Online Multiple Kernel Learning for Structured Prediction

André F. T. Martins*† Noah A. Smith* Eric P. Xing*
Pedro M. Q. Aguiar‡ Mário A. T. Figueiredo†
{afm,nasmith,epxing}@cs.cmu.edu
aguiar@isr.ist.utl.pt, mtf@lx.it.pt

*School of Computer Science
Carnegie Mellon University, Pittsburgh, PA, USA

†Instituto de Telecomunicações
Instituto Superior Técnico, Lisboa, Portugal

‡Instituto de Sistemas e Robótica
Instituto Superior Técnico, Lisboa, Portugal

October 15, 2010

Abstract

Despite the recent progress towards efficient multiple kernel learning (MKL), the structured output case remains an open research front. Current approaches involve repeatedly solving a batch learning problem, which makes them inadequate for large scale scenarios. We propose a new family of *online* proximal algorithms for MKL (as well as for group-LASSO and variants thereof), which overcomes that drawback. We show regret, convergence, and generalization bounds for the proposed method. Experiments on handwriting recognition and dependency parsing testify for the successfulness of the approach.

1 Introduction

Structured prediction (Lafferty et al., 2001; Taskar et al., 2003; Tsochantaridis et al., 2004) deals with problems with a strong interdependence among the output variables, often with sequential, graphical, or combinatorial structure. Despite recent advances toward a unified formalism, obtaining a good predictor often requires a significant effort in designing kernels (*i.e.*, features and similarity measures) and tuning hyperparameters. The slowness in training structured predictors in large scale settings makes this an expensive process.

The need for careful kernel engineering can be sidestepped using the kernel learning approach initiated in Bach et al. (2004); Lanckriet et al. (2004), where a combination of multiple kernels is learned from the data. While multi-class and scalable multiple kernel learning (MKL) algorithms have been proposed (Sonnenburg et al., 2006; Zien and Ong, 2007; Rakotomamonjy et al., 2008; Chapelle and Rakotomamonjy, 2008; Xu et al., 2009; Suzuki and Tomioka, 2009), none are well suited for large-scale structured prediction, for the following reason: all involve an inner loop in which a standard learning problem (*e.g.*, an SVM) is repeatedly solved; in large-scale structured prediction, it is often prohibitive to tackle this problem in its batch form, and one typically resorts to online methods (Bottou,

1991; Collins, 2002; Ratliff et al., 2006; Collins et al., 2008). These methods are fast in achieving low generalization error, but converge slowly to the training objective, thus are unattractive for repeated use in the inner loop.

In this paper, we overcome the above difficulty by proposing a stand-alone online MKL algorithm. The algorithm is based on the kernelization of the recent forward-backward splitting scheme FOBOS Duchi and Singer (2009) and iterates between subgradient and proximal steps. In passing, we improve the FOBOS regret bound and show how to efficiently compute the proximal projections associated with the *squared* ℓ_1-norm, despite the fact that the underlying optimization problem is not separable.

After reviewing structured prediction and MKL (§2), we present a wide class of online proximal algorithms (§3) which extend FOBOS by handling composite regularizers with multiple proximal steps. These algorithms have convergence guarantees and are applicable in MKL, group-LASSO (Yuan and Lin, 2006) and other structural sparsity formalisms, such as hierarchical LASSO/MKL Bach (2008b); Zhao et al. (2008), group-LASSO with overlapping groups Jenatton et al. (2009), sparse group-LASSO (Friedman et al., 2010), and the elastic net MKL (Tomioka and Suzuki, 2010). We apply our MKL algorithm to structured prediction (§4), using the two following testbeds: sequence labeling for handwritten text recognition, and natural language dependency parsing. We show the potential of our approach by learning combinations of kernels from tens of thousands of training instances, with encouraging results in terms of runtimes, accuracy and identifiability.

2 Structured Prediction, Group Sparsity, and Multiple Kernel Learning

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output sets, respectively. In structured prediction, to each input $x \in \mathcal{X}$ corresponds a (structured and exponentially large) set $\mathcal{Y}(x) \subseteq \mathcal{Y}$ of legal outputs; *e.g.*, in sequence labeling, each $x \in \mathcal{X}$ is an observed sequence and each $y \in \mathcal{Y}(x)$ is the corresponding sequence of labels; in parsing, each $x \in \mathcal{X}$ is a string, and each $y \in \mathcal{Y}(x)$ is a parse tree that spans that string.

Let $\mathcal{U} \triangleq \{(x, y) \mid x \in \mathcal{X}, y \in \mathcal{Y}(x)\}$ be the set of all legal input-output pairs. Given a labeled dataset $\mathcal{D} \triangleq \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathcal{U}$, we want to learn a predictor $h : \mathcal{X} \to \mathcal{Y}$ of the form

