'} \colon \mathcal{Y}_v \times \mathcal{Y}_{v'} \to \mathbb{R}^n$, $\{v, v'\} \in \mathcal{E}$.

Marginal rescaling loss is the most widely used surrogate in structured output classification [17], and it is defined as

$$\psi_{\text{mr}}(\boldsymbol{f}, \boldsymbol{y}) = \max_{\boldsymbol{y}' \in \mathcal{Y}} \big[\ell(\boldsymbol{y}, \boldsymbol{y}') + f(x, \boldsymbol{y}')\big] - f(x, \boldsymbol{y}) \,. \tag{8}$$

Given training examples $\mathcal{T}_{XY}$, the SO-SVM algorithm finds parameters $\boldsymbol{\theta}$ of the linear classifier (6) by solving a convex unconstrained problem

$$\boldsymbol{\theta}^m_{mr} = \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^n} \left[\frac{\lambda}{2} \|\boldsymbol{\theta}\|^2 + \frac{1}{m} \sum_{i=1}^m \psi_{\text{mr}}(\boldsymbol{\Phi}(\boldsymbol{x}^i)^T \boldsymbol{\theta}, \boldsymbol{y}^i) \right], \tag{9}$$

where $\lambda > 0$ is the regularization constant. The SO-SVM problem (9) corresponds to the ERM (5) with proxy ψ_{mr} and a class of linear scores $\mathcal{F} = \{\boldsymbol{f}(x) = \boldsymbol{\Phi}(x)^T\boldsymbol{\theta} \mid \|\boldsymbol{\theta}\| \leqslant r(\lambda)\}$, where $r\colon \mathbb{R} \to \mathbb{R}$ is a monotonic function of λ.

There are two issues with the SO-SVM algorithm. First, the margin-rescaling loss ψ_{mr} is not Fisher-consistent, in general. It is Fisher-consistent in the binary case $|\mathcal{Y}| = 2$, when ψ_{mr} becomes the hinge loss of the binary SVM [9]. In the multiclass case, $|\mathcal{Y}| > 2$, it is Fisher-consistent if only if $\max_{\boldsymbol{y}\in\mathcal{Y}} p_{Y|X}(\boldsymbol{y} \mid x) > 0.5$, $\forall x \in \mathcal{X}$ [10]. Second, evaluating the margin-rescaling loss requires an oracle solving the loss-augmented prediction

$$\hat{\boldsymbol{y}} \in \operatorname*{Argmax}_{\boldsymbol{y}\in\mathcal{Y}} \left[\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) + f(x, \boldsymbol{y})\right]. \tag{10}$$

In the case of the MN classifier, (10) is intractable, in general. It is tractable when the target loss is additive and the neighborhood graph $(\mathcal{V}, \mathcal{E})$ is restricted to be acyclic, in which case (10) can be solved by dynamic programming. The intractability of loss-augmented prediction can be resolved by replacing the intractable maximization problem (10) with a linear programming (LP) upper bound [13, 19], which was done for a generic MN classifier in [4]. The linear programming margin-rescaling (LP-MR) loss for the MN classifier (1) reads

$$\begin{aligned}\psi_{\text{lp}}(\boldsymbol{f}, \hat{\boldsymbol{y}}) = \min_{\boldsymbol{\alpha}\in\mathbb{R}^{n_\alpha}} \Bigg[&\sum_{v\in\mathcal{V}} \max_{y\in\mathcal{Y}_v} \Big[f_v(x, y) - \sum_{v'\in\mathcal{N}(v)} \alpha_{vv'}(y) + \ell_v(\hat{y}_v, y)\Big] \\ &+ \sum_{\{v,v'\}\in\mathcal{E}} \max_{(y,y')\in\mathcal{Y}_v\times\mathcal{Y}_{v'}} \Big[f_{vv'}(y, y') + \alpha_{vv'}(y) + \alpha_{v'v}(y')\Big]\Bigg] - f(x, \hat{\boldsymbol{y}}),\end{aligned} \tag{11}$$

where $\boldsymbol{\alpha} = (\alpha_{vv'}\colon \mathcal{Y}_v \times \mathcal{Y}_{v'} \to \mathbb{R}, \alpha_{v'v}\colon \mathcal{Y}_{v'} \times \mathcal{Y}_v \to \mathbb{R}, \{v, v'\} \in \mathcal{E})$ is a vector of $n_\alpha = 2\sum_{\{v,v'\}\in\mathcal{E}}(|\mathcal{Y}_v| + |\mathcal{Y}_{v'}|)$ auxiliary variables. Evaluating $\psi_{\text{lp}}(\boldsymbol{f}, \hat{\boldsymbol{y}})$ requires solving a convex unconstrained problem, which can be done using gradient methods simultaneously with learning the score function. In contrast to the original margin-rescaling loss, the optimization of ψ_{lp} is tractable for an arbitrary neighborhood graph $(\mathcal{V}, \mathcal{E})$. In the case of the acyclic graph $(\mathcal{V}, \mathcal{E})$, the bound is tight and $\psi_{\text{lp}}(\boldsymbol{f}, \boldsymbol{y}) = \psi_{\text{mr}}(\boldsymbol{f}, \boldsymbol{y})$, $\forall \boldsymbol{f}, \boldsymbol{y}$. Therefore, ψ_{lp} is not Fisher consistent in general.

Adversarial loss [2, 3] posed the prediction as an adversarial problem between the predictor minimizing the risk and an adversarial maximizing the risk with respect to the posterior distribution that matches the statistics computed on the examples. They show that adversarial prediction is an example of the risk minimization approach. In this case, the adversarial surrogate loss is expressed as a Min-Max problem:

$$\psi_{\text{adv}}(\boldsymbol{f}, \hat{\boldsymbol{y}}) = \max_{\boldsymbol{q}\in\Delta} \min_{\boldsymbol{p}\in\Delta} \mathbb{E}_{\boldsymbol{y}\sim\boldsymbol{p}, \boldsymbol{y}'\sim\boldsymbol{q}}\left[\ell(\boldsymbol{y}, \boldsymbol{y}') + f(x, \boldsymbol{y}) - f(x, \hat{\boldsymbol{y}})\right] \tag{12}$$

where $\Delta = \{\boldsymbol{q} \in \mathbb{R}_+^{|\mathcal{Y}|} \mid \sum_{\boldsymbol{y}\in\mathcal{Y}} q(\boldsymbol{y}) = 1\}$ is a probability simplex on $\mathcal{Y}$. The adversarial loss is Fisher-consistent (Theorem 15 in [3]):

Theorem 1. *Let* $R_{\mathrm{adv}}(\boldsymbol{f}, p_{XY}) = \mathbb{E}_{x,\boldsymbol{y}\sim p_{XY}} \psi_{\mathrm{adv}}(\boldsymbol{f}(x), \boldsymbol{y})$ *be* ψ_{adv}*-risk given by the adversarial loss (12), induced from a target loss* $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ *satisfying* $\ell(\boldsymbol{y}, \boldsymbol{y}) < \ell(\boldsymbol{y}, \boldsymbol{y}')$, $\forall \boldsymbol{y} \neq \boldsymbol{y}'$. *Let the set of optimal predictions* $\hat{\mathcal{Y}}(x) = \operatorname{Argmin}_{\boldsymbol{y}'\in\mathcal{Y}} \mathbb{E}_{\boldsymbol{y}\sim p_{Y|X}} \ell(\boldsymbol{y}, \boldsymbol{y}')$ *be a singleton,* $|\hat{\mathcal{Y}}(x)| = 1$*, for all inputs* $x \in \mathcal{X}$*. Then, we have*

$$R^*_\ell(p_{XY}) = R_\ell(\boldsymbol{T} \circ \boldsymbol{f}_{\mathrm{adv}}, p_{XY})\,,$$

where $\boldsymbol{T}(x) \in \operatorname{Argmax}_{\boldsymbol{y}\in\mathcal{Y}} f_{\boldsymbol{y}}$ *and* $\boldsymbol{f}_{\mathrm{adv}} \in \operatorname{Argmin}_{\boldsymbol{f}\colon \mathcal{X}\to\mathbb{R}^{|\mathcal{Y}|}} R_{\mathrm{adv}}(\boldsymbol{f}, p_{XY})$ *is a minimizer of the* ψ_{adv}*-risk with respect to all measurable functions.*

Nice statistical properties of the adversarial loss are paid off by the computational issues, i.e., evaluating the loss (12) requires solving the Min-Max problem with $2|\mathcal{Y}|$ variables, which in case of the structured prediction is intractable. A generalized Block Coordinate Frank-Wolfe (GBCFW) algorithm to learn structured output linear classifiers by regularized ERM with the adversarial loss was recently proposed in [12]. The GBCFW relies on an oracle solving a Min-Max problem of as similar complexity as (12). [12] propose an alternating procedure to solve the Min-Max approximately which, however, has no guarantee to reach a global optimum and, in case of the MN classifier it is tractable only when the neighbourhood graph $(\mathcal{V}, \mathcal{E})$ is restricted to be acyclic.

2.2 Learning from partially annotated examples

Assume that we do not have access to full labeling $\boldsymbol{y} \in \mathcal{Y} = \times_{v\in\mathcal{V}} \mathcal{Y}_v$ but instead we obtain an annotation $\boldsymbol{a} \in \mathcal{A} = \times_{v\in\mathcal{V}} \mathcal{A}_v$ where $\mathcal{A}_v = \{\mathcal{Y}_v \cup \{?\}\}$. That is, for an object $v \in \mathcal{V}$ we either know the true label, $a_v = y_v$, or the label is not given, $a_v =?$. Given the instance $(x, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$ generated from $p_{XY}(x, \boldsymbol{y})$, the annotation $\boldsymbol{a} \in \mathcal{A}$ is generated from $p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y})$. A partially annotated training set $\mathcal{T}_{XA} = \{(x^i, \boldsymbol{a}^i) \in \mathcal{X} \times \mathcal{A} \mid i = 1, \ldots, m\}$ contains examples drawn from i.i.d. random variables with distribution

$$p_{XA}(x, \boldsymbol{a}) = \sum_{\boldsymbol{y}\in\mathcal{Y}} p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y})\, p_{XY}(x, \boldsymbol{y})\,. \tag{13}$$

The goal is to use $\mathcal{T}_{XA}$ to learn a classifier with ℓ-risk $R_\ell(\boldsymbol{h}, p_{XY})$ close to the Bayes ℓ-risk $R^*_\ell(p_{XY})$. That is, the goals of supervised learning and learning from partially annotated examples are the same, but the training sets are different.

To apply the risk minimization approach, we need a surrogate loss $\psi^p\colon \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}$, whose value $\psi^p(\boldsymbol{f}, \boldsymbol{a})$ evaluates the score map $\boldsymbol{f}\colon \mathcal{X} \to \mathbb{R}^d$ based on the partial annotation $\boldsymbol{a} \in \mathcal{A}$. Let us define the ψ^p-risk $R_{\psi^p}(\boldsymbol{f}, p_{XA}) = \mathbb{E}_{x,\boldsymbol{a}\sim p_{XA}} \psi^p(\boldsymbol{f}(x), \boldsymbol{a})$. An optimal score under ψ^p-risk is obtained by solving

$$\boldsymbol{f}_{\psi^p} \in \operatorname*{Argmin}_{\boldsymbol{f}\colon \mathcal{X}\to\mathbb{R}^d} R_{\psi^p}(\boldsymbol{f}, p_{XA})\,.$$

As in the supervised case, we require the surrogate ψ^p to be tractable and Fisher-consistent:

$$R^*_\ell(p_{XY}) = R_\ell(\boldsymbol{T} \circ \boldsymbol{f}_{\psi^p}, p_{XY})\,,$$

i.e., the classifier found by minimizing the ψ^p-risk on $p_{XA}(x, \boldsymbol{a})$, achieves the Bayes ℓ-risk on $p_{XY}(x, \boldsymbol{y})$.