$$h(x) \triangleq \arg \max_{y \in \mathcal{Y}(x)} f(x, y), \tag{1}$$

where $f : \mathcal{U} \to \mathbb{R}$ is a compatibility function. Problem (1) is called *inference* (or *decoding*) and involves combinatorial optimization (*e.g.*, dynamic programming). In this paper, we use linear functions, $f(x, y) = \langle \boldsymbol{\theta}, \boldsymbol{\phi}(x, y) \rangle$, where $\boldsymbol{\theta}$ is a parameter vector and $\boldsymbol{\phi}(x, y)$ a feature vector. The structure of the output is usually taken care of by assuming a decomposition of the form $\boldsymbol{\phi}(x, y) = \sum_{r \in \mathcal{R}} \boldsymbol{\phi}_r(x, y_r)$, where $\mathcal{R}$ is a set of *parts* and the y_r are partial output assignments (see (Taskar et al., 2003) for details). Instead of explicit features, one may use a positive definite kernel, $K : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$, and let f belong to the induced RKHS $\mathcal{H}_K$. Given a convex loss function $L : \mathcal{H}_K \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, the *learning* problem is usually formulated as a minimization of the regularized empirical risk:

$$\min_{f \in \mathcal{H}_K} \frac{\lambda}{2} \|f\|^2_{\mathcal{H}_K} + \frac{1}{m} \sum_{i=1}^{m} L(f; x_i, y_i), \tag{2}$$

where $\lambda \geq 0$ is a regularization parameter and $\|.\|_{\mathcal{H}_K}$ is the norm in $\mathcal{H}_K$. In structured prediction, the

logistic loss (in CRFs) and the structured hinge loss (in SVMs) are common choices:

$$L_{\text{CRF}}(f;x,y) \triangleq \log \textstyle\sum_{y'\in\mathcal{Y}(x)} \exp(f(x,y') - f(x,y)), \tag{3}$$

$$L_{\text{SVM}}(f;x,y) \triangleq \max_{y'\in\mathcal{Y}(x)} f(x,y') - f(x,y) + \ell(y',y). \tag{4}$$

In (4), $\ell : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}_+$ is a user-given cost function. The solution of (2) can be expressed as a kernel expansion (structured version of the representer theorem (Hofmann et al., 2008, Corollary 13)).

In the kernel learning framework Bach et al. (2004); Lanckriet et al. (2004), the kernel is expressed as a convex combination of elements of a finite set $\{K_1,\ldots,K_p\}$, the coefficients of which are learned from data. That is, $K \in \mathcal{K}$, where

$$\mathcal{K} \triangleq \left\{ K = \textstyle\sum_{j=1}^p \beta_j K_j \;\middle|\; \boldsymbol{\beta} \in \Delta^p \right\}, \quad \text{with} \quad \Delta^p \triangleq \left\{ \boldsymbol{\beta} \in \mathbb{R}_+^p \mid \textstyle\sum_{j=1}^p \beta_j = 1 \right\}. \tag{5}$$

The so-called MKL problem is the minimization of (2) with respect to K. Letting $\mathcal{H}_{\mathcal{K}} = \bigoplus_{j=1}^p \mathcal{H}_{K_j}$ be the direct sum of the RKHS, this optimization can be written (as shown in (Bach et al., 2004; Rakotomamonjy et al., 2008)) as:

$$f^* = \arg\min_{f\in\mathcal{H}_{\mathcal{K}}} \quad \frac{\lambda}{2}\left(\sum_{j=1}^p \|f_j\|_{\mathcal{H}_{K_j}}\right)^2 + \frac{1}{m}\sum_{i=1}^m L\left(\sum_{j=1}^p f_j; x_i, y_i\right), \tag{6}$$

where the optimal kernel coefficients are $\beta_j^* = \|f_j^*\|_{\mathcal{H}_{K_j}} / \sum_{l=1}^p \|f_l^*\|_{\mathcal{H}_{K_l}}$. For explicit features, the parameter vector is split into p groups, $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_p)$, and the minimization in (6) becomes

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \frac{\lambda}{2}\|\boldsymbol{\theta}\|_{2,1}^2 + \frac{1}{m}\sum_{i=1}^m L(\boldsymbol{\theta}; x_i, y_i), \tag{7}$$

where $\|\boldsymbol{\theta}\|_{2,1} \triangleq \sum_{j=1}^p \|\boldsymbol{\theta}_j\|$ is a sum of ℓ_2-norms, called the *mixed* $\ell_{2,1}$*-norm*. The group-LASSO criterion (Yuan and Lin, 2006) is similar to (7), without the square in the regularization term, revealing a close relationship with MKL (Bach, 2008a). In fact, the two problems are equivalent up to a change of λ. The $\ell_{2,1}$-norm regularizer favors *group sparsity*: groups that are found irrelevant tend to be entirely discarded.

Early approaches to MKL (Lanckriet et al., 2004; Bach et al., 2004) considered the dual of (6) in a QCQP or SOCP form, thus were limited to small scale problems. Subsequent work focused on scalability: in (Sonnenburg et al., 2006), a semi-infinite LP formulation and a cutting plane algorithm are proposed; SimpleMKL (Rakotomamonjy et al., 2008) alternates between learning an SVM and a gradient-based (or Newton Chapelle and Rakotomamonjy (2008)) update of the kernel weights; other techniques include the extended level method (Xu et al., 2009) and SpicyMKL (Suzuki and Tomioka, 2009), based on an augmented Lagrangian method. These are all batch algorithms, requiring the repeated solution of problems of the form (2); even if one can take advantage of warm-starts, the convergence proofs of these methods, when available, rely on the exactness (or prescribed accuracy in the dual) of these solutions.