Labels missing at random The distribution $p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y})$ governing the annotation process cannot be arbitrary to make the learning possible. For example, when $p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y}) = p_A(\boldsymbol{a})$, the annotation $\boldsymbol{a}$ is useless as it carries no information about the labeling $\boldsymbol{y}$. In this paper, we consider labels Missing At Random (MAR) annotation process [1, 6] defined by

$$p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y}) = \sum_{\boldsymbol{z} \in \{0,1\}^{\mathcal{V}}} p_{Z|X}(\boldsymbol{z} \mid x) \prod_{v \in \mathcal{V}} [\![a_v = c(y_v, z_v)]\!] , \tag{14}$$

where $c(y_v, z_v) = y_v$ if $z_v = 1$, $c(y_v, z_v) =?$ if $z_v = 0$, and $p_{Z|X}(\boldsymbol{z} \mid x)$, $x \in \mathcal{X}$, are conditional distributions on $\boldsymbol{z} \in \{0,1\}^{\mathcal{V}}$ such that $p_{Z_v|X}(z_v \mid x) > 0$, $\forall v \in \mathcal{V}$. The MAR process implies that the annotation in $\mathcal{T}_{XA}$ is generated as follows. Nature generates $(x, \boldsymbol{y})$ from $p_{XY}(x, \boldsymbol{y})$. The annotator decides the objects to label based on the observation of the input x. His decision is stochastic, represented by a binary vector $\boldsymbol{z} \in \{0,1\}^{\mathcal{V}}$ generated from $p_{Z|X}(\boldsymbol{z} \mid x)$. The annotator reveals the labels of the objects $\mathcal{V}_{lab} = \{v \in \mathcal{V} \mid z_v = 1\}$, i.e., he sets $a_v = y_v$, $v \in \mathcal{V}_{lab}$, while the labels of the remaining objects are not provided, i.e., $a_v =?$, $v \in \mathcal{V} \backslash \mathcal{V}_{lab}$.

Ramp-loss SO-SVM was extended to learn from partially annotated examples in [11]. The method uses the Ramp loss defined as

$$\psi^p_{\text{ramp}}(\boldsymbol{f}, \boldsymbol{a}) = \max_{\boldsymbol{y} \in \mathcal{Y}} \left[\ell^p(\boldsymbol{a}, \boldsymbol{y}) + f(x, \boldsymbol{y}) \right] - \max_{\boldsymbol{y} \in \mathcal{Y}} f(x, \boldsymbol{y})$$

where $\ell^p(\boldsymbol{a}, \boldsymbol{y}) = \sum_{v \in \mathcal{V}} [\![a_v \neq ?]\!] \ell_v(a_v, y_v)$ is the partial additive loss. In case of the MAR annotation, the ramp-loss is Fisher-consistent [1]. However, the ramp-loss is non-convex, and in case of the score of the MN-classifier even its evaluation is not tractable in general. Unlike the margin-rescaling loss, the LP upper bound is not applicable here.

Partial LP margin-rescaling loss Partial LP margin-rescaling loss for learning linear MN classifiers from partially annotated examples was proposed in [6]. The loss reads [2]

$$\begin{aligned} \psi^p_{\text{lp}}(x, \boldsymbol{\theta}, \boldsymbol{a}) = \min_{\boldsymbol{\alpha} \in \mathbb{R}^{n_\alpha}} \Bigg[& \sum_{v \in \mathcal{V}} \max_{y \in \mathcal{Y}_v} \Big[\boldsymbol{\phi}_v(x, y)^T \boldsymbol{\theta} - \sum_{v' \in \mathcal{N}(v)} \alpha_{vv'}(y) + \ell_v(\hat{y}_v, y) \Big] \\ & + \sum_{\{v, v'\} \in \mathcal{E}} \max_{(y, y') \in \mathcal{Y}_v \times \mathcal{Y}_{v'}} \Big[\boldsymbol{\phi}_{vv'}(y, y')^T \boldsymbol{\theta} + \alpha_{vv'}(y) + \alpha_{v'v}(y') \Big] \Bigg] - \boldsymbol{\phi}^p(x, \boldsymbol{a})^T \boldsymbol{\theta} \,, \end{aligned} \tag{15}$$

where $\boldsymbol{\phi}^p \colon \mathcal{X} \times \mathcal{A} \to \mathbb{R}^n$ is input-annotation feature map defined as

$$\boldsymbol{\phi}^p(x, \boldsymbol{a}) = \sum_{v \in \mathcal{V}} \frac{[\![a_v \neq ?]\!]}{p_{Z_v|X}(1 \mid x)} \boldsymbol{\phi}_v(x, y_v) + \sum_{v, v' \in \mathcal{E}} \frac{[\![a_v \neq ? \wedge a_{v'} \neq ?]\!]}{p_{Z_v, Z_{v'}|X}(1, 1 \mid x)} \boldsymbol{\phi}_{vv'}(y_v, y_{v'}) \,. \tag{16}$$

[2] To emphasize that ψ^p_{lp} is applicable only for the linear MN classifier, we use the notation $\psi^p_{\text{lp}}(x, \boldsymbol{\theta}, \boldsymbol{a})$ instead of $\psi^p_{\text{lp}}(\boldsymbol{f}, \boldsymbol{a})$ with $\boldsymbol{f}(x) = \boldsymbol{\Phi}(x)^T \boldsymbol{\theta}$.

The loss ψ^p_{lp} is obtained from the LP-MR loss (11) by replacing the correct labeling score $f(x, \boldsymbol{y}) = \boldsymbol{\phi}(x, \boldsymbol{y})^T \boldsymbol{\theta}$, which cannot be computed since the complete labeling $\boldsymbol{y}$ is unknown, by the score $\boldsymbol{\phi}^p(x, \boldsymbol{a})^T \boldsymbol{\theta}$ which can be computed on the partial annotation $\boldsymbol{a}$. The replacement is justified by the fact that the expectation of the input-output features equals the expectation of the input-annotation features as stated by the following theorem (Theorem 1 in [6]):

Theorem 2. *Let $\boldsymbol{\phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ be the input-output feature map defined by (7) and $\boldsymbol{\phi}^p\colon \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ the input-annotation feature map defined by (16). Let both $\boldsymbol{\phi}$ and $\boldsymbol{\phi}^p$ be constructed from the same set of $\boldsymbol{\phi}_v\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, $v \in \mathcal{V}$, and $\boldsymbol{\phi}_{vv'}\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^d$, $\{v, v'\} \in \mathcal{E}$. Let $p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y})$ be the MAR annotation process (14). Then, we have*

$$\mathbb{E}_{\boldsymbol{a} \sim p_{A|X}} \boldsymbol{\phi}^p(x, \boldsymbol{a}) = \mathbb{E}_{\boldsymbol{y} \sim p_{Y|X}} \boldsymbol{\phi}(x, \boldsymbol{y}) \,, \qquad \forall x \in \mathcal{X} \,,$$

where $p_{A|X}(\boldsymbol{a} \mid x)$ and $p_{Y|X}(\boldsymbol{y} \mid x)$ are conditional distributions derived from $p_{XYA}(x, \boldsymbol{y}, \boldsymbol{a}) = p_{XY}(x, \boldsymbol{y})\, p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y})$.

Note that computation of the input-annotation feature map (16) requires the unary marginals $p_{Z_v|X}(z_v \mid \boldsymbol{x})$, $v \in \mathcal{V}$, and pair-wise marginals $p_{Z_v, Z_{v'}|X}(z_v, z_{v'} \mid \boldsymbol{x})$, $\{v, v'\} \in \mathcal{E}$, of the distribution $p_{Z|X}(\boldsymbol{z} \mid x)$ describing the label missingness. The marginals can be easily estimated from the partially annotated examples $\mathcal{T}_{XA}$ using the maximum likelihood method [6].

The partial LP-MR loss ψ^p_{lp} is convex, and it can be efficiently optimized by gradient methods. In the limit case, when no labels are missing, it coincides with the supervised margin-rescaling loss, and hence, it is not Fisher-consistent.

3 Contributions

3.1 Tractable adversarial loss for the MN classifier

The additive loss (2) and the score $f(x, \boldsymbol{y})$ of the MN classifier (3), both decompose as a sum of functions with arity at most two. We noticed that in this case the Min-Max problem that defines adversarial loss (12) can be converted to a linear program whose dual form is tractable. This leads to a novel surrogate loss, termed the MArkov Network Adversarial (MANA) loss, which is defined as a tractable convex optimization

$$\begin{aligned}\psi_{\text{mana}}(\boldsymbol{f}, \hat{\boldsymbol{y}}) = \min_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^{n_\alpha} \\ \boldsymbol{\mu} \in \mathcal{M}}} \Bigg[& \sum_{v \in \mathcal{V}} \max_{y \in \mathcal{Y}_v} \Big[f_v(x, y) - \sum_{v' \in \mathcal{N}(v)} \alpha_{vv'}(y) + \sum_{y' \in \mathcal{A}} \mu_v(y') \ell_v(y_v, y') \Big] \\ & + \sum_{\{v, v'\} \in \mathcal{E}} \max_{(y, y') \in \mathcal{Y}_v \times \mathcal{Y}_{v'}} \Big[f_{vv'}(y, y') + \alpha_{vv'}(y) + \alpha_{v'v}(y') \Big] \Bigg] - f(x, \hat{\boldsymbol{y}}) \,,\end{aligned} \tag{17}$$

where the vector $\boldsymbol{\alpha} = (\alpha_{vv'}\colon \mathcal{Y}_v \times \mathcal{Y}_{v'} \to \mathbb{R}\,, \alpha_{v'v}\colon \mathcal{Y}_{v'} \times \mathcal{Y}_v \to \mathbb{R}\,, \{v, v'\} \in \mathcal{E})$ has $n_\alpha = 2 \sum_{\{v, v'\} \in \mathcal{E}} (|\mathcal{Y}_v| + |\mathcal{Y}_{v'}|)$ variables, the vector $\boldsymbol{\mu} = (\boldsymbol{\mu}_v \in \Delta_v\,, v \in \mathcal{V}) \in \mathcal{M} \subset \mathbb{R}^{n_\mu}$ is composed of vectors $\boldsymbol{\mu}_v \in \Delta_v$, $v \in \mathcal{V}$, from the probability simplex

on $\mathcal{Y}_v$ and it has $n_\mu = \sum_{v\in\mathcal{V}} |\mathcal{Y}_v|$ variables in total. Note that evaluating the objective of the minimization problem (17) for fixed $(\boldsymbol{\alpha}, \boldsymbol{\mu})$ does not require any oracle to solve an intractable problem, unlike the algorithm of [12]. The following theorem, one of the main results of this paper, ensures that the MANA loss (17) coincides with the Fisher consistent adversarial loss (12).

Theorem 3. *Let $\ell\colon \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$ be an additive loss (2). Let $\mathcal{F} = \{\boldsymbol{f}\colon \mathcal{X}\to\mathbb{R}^{|\mathcal{Y}|}\}$ be a set composed of the MN classifier score maps given by (3). Then, we have*

$$\psi_{\text{adv}}(\boldsymbol{f}, \boldsymbol{y}) = \psi_{\text{mana}}(\boldsymbol{f}, \boldsymbol{y})\,, \quad \forall \boldsymbol{f}\in\mathcal{F}, \forall \boldsymbol{y}\in\mathcal{Y}\,.$$

Comparison with the LP margin-rescaling loss We would like to point out a striking similarity between the MANA loss (17) and the LP-MR loss (11) although they were derived from completely different principles. The MANA loss can be obtained from the LP-MR loss by replacing the ground truth labels in the maximization terms of (11) by their one-hot encodings, and minimizing the value of the loss w.r.t those encodings. Or, equivalently, fixing the values of $\boldsymbol{\mu}_v(y)$, $v\in\mathcal{V}$, in (17) to one-hot encoding of ground truth labels $\hat{y}_v$, $v\in\mathcal{V}$, instead of minimizing them, makes MANA loss equal to the LP-MR loss. This subtle change makes the inconsistent LP-MR loss to Fisher-consistent MANA loss without significantly increasing the computational complexity. However, the LP-MR loss can be seen as a close approximation of the consistent MANA loss which also provides additional explanation for its good empirically observed performance.

MANA as unconstrained convex optimization Most frequently, the single-label losses $\ell_v\colon \mathcal{Y}_v\times\mathcal{Y}_v\to\mathbb{R}$, $v\in\mathcal{V}$, defining the additive loss $\ell(\boldsymbol{y}, \boldsymbol{y}')$ are normalized $0/1$ losses, e.g., when $\ell(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} [\![y_v \neq \hat{y}_{v'}]\!]$ is the Hamming loss. In this case, the MANA loss (17) can be simplified by eliminating the variables $\boldsymbol{\mu}$ as stated in the following theorem.