In contrast, we tackle (6) and (7) in *primal* form. Rather than repeatedly calling off-the-shelf solvers for (2), we propose a stand-alone online algorithm with runtime comparable to that of solving a *single* instance of (2) by online methods (the fastest in large-scale settings (Shalev-Shwartz et al., 2007; Bottou, 1991)). This paradigm shift paves the way for extending MKL to structured prediction, a large territory yet to be explored.

3 Online Proximal Algorithms

We frame our online MKL algorithm in a wider class of *online proximal algorithms*. The theory of proximity operators (Moreau, 1962), which is widely known in optimization and has recently gained prominence in the signal processing community (Combettes and Wajs, 2006; Wright et al., 2009), provides tools for analyzing these algorithms and generalizes many known results, sometimes with remarkable simplicity. We thus start by summarizing its important concepts in §3.1, together with a quick review of convex analysis.

3.1 Convex Functions, Subdifferentials, Proximity Operators, and Moreau Projections

Throughout, we let $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ (where $\bar{\mathbb{R}} \triangleq \mathbb{R} \cup \{+\infty\}$) be a convex, lower semicontinuous (lsc) (the epigraph $\mathrm{epi}\varphi \triangleq \{(x,t) \in \mathbb{R}^p \times \mathbb{R} \,|\, \varphi(x) \leq t\}$ is closed in $\mathbb{R}^p \times \mathbb{R}$), and proper ($\exists \mathbf{x} : \varphi(\mathbf{x}) \neq +\infty$) function. The *subdifferential* of φ at $\mathbf{x}_0$ is the set

$$\partial\varphi(\mathbf{x}_0) \triangleq \{\mathbf{g} \in \mathbb{R}^d \mid \forall \mathbf{x} \in \mathbb{R}^d,\ \varphi(\mathbf{x}) - \varphi(\mathbf{x}_0) \geq \mathbf{g}^\top(\mathbf{x} - \mathbf{x}_0)\},$$

the elements of which are the *subgradients*. We say that φ is *G-Lipschitz* in $\mathcal{S} \subseteq \mathbb{R}^d$ if $\forall \mathbf{x} \in \mathcal{S}, \forall \mathbf{g} \in \partial\varphi(\mathbf{x}), \|\mathbf{g}\| \leq G$. We say that φ is *σ-strongly convex* in $\mathcal{S}$ if

$$\forall \mathbf{x}_0 \in \mathcal{S}, \quad \forall \mathbf{g} \in \partial\varphi(\mathbf{x}_0), \quad \forall \mathbf{x} \in \mathbb{R}^d, \quad \varphi(\mathbf{x}) \geq \varphi(\mathbf{x}_0) + \mathbf{g}^\top(\mathbf{x} - \mathbf{x}_0) + (\sigma/2)\|\mathbf{x} - \mathbf{x}_0\|^2.$$

The *Fenchel conjugate* of φ is $\varphi^\star : \mathbb{R}^p \to \bar{\mathbb{R}}$, $\varphi^\star(\mathbf{y}) \triangleq \sup_{\mathbf{x}} \mathbf{y}^\top\mathbf{x} - \varphi(\mathbf{x})$. Let:

$$M_\varphi(\mathbf{y}) \triangleq \inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 + \varphi(\mathbf{x}), \quad \text{and} \quad \mathrm{prox}_\varphi(\mathbf{y}) = \arg\inf_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 + \varphi(\mathbf{x});$$

the function $M_\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ is called the *Moreau envelope* of φ, and the map $\mathrm{prox}_\varphi : \mathbb{R}^p \to \mathbb{R}^p$ is the *proximity operator* of φ (Combettes and Wajs, 2006; Moreau, 1962). Proximity operators generalize Euclidean projectors: consider the case $\varphi = \iota_{\mathcal{C}}$, where $\mathcal{C} \subseteq \mathbb{R}^p$ is a convex set and $\iota_{\mathcal{C}}$ denotes its indicator (*i.e.*, $\varphi(\mathbf{x}) = 0$ if $\mathbf{x} \in \mathcal{C}$ and $+\infty$ otherwise). Then, prox_φ is the Euclidean projector onto $\mathcal{C}$ and M_φ is the residual. Two other important examples of proximity operators follow:

- if $\varphi(\mathbf{x}) = (\lambda/2)\|\mathbf{x}\|^2$, then $\mathrm{prox}_\varphi(\mathbf{y}) = \mathbf{y}/(1+\lambda)$;
- if $\varphi(\mathbf{x}) = \tau\|\mathbf{x}\|_1$, then $\mathrm{prox}_\varphi(\mathbf{y}) = \mathrm{soft}(\mathbf{y}, \tau)$ is the *soft-threshold* function Wright et al. (2009), defined as $[\mathrm{soft}(\mathbf{y}, \tau)]_k = \mathrm{sgn}(y_k) \cdot \max\{0, |y_k| - \tau\}$.

If $\varphi : \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_p} \to \bar{\mathbb{R}}$ is (group-)separable, *i.e.*, $\varphi(\mathbf{x}) = \sum_{k=1}^p \varphi_k(\mathbf{x}_k)$, where $\mathbf{x}_k \in \mathbb{R}^{d_k}$, then its proximity operator inherits the same (group-)separability: $[\mathrm{prox}_\varphi(\mathbf{x})]_k = \mathrm{prox}_{\varphi_k}(\mathbf{x}_k)$ Wright et al. (2009). For example, the proximity operator of the mixed $\ell_{2,1}$-norm, which is group-separable, has this form. The following proposition, that we prove in Appendix A, extends this result by showing how to compute proximity operators of functions (maybe not separable) that only depend on the ℓ_2-norms of groups of components; *e.g.*, the proximity operator of the squared $\ell_{2,1}$-norm reduces to that of squared ℓ_1.