Theorem 4. *Let $\ell\colon \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$ be an additive loss (2) composed of $\ell_v(y, y') = K_v [\![y\neq y']\!]$, $v\in\mathcal{V}$, where $K_v > 0$, $v\in\mathcal{V}$, are positive scalars. Then*

$$\psi_{\text{mana}}(\boldsymbol{f}, \hat{\boldsymbol{y}}) = \min_{\boldsymbol{\alpha}\in\mathbb{R}^{n_\alpha}} \Bigg[\sum_{v\in\mathcal{V}} \max_{S\subseteq\mathcal{Y}_v, |S|>0} \Bigg[\frac{1}{|S|}\sum_{y\in S}\Big[f_v(x,y) - \sum_{v'\in\mathcal{N}(v)} \alpha_{vv'}(y)\Big] + K_v - \frac{K_v}{|S|}\Bigg] + \sum_{\{v,v'\}\in\mathcal{E}} \max_{(y,y')\in\mathcal{Y}_v\times\mathcal{Y}_{v'}} \Big[f_{vv'}(y,y') + \alpha_{vv'}(y) + \alpha_{v'v}(y')\Big]\Bigg] - f(x, \hat{\boldsymbol{y}})\,. \tag{18}$$

Remark 1. The inner maximization in (18) is of type

$$\max_{S\subseteq\mathcal{Y}_v, |S|>0} \left[\frac{1}{|S|}\sum_{y\in S} g_v(x,y) + K_v - \frac{K_v}{|S|}\right]$$

and it can be solved in $\mathcal{O}(|\mathcal{Y}_v| \log |\mathcal{Y}_v|)$ time as follows. First, sort the values $g_v(x, y)$, $y \in \mathcal{Y}_v$, in non-increasing order into a sequence $a_1, \ldots, a_{|\mathcal{Y}_v|}$. Second, compute $k^* \in \operatorname{Argmax}_{k \in \{1,\ldots,|\mathcal{Y}_v|\}} [\frac{1}{k} \sum_{i=1}^{k} a_i + K_v - \frac{K_v}{k}]$. Third, construct the optimal set S from the first k^* labels in the sorted order.

Using the MANA loss (18) as a surrogate in the regularized ERM problem (9), leads to an unconstrained convex optimization with $\mathcal{O}(m \cdot n_\alpha + n)$ variables. The optimization problem can be solved efficiently using standard sub-gradient methods.

3.2 Fisher-consistent surrogate for partially annotated examples

In this section, we extend the MANA for learning the linear MN classifier on partially annotated examples. In particular, we assume the linear predictor (6) with the score map $\boldsymbol{f}(x) = \boldsymbol{\Phi}(x)^T \boldsymbol{\theta}$ given by the parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ and the input-output feature map (7). We further assume that the partial annotations are generated by the MAR process (14). For this setting, we propose the partial MANA loss defined as

$$\psi^p_{\text{mana}}(x, \boldsymbol{\theta}, \boldsymbol{a}) = \min_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^{n_\alpha} \\ \boldsymbol{\mu} \in \mathcal{M}}} \Bigg[\sum_{v \in \mathcal{V}} \max_{y \in \mathcal{Y}_v} \Big[\boldsymbol{\theta}^T \boldsymbol{\phi}_v(x, y) - \sum_{v' \in \mathcal{N}(v)} \alpha_{vv'}(y) + \sum_{y' \in \mathcal{A}} \mu_v(y') \ell_v(y_v, y') \Big] + \sum_{\{v,v'\} \in \mathcal{E}} \max_{(y,y') \in \mathcal{Y}_v \times \mathcal{Y}_{v'}} \Big[\boldsymbol{\theta}^T \boldsymbol{\phi}_{vv'}(y, y') + \alpha_{vv'}(y) + \alpha_{v'v}(y') \Big] \Bigg] - \boldsymbol{\theta}^T \boldsymbol{\phi}^p(x, \boldsymbol{a}) \Big) \tag{19}$$

where $\boldsymbol{\phi}^p(x, \boldsymbol{a})$ is the input-annotation feature map (16). The partial MANA loss (19) is obtained from the (supervised) MANA loss (17) after substituting the linear score $f(x, \boldsymbol{y}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(x, \boldsymbol{y})$ and replacing the correct labeling score (the last term in (17)), which cannot be evaluated as the complete labeling $\boldsymbol{y}$ is unknown, by $\boldsymbol{\theta}^T \boldsymbol{\phi}^p(x, \boldsymbol{a})$, which can be evaluated on a partial annotation $\boldsymbol{a}$. Note that the partial MANA loss has the exact same computational complexity as the (supervised) MANA loss. In the case of fully annotated examples, it follows from (16) that $\boldsymbol{\phi}^p(x, \boldsymbol{a}) = \boldsymbol{\phi}(x, \boldsymbol{y})$, and hence both losses coincide.

The following theorem, another main contribution of this paper, ensures that the partial MANA loss is Fisher-consistent.

Theorem 5. *Assume the same setup as in Theorem 1. In addition, assume that:*

1. *The partially annotated examples $(x, \boldsymbol{a}) \in \mathcal{X} \times \mathcal{A}$ are generated from $p_{XA}(x, \boldsymbol{a}) = \sum_{\boldsymbol{y} \in \mathcal{Y}} p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y})\, p_{XY}(x, \boldsymbol{y})$ where $p_{A|XY}(\boldsymbol{a} \mid x, \boldsymbol{y})$ is a MAR annotation process (14).*
2. *The set $\mathcal{F} = \{\boldsymbol{f}(x) = \boldsymbol{\Phi}(x)^T \boldsymbol{\theta} \mid \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^n\}$ contains score maps of linear MN classifier (7), and $\mathcal{F} \supset \operatorname{Argmin}_{\boldsymbol{f}\colon \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}} R_{\text{adv}}(\boldsymbol{f}, p_{XY})$.*

Then, we have

$$R^\ell_*(p_{XY}) = R^\ell(\boldsymbol{T} \circ \boldsymbol{\Phi}(x)^T \boldsymbol{\theta}^p_{\text{mana}}, p_{XY})\,,$$

where $\boldsymbol{T}(x) \in \operatorname{Argmax}_{\boldsymbol{y} \in \mathcal{Y}} f_{\boldsymbol{y}}$ and $\boldsymbol{\theta}^p_{\text{mana}} \in \operatorname{Argmin}_{\boldsymbol{\theta} \in \Theta} R^p_{\text{mana}}(\boldsymbol{\theta}, p_{XA})$ is a minimizer of $R^p_{\text{mana}}(\boldsymbol{\theta}, p_{XA}) = \mathbb{E}_{x, \boldsymbol{a} \sim p_{XA}} \psi^p_{\text{mana}}(x, \boldsymbol{\theta}, \boldsymbol{a})$.

The theorem guarantees that the linear MN classifier $\boldsymbol{h}(x) = \boldsymbol{T} \circ \boldsymbol{\Phi}(x)^T \boldsymbol{\theta}^p_{\text{mana}}$ with parameters found minimizing ψ^p_{mana}-risk on $p_{XA}(x, \boldsymbol{a})$ achieves the Bayes ℓ-risk on $p_{XY}(x, \boldsymbol{y})$. The theorem requires the annotations to be generated from MAR process (14), and that the class of linear scores $\mathcal{F}$ is sufficiently rich to contain a minimizer of the ψ_{adv}-risk.