Proposition 1 *Let $\varphi : \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_p} \to \bar{\mathbb{R}}$ be of the form $\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p) = \psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|)$ for some $\psi : \mathbb{R}^p \to \bar{\mathbb{R}}$. Then, $M_\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p) = M_\psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|)$ and $[\mathrm{prox}_\varphi(\mathbf{x}_1, \ldots, \mathbf{x}_p)]_k = [\mathrm{prox}_\psi(\|\mathbf{x}_1\|, \ldots, \|\mathbf{x}_p\|)]_k(\mathbf{x}_k/\|\mathbf{x}_k\|)$.*

Finally, we recall the *Moreau decomposition*, relating the proximity operators of Fenchel conjugate functions (Combettes and Wajs, 2006) and present a corollary (proved in Appendix B) that is the key to our regret bound in §3.3.

Proposition 2 (Moreau (1962)) *For any convex, lsc, proper function* $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$,

$$\mathbf{x} = \operatorname{prox}_{\varphi}(\mathbf{x}) + \operatorname{prox}_{\varphi^\star}(\mathbf{x}) \quad \textit{and} \quad \|\mathbf{x}\|^2/2 = M_{\varphi}(\mathbf{x}) + M_{\varphi^\star}(\mathbf{x}). \tag{8}$$

Corollary 3 *Let* $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ *be as in Prop. 2, and* $\bar{\mathbf{x}} \triangleq \operatorname{prox}_{\varphi}(\mathbf{x})$. *Then, any* $\mathbf{y} \in \mathbb{R}^p$ *satisfies*

$$\|\mathbf{y} - \bar{\mathbf{x}}\|^2 - \|\mathbf{y} - \mathbf{x}\|^2 \leq 2(\varphi(\mathbf{y}) - \varphi(\bar{\mathbf{x}})). \tag{9}$$

Although the Fenchel dual $\varphi^\star$ does not show up in (9), it has a crucial role in proving Corollary 3.

3.2 A General Online Proximal Algorithm for Composite Regularizers

The general algorithmic structure that we propose and analyze in this paper, presented as Alg. 1, deals (in an online[1] fashion) with problems of the form

$$\min_{\boldsymbol{\theta} \in \Theta} \lambda R(\boldsymbol{\theta}) + \frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{\theta}; x_i, y_i), \tag{10}$$

where $\Theta \subseteq \mathbb{R}^d$ is convex[2] and the regularizer R has a composite form $R(\boldsymbol{\theta}) = \sum_{j=1}^{J} R_j(\boldsymbol{\theta})$. Like stochastic gradient descent (SGD (Bottou, 1991)), Alg. 1 is suitable for problems with large m; it also performs (sub-)gradient steps at each round (line 4), but only w.r.t. the loss function L. Obtaining a subgradient typically involves inference using the current model; *e.g.*, loss-augmented inference, if $L = L_{\text{SVM}}$, or marginal inference if $L = L_{\text{CRF}}$. Our algorithm differs from SGD by the inclusion of J proximal steps w.r.t. to each term R_j (line 7). As noted in (Duchi and Singer, 2009; Langford et al., 2009), this strategy is more effective than standard SGD for sparsity-inducing regularizers, due to their usual non-differentiability at the zeros, which causes oscillation and prevents SGD from returning sparse solutions.

When $J = 1$, Alg. 1 reduces to FOBOS (Duchi and Singer, 2009), which we kernelize and apply to MKL in §3.4. The case $J > 1$ has applications in variants of MKL or group-LASSO with composite regularizers (Tomioka and Suzuki, 2010; Friedman et al., 2010; Bach, 2008b; Zhao et al., 2008). In those cases, the proximity operators of $R_1, \ldots, R_J$ are more easily computed than that of their sum R, making Alg. 1 more suitable than FOBOS. We present a few particular instances (all with $\Theta = \mathbb{R}^d$).

Projected subgradient with groups. Let $J = 1$ and R be the indicator of a convex set $\Theta' \subseteq \mathbb{R}^d$. Then (see §3.1), each proximal step is the Euclidean projection onto Θ' and Alg. 1 becomes the online projected subgradient algorithm from (Zinkevich, 2003). Letting $\Theta' \triangleq \{\boldsymbol{\theta} \in \mathbb{R} \mid \|\boldsymbol{\theta}\|_{2,1} \leq \gamma\}$ yields an equivalent problem to group-LASSO and MKL (7). Using Prop. 1, each proximal step reduces to a projection onto a ℓ_1-ball whose dimension is the number of groups (see a fast algorithm in (Duchi et al., 2008)).

[1] For simplicity, we focus on the pure online setting, *i.e.*, each parameter update uses a single observation; analogous algorithms may be derived for the batch and mini-batch cases.