4 Experiments

We evaluate the precision of a linear MN classifier trained by solving the regularized ERM problem (9) with different surrogate losses. In all experiments, we use the normalized Hamming loss, $\ell(\boldsymbol{y}, \boldsymbol{y}') = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} [\![y_v \neq y'_v]\!]$, as the target loss plugged into the surrogates. We evaluate the following algorithms:

1. The baseline, referred to as the M3N algorithm, solves (9) with the MR-LP surrogate (11) when learning from fully annotated examples, and the partial MR-LP surrogate (15) when learning from the partial annotations. When $(\mathcal{V}, \mathcal{E})$ is a chain and the examples are fully annotated, the algorithm becomes the standard Maximum Margin Markov network algorithm [16, 17]. The generalization for an arbitrary graph $(\mathcal{V}, \mathcal{E})$ was proposed in [4]. The generalization for partially annotated examples comes from [6]. In all cases, the surrogates are derived from the margin rescaling loss (8), hence we use the M3N algorithm for all the variants.
2. The proposed algorithm, referred to as MANA algorithm, solves (9) with the MANA surrogate (18) when learning from fully annotated examples and partial loss of MANA (19) when learning from partial annotations.

As benchmark problems, in Section 4.1 we consider the prediction of sequences generated from the hidden Markov chain, and in Section 4.2 the prediction of the solution of the Sudoku puzzle [6].

Inference The inference of the MN classifier (1) is solved by the dynamic programming when $(\mathcal{V}, \mathcal{E})$ is a chain. For general $(\mathcal{V}, \mathcal{E})$, we use the Augmented Directed Acyclic Graph solver [13, 19].

Optimization Regardless of the surrogate used, ERM (9) leads to convex unconstrained optimization with the same number of variables. We solve ERM (9) using ADAM [7] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, 5000 passes through all m training examples, and the learning rate $\frac{1}{100\,\mathrm{t}}$, $t \in \{1, \ldots, 5000\,m\}$.

Computation of partial losses Partial loss of LP-MR (11) and Partial MANA loss (19) require knowing the marginals of the distribution $p_{Z|X}(\boldsymbol{z} \mid x)$ that govern the missingness of the labels. Following [6], we assume that the distribution is homogeneous and the labels are missing completely at random, that is,

$$p_{Z|X}(\boldsymbol{z} \mid x) = \prod_{v \in \mathcal{V}} \tau^t (1 - \tau)^t \tag{20}$$

where $t = \sum_{v \in \mathcal{V}} z_v$ and $\tau \in [0, 1]$ is the probability that a randomly chosen object $v \in \mathcal{V}$ is annotated. Under this assumption, marginals can be estimated from the partially annotated examples $\mathcal{T}_{XA}$ using the maximum likelihood approach:

$$\begin{array}{c} \hat{p}_{Z_v|X}(1 \mid x) = \tau, \quad v \in \mathcal{V}, \\ \hat{p}_{Z_v,Z_{v'}|X}(1,1 \mid x) = \tau^2, \ \{v, v'\} \in \mathcal{E}\,, \end{array} \quad \text{where} \quad \tau = \frac{1}{m|\mathcal{V}|} \sum_{i=1}^{m} \sum_{v \in \mathcal{V}} [\![a_v^i \neq ?]\!] \,. \tag{21}$$

The estimated marginals are used to calculate $\boldsymbol{\phi}^p(x, \boldsymbol{a})$ defined by (16).

Evaluation protocol For each data set, we generate K random divisions of the examples into training, validation, and testing parts. The training part is used to learn the parameters $\boldsymbol{\theta}$. The optimal regularization constant $\lambda \in \{0, 1, 10, 100\}$ selected based on the minimal Hamming loss evaluated on the validation= part. We report the mean and standard deviation of the Hamming loss and the $0/1$ loss of the model with the optimal λ calculated on the K example divisions.

4.1 Synthetic data: Hidden Markov chain

The input and output are sequences of symbols $\boldsymbol{x} = (x_1, \ldots, x_{100}) \in \{1, \ldots, 30\}^{100}$ and $\boldsymbol{y} = (y_1, \ldots, y_{100}) \in \{1, \ldots, 30\}^{100}$ generated from the hidden Markov chain:

$$p_{XY}(\boldsymbol{x}, \boldsymbol{y}) = p(y_1) \prod_{i=2}^{100} p(y_i \mid y_{i-1}) p(x_i \mid y_i) \,. \tag{22}$$

The initial state distribution $p(y_1)$ is randomly generated, the emission probability is $p(x_i \mid y_i) = 7/10$ if $x_i = y_i$ and $p(x_i \mid y_i) = 3/290$ otherwise, and the transition probability is $p(y_i \mid y_{i-1}) = 7/10$ if $y_i = y_{i-1}$ and $p(y_i \mid y_{i-1}) = 3/290$ otherwise. The known model allows us to construct the Bayes classifier, optimal for the Hamming loss. The Bayes risk estimated from 100,000 examples is 0.2013.

We generate the partial annotation $\boldsymbol{a} \in (\{1, \ldots, 30\} \cup \{?\})^{100}$ using $p(\boldsymbol{a} \mid \boldsymbol{x}, \boldsymbol{y})$ given by (14), and the missingness distribution $p(\boldsymbol{z} \mid \boldsymbol{x})$ given by (20). We vary the probability $\tau \in \{0, 0.1, 0.2\}$ to generate the complete annotation and partial annotations with 10% and 20% labels missing at random. The graph $(\mathcal{V}, \mathcal{E})$ is a chain. The feature maps $\boldsymbol{\psi}_v(\boldsymbol{x}, y) = \mathbf{1}_{x_v, y}$, and $\boldsymbol{\psi}_{vv'}(y, y') = \mathbf{1}_{y, y'}$, are one-hot encodings of the symbols (x_v, y) and (y, y'), respectively. We used $K = 5$ random divisions of the data. The test set has 10,000 examples, the validation 5000 examples, and the size of the training set was $m \in \{10, 100, 1000\}$.