[2] We are particularly interested in the case where $\boldsymbol{\theta} \in \Theta$ is a "vacuous" constraint whose goal is to confine each iterate $\boldsymbol{\theta}_t$ to a region containing the optimum, by virtue of the projection step in line 9. The analysis in §3.3 will make this more clear. The same trick is used in PEGASOS (Shalev-Shwartz et al., 2007).

Algorithm 1 Online Proximal Algorithm

input: dataset $\mathcal{D}$, parameter λ, number of rounds T, learning rate sequence $(\eta_t)_{t=1,\dots,T}$
initialize $\boldsymbol{\theta}_1 = \mathbf{0}$; set $m = |\mathcal{D}|$
for $t = 1$ **to** T **do**
  take a training pair (x_t, y_t) and obtain a subgradient $\mathbf{g} \in \partial L(\boldsymbol{\theta}_t; x_t, y_t)$
  $\tilde{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_t - \eta_t \mathbf{g}$ (gradient step)
  for $j = 1$ **to** J **do**
    $\tilde{\boldsymbol{\theta}}_{t+j/J} = \mathrm{prox}_{\eta_t \lambda R_j}(\tilde{\boldsymbol{\theta}}_{t+(j-1)/J})$ (proximal step)
  end for
  $\boldsymbol{\theta}_{t+1} = \Pi_\Theta(\tilde{\boldsymbol{\theta}}_{t+1})$ (projection step)
end for
output: the last model $\boldsymbol{\theta}_{T+1}$ or the averaged model $\bar{\boldsymbol{\theta}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\theta}_t$

Algorithm 2 Moreau Projection for ℓ_1^2

input: vector $\mathbf{x} \in \mathbb{R}^d$ and parameter $\lambda > 0$
sort the entries of $|\mathbf{x}|$ into $\mathbf{y}$ (*i.e.*, such that $y_1 \geq \ldots \geq y_p$)
find $\rho = \max\left\{j \in \{1, \ldots, p\} \mid y_j - (\lambda/(1+j\lambda))\sum_{r=1}^{j} y_r > 0\right\}$
output: $\mathbf{z} = \mathrm{soft}(\mathbf{x}, \tau)$, where $\tau = (\lambda/(1+\rho\lambda))\sum_{r=1}^{\rho} y_r$

Truncated subgradient with groups. Let $J = 1$ and $R(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{2,1}$, so that (10) becomes the usual formulation of group-LASSO, for a general loss L. Then, Alg. 1 becomes a group version of truncated gradient descent (Langford et al., 2009), studied in (Duchi and Singer, 2009) for multi-task learning. Similar batch algorithms have also been proposed (Wright et al., 2009). The reduction from $\ell_{2,1}$ to ℓ_1 can again be made due to Prop. 1; and each proximal step becomes a simple soft thresholding operation (as shown in §3.1).

Proximal subgradient for *squared* mixed $\ell_{2,1}$. With $R(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_{2,1}^2$, we have the MKL problem (7). Prop. 1 allows reducing each proximal step w.r.t. the squared $\ell_{2,1}$ to one w.r.t. the squared ℓ_1; however, unlike in the previous example, squared ℓ_1 is not separable. This apparent difficulty has led some authors (*e.g.*, Suzuki and Tomioka (2009)) to remove the square from R, which yields the previous example. However, despite the non-separability of R, the proximal steps can still be efficiently computed: see Alg. 2. This algorithm requires sorting the weights of each group, which has $O(p \log p)$ cost; we show its correctness in Appendix F. Non-MKL applications of the squared $\ell_{2,1}$ norm are found in (Kowalski and Torrésani, 2009; Zhou et al., 2010).

Other variants of group-LASSO and MKL. In hierarchical LASSO and group-LASSO with overlaps (Bach, 2008b; Zhao et al., 2008; Jenatton et al., 2009), each feature may appear in more than one group. Alg. 1 handles these problems by enabling a proximal step for each group. Sparse group-LASSO (Friedman et al., 2010) simultaneously promotes group-sparsity and sparsity *within* each group, by using $R(\boldsymbol{\theta}) = \sigma\|\boldsymbol{\theta}\|_{2,1} + (1-\sigma)\|\boldsymbol{\theta}\|_1$; Alg. 1 can handle this regularizer by using two proximal steps, both involving simple soft-thresholding: one at the group level, and another within each group. In non-sparse MKL ((Kloft et al., 2010), §4.4), $R = \frac{1}{2}\sum_{k=1}^{p} \|\boldsymbol{\theta}_k\|^q$. Invoking Prop. 1 and separability, the resulting proximal step amounts to solving p scalar equations of the form

$x - x_0 + \lambda\,\eta_t\,q\,x^{q-1} = 0$, also valid for $q \geq 2$ (unlike the method described in (Kloft et al., 2010)).

3.3 Regret, Convergence, and Generalization Bounds

We next show that, for a convex loss L and under standard assumptions, Alg. 1 converges up to ϵ precision, with high confidence, in $O(1/\epsilon^2)$ iterations. If L or R are strongly convex, this bound is improved to $\tilde{O}(1/\epsilon)$, where $\tilde{O}$ hides logarithmic terms. Our proofs combine tools of online convex programming (Zinkevich, 2003; Hazan et al., 2007) and classical results about proximity operators (Moreau, 1962; Combettes and Wajs, 2006). The key is the following lemma (that we prove in Appendix C).

Lemma 4 *Assume that $\forall (x, y) \in \mathcal{U}$, the loss $L(\cdot; x, y)$ is convex and G-Lipschitz on Θ, and that the regularizer $R = R_1 + \ldots + R_J$ satisfies the following conditions:* **(i)** *each R_j is convex;* **(ii)** *$\forall \boldsymbol{\theta} \in \Theta$, $\forall j' < j$, $R_{j'}(\boldsymbol{\theta}) \geq R_{j'}(\mathrm{prox}_{\lambda R_j}(\boldsymbol{\theta}))$ (each proximity operator $\mathrm{prox}_{\lambda R_j}$ does not increase the previous $R_{j'}$);* **(iii)** *$R(\boldsymbol{\theta}) \geq R(\Pi_\Theta(\boldsymbol{\theta}))$ (projecting the argument onto Θ does not increase R). Then, for any $\bar{\boldsymbol{\theta}} \in \Theta$, at each round t of Alg. 1,*

$$L(\boldsymbol{\theta}_t) + \lambda R(\boldsymbol{\theta}_{t+1}) \leq L(\bar{\boldsymbol{\theta}}) + \lambda R(\bar{\boldsymbol{\theta}}) + \frac{\eta_t}{2} G^2 + \frac{\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t\|^2 - \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t+1}\|^2}{2\eta_t}. \tag{11}$$

If, in addition, L is σ-strongly convex, then the bound in (11) can be strengthened to

$$L(\boldsymbol{\theta}_t) + \lambda R(\boldsymbol{\theta}_{t+1}) \leq L(\bar{\boldsymbol{\theta}}) + \lambda R(\bar{\boldsymbol{\theta}}) + \frac{\eta_t}{2} G^2 + \frac{\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t\|^2 - \|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_{t+1}\|^2}{2\eta_t} - \frac{\sigma}{2}\|\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_t\|^2. \tag{12}$$

A related, but less tight, bound for $J = 1$ was derived in Duchi and Singer (2009); instead of our term $\frac{\eta}{2}G^2$ in (11), the bound of (Duchi and Singer, 2009) has $7\frac{\eta}{2}G^2$.[3] When $R = \|\cdot\|_1$, FOBOS becomes the truncated gradient algorithm of Langford et al. (2009) and our bound matches the one therein derived, closing the gap between (Duchi and Singer, 2009) and (Langford et al., 2009). The classical result in Prop. 2, relating Moreau projections and Fenchel duality, is the crux of our bound, via Corollary 3. Finally, note that the conditions **(i)–(iii)** are not restrictive: they hold whenever the proximity operators are shrinkage functions (*e.g.*, if $R_j = \|\boldsymbol{\theta}\|_{p_j}^{q_j}$, with $p_j, q_j \geq 1$).

We next characterize Alg. 1 in terms of its cumulative regret w.r.t. the best fixed hypothesis, *i.e.*,

$$\mathrm{Reg}_T \triangleq \sum_{t=1}^{T} \left(\lambda R(\boldsymbol{\theta}_t) + L(\boldsymbol{\theta}_t; x_t, y_t)\right) - \min_{\boldsymbol{\theta} \in \Theta} \sum_{t=1}^{T} \left(\lambda R(\boldsymbol{\theta}) + L(\boldsymbol{\theta}; x_t, y_t)\right). \tag{13}$$

Proposition 5 (regret bounds with fixed and decaying learning rates) *Assume the conditions of Lemma 4, along with $R \geq 0$ and $R(\mathbf{0}) = 0$. Then:*

1. *Running Alg. 1 with fixed learning rate η yields*

$$\mathrm{Reg}_T \leq \frac{\eta T}{2} G^2 + \frac{\|\boldsymbol{\theta}^*\|^2}{2\eta}, \quad \textit{where } \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Theta} \sum_{t=1}^{T} \left(\lambda R(\boldsymbol{\theta}) + L(\boldsymbol{\theta}; x_t, y_t)\right). \tag{14}$$

Setting $\eta = \|\boldsymbol{\theta}^*\|/(G\sqrt{T})$ yields a sublinear regret of $\|\boldsymbol{\theta}^*\| G \sqrt{T}$. (Note that this requires knowing in advance $\|\boldsymbol{\theta}^*\|$ and the number of rounds T.)

[3] This can be seen from their Eq. 9, setting $A = 0$ and $\eta_t = \eta_{t+\frac{1}{2}}$.

2. *Assume that Θ is bounded with diameter F (i.e., $\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| \leq F$). Let the learning rate be $\eta_t = \eta_0/\sqrt{t}$, with arbitrary $\eta_0 > 0$. Then,*

$$\mathrm{Reg}_T \quad \leq \quad \left(\frac{F^2}{2\eta_0} + G^2\eta_0\right)\sqrt{T}. \tag{15}$$

Optimizing the bound gives $\eta_0 = F/(\sqrt{2}G)$, yielding $\mathrm{Reg}_T \leq FG\sqrt{2T}$.

3. *If L is σ-strongly convex, and $\eta_t = 1/(\sigma t)$, we obtain a logarithmic regret bound:*

$$\mathrm{Reg}_T \leq G^2(1 + \log T)/(2\sigma). \tag{16}$$

Similarly to other analyses of online learning algorithms, once an online-to-batch conversion is specified, regret bounds allow us to obtain PAC bounds on optimization and generalization errors. The following proposition can be proved using the same techniques as in (Cesa-Bianchi et al., 2004; Shalev-Shwartz et al., 2007).