The test error of the MN classifier for different sizes of the training set and different amounts of missing labels is summarized in Table 1. The errors obtained for the M3N and MANA algorithms are very similar. The M3N performs slightly better when the number of training examples is small, while the MANA performs slightly better when the number of training examples is high. Differences become more pronounced with a greater number of missing labels. The best test risk 0.2050 ± 0.0005, obtained with the MANA algorithm on 1,000 fully annotated examples, is close to the Bayes risk 0.2013 estimated from 100,000 examples and using the ground truth model (22) to construct the Bayes predictor.

Table 1. Test error of linear MN classifiers predicting sequences of symbols generated by hidden Markov chain. The classifiers are trained by M3N and MANA algorithm on varying number of training examples with varying amount of missing labels.

			M3N	MANA
		#trn	Test error Hamming loss	Test error Hamming loss
missing labels	0%	10	0.3500 ± 0.0124	0.3764 ± 0.0152
		100	0.2351 ± 0.0007	0.2319 ± 0.0016
		1000	0.2094 ± 0.0008	0.2050 ± 0.0005
	10%	10	0.3495 ± 0.0189	0.3528 ± 0.0154
		100	0.2441 ± 0.0023	0.2420 ± 0.0018
		1000	0.2117 ± 0.0008	0.2065 ± 0.0007
	20%	10	0.3409 ± 0.0200	0.3423 ± 0.0177
		100	0.2547 ± 0.0017	0.2526 ± 0.0017
		1000	0.2135 ± 0.0008	0.2078 ± 0.0007

4.2 Sudoku solver

Symbolic inputs The Sudoku is made up of 9×9 cells $\mathcal{V} = \{(i,j) \in \mathcal{N} \mid 1 \leqslant i \leqslant 9, 1 \leqslant j \leqslant 9\}$ filled with numbers 1 to 9 or kept empty $\square$. The puzzle assignment is $\boldsymbol{x} = (x_v \in \{\square, 1, \ldots, 9\} \mid v \in \mathcal{V})$. The task is to fill the empty cells so that the rows, columns, and non-overlapping subgrids 3×3 contain all numbers from 1 to 9. The puzzle solution is $\boldsymbol{y} = (y_v \in \{1, \ldots, 9\} \mid v \in \mathcal{V})$. Prior knowledge is encoded by revealing the algorithm that cells in rows, columns, and 3×3 sub-grids are related, that is, by setting $\mathcal{E} = \{\{(v,v'),(u,u')\} \mid v = v' \vee v' = u' \vee (\lceil v/3 \rceil = \lceil u/3 \rceil \wedge \lceil v'/3 \rceil = \lceil u'/3 \rceil)\}$. The feature maps $\boldsymbol{\psi}_v(\boldsymbol{x}, y) = \mathbf{1}_{x_v, y}$, and $\boldsymbol{\psi}_{vv'}(y, y') = \mathbf{1}_{y,y'}$, are one-hot encodings of the pair of symbols (x_v, y) and (y, y'), respectively.

We use a database of Sudoku assignments and their correct solutions to create a training set. The partial annotation/solution was generated using the MAR process (14) with $p_{Z_v|X}(1 \mid \boldsymbol{x}) = 1$ if $x_v \in \{1, \ldots, 9\}$ and $p_{Z_v|X}(1 \mid \boldsymbol{x}) = 1 - \tau$ if $x_v = \square$, where $\tau \in \{0, 0.1, 0.2\}$ is the probability that the empty cell is not annotated. We generate three training sets with a complete solution, with 10% and 20% of the empty cells left empty, respectively. We varied the number of training examples $m \in \{10, 100, 1000\}$. We tested on 100 puzzles; note that it involves the prediction of $9 \cdot 9 \cdot 100 = 8,100$ labels. In addition to Hamming loss, we also evaluated the prediction using the 0/1 loss, in which case the test error corresponds to the portion of puzzles that were not solved perfectly.

The results are summarized in Table 2. The precisions obtained for the M3N and MANA algorithm are similar. It was enough to use $m = 100$ training examples to reach zero test error regardless of the amount of missing labels used. In the case of $m = 10$ training examples, the differences are at the level of the standard deviation for both 0/1 loss and Hamming loss.

Table 2. Test error of linear MN classifiers predicting solution of Sudoku puzzle from either symbolic assignment or visual assignment composed of the MNIST digits. The classifiers are trained by M3N and the proposed MANA algorithm on varying number of training examples with varying amount of missing labels.

Symbolic Sudoku

			M3N		MANA	
			Test error		Test error	
		#trn	0/1-loss [%]	Hamming loss	0/1-loss [%]	Hamming loss
missing labels	0%	10	0.6±0.9	0.0021±0.0029	0.6±0.5	0.0021±0.0020
		100	0.0±0.0	0.0000±0.0000	0.0±0.0	0.0000±0.0000
	10%	10	0.6±0.9	0.0018±0.0028	0.4±0.5	0.0013±0.0018
		100	0.0±0.0	0.0000±0.0000	0.0±0.0	0.0000±0.0000
	20%	10	0.6±0.9	0.0018±0.0028	1.0±1.2	0.0031±0.0038
		100	0.0±0.0	0.0000±0.0000	0.0±0.0	0.0000±0.0000

Visual Sudoku

			M3N		MANA	
			Test error		Test error	
		#trn	0/1-loss [%]	Hamming loss	0/1-loss [%]	Hamming loss
missing labels	0%	10	96.2±1.8	0.4407± 0.0070	96.8±1.3	0.4475± 0.0201
		100	19.2±4.3	0.0625± 0.0153	20.4±3.9	0.0710± 0.0160
		1000	5.8±1.3	0.0149± 0.0035	5.8±0.8	0.0155± 0.0037
	10%	10	95.6±2.6	0.4402±0.0155	96.2±2.4	0.4512±0.0106
		100	36.2±4.9	0.1254±0.0205	42.6±4.7	0.1467±0.0238
		1000	37.2±4.8	0.0928±0.0195	40.6±3.8	0.0952±0.0160
	20%	10	97.6±2.5	0.4557±0.0213	98.0±1.9	0.4643± 0.0129
		100	46.2±2.3	0.1593±0.0120	50.4±3.4	0.1706± 0.0150
		1000	52.4±3.2	0.1260±0.0184	52.8±3.3	0.1261± 0.0180