Proposition 6 (optimization and estimation error) *If the assumptions of Prop. 5 hold and $\eta_t = \eta_0/\sqrt{t}$ as in 2., then the version of Alg. 1 that returns the averaged model solves the optimization problem (10) with accuracy ϵ in $T = O((F^2G^2 + \log(1/\delta))/\epsilon^2)$ iterations, with probability at least $1 - \delta$. If L is also σ-strongly convex and $\eta_t = 1/(\sigma t)$ as in 3., then, for the version of Alg. 1 that returns $\boldsymbol{\theta}_{T+1}$, we get $T = \tilde{O}(G^2/(\sigma\delta\epsilon))$. The generalization bounds are of the same orders.*

We now pause to see how the analysis applies to some concrete cases. The requirement that the loss is G-Lipschitz holds for the hinge and logistic losses, where $G = 2\max_{u\in\mathcal{U}} \|\boldsymbol{\phi}(u)\|$ (see Appendix E). These losses are not strongly convex, and therefore Alg. 1 has only $O(1/\epsilon^2)$ convergence. If the *regularizer* R is σ-strongly convex, a possible workaround to obtain $\tilde{O}(1/\epsilon)$ convergence is to let L "absorb" that strong convexity by redefining $\tilde{L}(\boldsymbol{\theta}; x_t, y_t) = L(\boldsymbol{\theta}; x_t, y_t) + \sigma\|\boldsymbol{\theta}\|^2/2$. Since neither the $\ell_{2,1}$-norm nor its square are strongly convex, we cannot use this trick for the MKL case (7), but it *does* apply for non-sparse MKL (Kloft et al., 2010) ($\ell_{2,q}$-norms are strongly convex for $q > 1$) and for elastic net MKL (Suzuki and Tomioka, 2009). Still, the $O(1/\epsilon^2)$ rate for MKL is competitive with the best batch algorithms; *e.g.*, the method in Xu et al. (2009) achieves ϵ primal-dual gap in $O(1/\epsilon^2)$ iterations. Some losses of interest (*e.g.*, the squared loss, or the modified loss $\tilde{L}$ above) are G-Lipschitz in any compact subset of $\mathbb{R}^d$ but not in $\mathbb{R}^d$. However, if it is known in advance that the optimal solution must lie in some compact convex set Θ, we can add a vacuous constraint and run Alg. 1 with the projection step, making the analysis still applicable; we present concrete examples in Appendix E.

3.4 Online MKL

The instantiation of Alg. 1 for $R(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|^2_{2,1}$ yields Alg. 3. We consider $L = L_{\mathrm{SVM}}$; adapting to any generalized linear model (*e.g.*, $L = L_{\mathrm{CRF}}$) is straightforward. As discussed in the last paragraph of §3.3, it may be necessary to consider "vacuous" projection steps to ensure fast convergence. Hence, an optional upper bound γ on $\|\boldsymbol{\theta}\|$ is accepted as input. Suitable values of γ for the SVM and CRF case are given in Appendix E. In line 4, the scores of candidate outputs are computed groupwise; in structured prediction (see §2), a factorization over parts is assumed and the scores are for partial output assignments (see Taskar et al. (2003); Tsochantaridis et al. (2004) for details). The key novelty of Alg. 3 is in line 8, where the group structure is taken into account, by applying a proximity operator which corresponds to a groupwise shrinkage/thresholding, where some groups may be set to zero.

Algorithm 3 Online-MKL

1: **input:** $\mathcal{D}$, λ, T, radius γ, learning rate sequence $(\eta_t)_{t=1,\dots,T}$
2: initialize $\boldsymbol{\theta}^1 \leftarrow \mathbf{0}$
3: **for** $t = 1$ **to** T **do**
4: take an instance x_t, y_t and compute scores $f_k(x_t, y'_t) = \langle \boldsymbol{\theta}_k^t, \boldsymbol{\phi}_k(x_t, y'_t)\rangle$, for $k = 1, \dots, p$
5: decode: $\hat{y}_t \in \operatorname{argmax}_{y'_t \in \mathcal{Y}(x)} \sum_{k=1}^p f_k(x_t, y'_t) + \ell(y'_t, y_t)$
6: Gradient step: $\tilde{\boldsymbol{\theta}}_k^t = \boldsymbol{\theta}_k^t - \eta_t(\boldsymbol{\phi}_k(x_t, \hat{y}_t) - \boldsymbol{\phi}_k(x_t, y_t))$
7: compute weights $\tilde{b}_k^t = \|\tilde{\boldsymbol{\theta}}_k^t\|$, $k = 1, \dots, p$, and shrink them $\mathbf{b}^t = \operatorname{prox}_{\eta_t \lambda \|.\|_{2,1}^2}(\tilde{\mathbf{b}}^t)$ with Alg. 2
8: Proximal step: $\tilde{\boldsymbol{\theta}}_k^{t+1} = (b_k^t / \tilde{b}_k^t) \cdot \tilde{\boldsymbol{\theta}}_k^t$, for $k = 1, \dots, p$
9: Projection step: $\boldsymbol{\theta}^{t+1} = \tilde{\boldsymbol{\theta}}^{t+1} \cdot \min\{1, \gamma / \|\tilde{\boldsymbol{\theta}}^{t+1}\|\}$
10: **end for**
11: compute $\beta_k = \|\boldsymbol{\theta}_k^{T+1}\| / \sum_{l=1}^p \|\boldsymbol{\theta}_l^{T+1}\|$, for $k = 1, \dots, p$
12: return $\boldsymbol{\beta}$, and the last model $\boldsymbol{\theta}^{T+1}$