MNIST digits used as input We replace the input symbols $\{1, \ldots, 9\}$ with 28×28 images of handwritten digits from the MNIST data set [8]. The empty cells are replaced by all-black images. As a feature map of the unary scores, we use $\boldsymbol{\psi}_v(\boldsymbol{x}, y) = (\bar{\boldsymbol{\psi}}_1, \ldots, \bar{\boldsymbol{\psi}}_9)$, where $\bar{\boldsymbol{\psi}}_{y'} \in \mathbb{R}^{2000}$, $y' \neq y$, are all-zero vectors, $\bar{\boldsymbol{\psi}}_y = (k(\boldsymbol{x}_v, \boldsymbol{\mu}_1), \ldots, k(\boldsymbol{x}_v, \boldsymbol{\mu}_{2000})) \in \mathbb{R}^{2000}$ is a vector of RBF kernels $k(\boldsymbol{x}_v, \boldsymbol{\mu}_i) = \exp(-2\|\boldsymbol{x}_v - \boldsymbol{\mu}_i\|^2)$ evaluated for the image $\boldsymbol{x}_v$ of the v-th cell and 2,000 randomly sampled training images. All other settings are the same as for the symbolic Sudoku experiment. The results are summarized in Table 2. The M3N algorithm achieves slightly better results when the number of training examples is small. For $m = 1000$, the differences are at the standard deviation level.

Comparison with neural architectures Learning deep NN to solve Sudoku was considered in [18]. They used both the symbolic and the MNIST digits as inputs. They trained the SATNet architecture, which is a CNN with a maximum satisfiability (MAXSAT) solver as the last layer. SATNet can better learn hard interactions between output variables than canonical neural architectures (ConvNet) used as a baseline. They use $9,000$ fully annotated training examples and $1,000$ test examples. Table 3 presents the portion of incorrectly predicted solu-

tions of the test Sudoku puzzles. For comparison, we include the performance of linear MN classifiers trained with the M3N and MANA algorithm on 1,000 completely annotated examples. Although the MN classifier is trained on a smaller number of examples, it significantly outperforms both neural architectures in both symbolic and visual Sudoku.

Table 3. Comparison of the MN classifier trained from 1,000 examples, and neural architectures trained from 9,000 examples on the problem of predicting Sudoku solution from symbolic and visual assignments composed of MNIST digits.

	Test error, 0/1-loss [%]	
Method	Symbolic	Visual
MN classifier - M3N	0.0±0.0	5.8±1.3
MN classifier - MANA	0.0±0.0	5.8±0.8
ConvNet	84.9	99.9
SATNet	1.7	63.8

5 Conclusions

We proposed a novel surrogate loss, the MANA loss, to train MN classifiers. Minimizing MANA loss leads to tractable convex optimization that is amenable to standard gradient methods. We prove that the MANA loss is equivalent to the adversarial loss defined by the Min-Max problem, which is Fisher consistent but intractable in the context of the structure prediction. To our knowledge, the proposed MANA loss is the first surrogate for learning MN classifiers with a generic neighborhood graph that is simultaneously statistically consistent, convex, and tractable. This is not an obvious result because even an evaluation of a generic MN classifier leads to discrete optimization, which is intractable, in general.

We also proposed a partial MANA loss applicable to learning linear MN classifiers on partially annotated examples when the labels are missing at random. The partial MANA loss has the same computational complexity as its supervised counterpart, and we prove that the partial MANA loss is also Fisher-consistent.

The experiments show that the empirical performance of the ERM algorithms minimizing the MANA loss, which is consistent, and the LP margin scaling loss, which is not consistent, are comparable. The deviations are usually at the level of the estimation error. The comparable performance is not that surprising, because we have also shown that the LP margin rescaling loss is a close approximation of the MANA loss, although both surrogates were originally developed from completely different principles.

The code and data are available at: https://github.com/xfrancv/manet

Acknowledgments

The research was supported by the Czech Science Foundation project GACR GA19-21198S and OP VVV project CZ.02.1.01\0.0\0.0\16 019\0000765 Research Center for Informatics.

References

1. Antoniuk, K., Franc, V., Hlaváč, V.: Consistency of structured output learning with missing labels. In: Asian Conference on Machine Learning (ACML) (2015)
2. Fathony, R., Liu, A., Asif, K., Ziebart, B.: Adversarial multiclass classification: A risk minimization perspective. In: NIPS (2016)
3. Fathony, R., Asif, K., Liu, A., Bashiri, M.A., Xing, W., Behpour, S., Zhang, X., Ziebart, B.D.: Consistent robust adversarial prediction for general multiclass classification (2018), https://arxiv.org/abs/1812.07526
4. Franc, V., Laskov, P.: Learning maximal margin markov networks via tractable convex optimization. Control Systems and Computers (2), 25–34 (2011)
5. Franc, V., Savchynskyy, B.: Discriminative learning of max-sum classifiers. Journal of Machine Learning Research **9**(1), 67–104 (2008)
6. Franc, V., Yermakov, A.: Learning maximum margin markov networks from examples with missing labels. In: Asian Conference on Machine Learning (2021)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. of International Conference on Learning Representations (ICLR) (2015)
8. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
9. Lin, Y.: A note on margin-based loss functions in classification. Statistics & Probability Letters **68**(1), 73–82 (2004)
10. Liu, Y.: Fisher consistency of multicategory support vector machines. In: International Conference on Artificial Intelligence and Statistics. pp. 291–298 (2007)
11. Lou, X., Hamprecht, F.A.: Structured learning from partial annotations. In: International Conference on Machine Learning (ICML). pp. 1519–1526 (2012)
12. Nowak, A., Bach, F., Rudi, A.: Consistent structured prediction with max-min margin markov networks. In: International Conference on Machine Learning (2020)
13. Schlesinger, M.: Syntactic analysis of two-dimensional visual signals in noisy conditions. Kibernetika (4), 113–130 (1976), in Russian
14. Taskar, B., Chatalbashev, V., Koller, D.: Learning associative markov networks. In: International Conference on Machine Learning (ICML) (2004)
15. Taskar, B., Guestrin, C., Koller, D.: Maximum-margin markov networks. In: Proc. of Neural Information Processing Systems (NIPS) (2004)
16. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: NIPS (2003)
17. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research **6**(50), 1453–1484 (2005)
18. Wang, P., Donti, P., Wilder, B., Kolter, J.: SATnet: Bridging deep learning and logical reasoning using a differential satisfiability solver. In: ICML (2019)
19. Werner, T.: A linear programming approach to max-sum problem: A review. IEEE Trans. on Pattern Analysis and Machine Intelligence **29**(7), 1165–1179 (2007)
20. Yhang, T.: Statistical analysis of some multi-category large margin classification methods. Journal of Machine Learning Research **5** (2004)