Although Alg. 3 is written in parametric form, it can be kernelized, as shown next (one can also use explicit features in some groups, and implicit in others). Observe that the parameters of the kth group after round t can be written as $\boldsymbol{\theta}_k^{t+1} = \sum_{s=1}^t \alpha_{ks}^{t+1}(\boldsymbol{\phi}_k(x_s, y_s) - \boldsymbol{\phi}_k(x_s, \hat{y}_s))$, where

$$\alpha_{ks}^{t+1} = \eta_s \prod_{r=s}^{t} \left((b_k^r / \tilde{b}_k^r) \min\{1, \gamma / \|\tilde{\boldsymbol{\theta}}^{r+1}\|\} \right) = \begin{cases} \eta_t (b_k^t / \tilde{b}_k^t) \min\{1, \gamma / \|\tilde{\boldsymbol{\theta}}^{t+1}\|\} & \text{if } s = t \\ \alpha_{ks}^t (b_k^t / \tilde{b}_k^t) \min\{1, \gamma / \|\tilde{\boldsymbol{\theta}}^{t+1}\|\} & \text{if } s < t. \end{cases}$$

Therefore, the inner products in line 4 can be kernelized. The cost of this step is $O(\min\{m, t\})$, instead of the $O(d_k)$ (where d_k is the dimension of the kth group) for the explicit feature case. After the decoding step (line 5), the supporting pair $(x_t, \hat{y}_t)$ is stored. Lines 7, 9 and 11 require the *norm* of each group, which can be manipulated using kernels: indeed, after each gradient step (line 6), we have (denoting $u_t = (x_t, y_t)$ and $\hat{u}_t = (x_t, \hat{y}_t)$):

$$\begin{aligned} \|\tilde{\boldsymbol{\theta}}_k^t\|^2 &= \|\boldsymbol{\theta}_k^t\|^2 - 2\eta_t \langle \boldsymbol{\theta}_k^t, \boldsymbol{\phi}_k(x_t, y_t)\rangle + \eta_t^2 \|\boldsymbol{\phi}_k(x_t, \hat{y}_t) - \boldsymbol{\phi}_k(x_t, y_t)\|^2 \\ &= \|\boldsymbol{\theta}_k^t\|^2 - 2\eta_t f_k(\hat{u}_t) + \eta_t^2 (K_k(u_t, u_t) + K_k(\hat{u}_t, \hat{u}_t) - 2K_k(u_t, \hat{u}_t)); \end{aligned} \tag{17}$$

and the proximal and projection steps merely scale these norms. When the algorithm terminates, it returns the kernel weights $\boldsymbol{\beta}$ and the sequence (α_{kt}^{T+1}).

In case of sparse explicit features, an implementation trick analogous to the one used in (Shalev-Shwartz et al., 2007) (where each $\boldsymbol{\theta}_k$ is represented by its norm and an unnormalized vector) can substantially reduce the amount of computation. In the case of implicit features with a sparse kernel matrix, a sparse storage of this matrix can also significantly speed up the algorithm, eliminating its dependency on m in line 4. Note also that all steps involving group-specific computation can be carried out in parallel using multiple machines, which makes Alg. 3 suitable for combining many kernels (large p).

4 Experiments

Handwriting recognition. We use the OCR dataset of Taskar et al. (2003) (`www.cis.upenn.edu/~taskar/ocr`), which has 6877 words written by 150 people (52152 characters). Each character is a

Kernel	Training Runtimes	Test Acc. (per char.)
Linear (L)	6 sec.	$72.8 \pm 4.4\%$
Quadratic (Q)	116 sec.	$85.5 \pm 0.3\%$
Gaussian (G) ($\sigma^2 = 5$)	123 sec.	$84.1 \pm 0.4\%$
Average $(L + Q + G)/3$	118 sec.	$84.3 \pm 0.3\%$
MKL $\beta_1 L + \beta_2 Q + \beta_3 G$	279 sec.	$87.5 \pm 0.4\%$
B_1-Spline (B_1)	8 sec.	$75.4 \pm 0.9\%$
Average $(L + B_1)/2$	15 sec.	$83.0 \pm 0.3\%$
MKL $\beta_1 L + \beta_2 B_1$	15 sec.	$85.2 \pm 0.3\%$

Table 1: Results for handwriting recognition. Averages over 10 runs on the same folds as in (Taskar et al., 2003), training on one and testing on the others. The linear and quadratic kernels are normalized to unit diagonal. In all cases, 20 epochs were used, with η_0 in (15) picked from $\{0.01, 0.1, 1, 10\}$ by selecting the one that most decreases the objective after 5 epochs. Results are for the best regularization coefficient $C = 1/(\lambda m)$ (chosen from $\{0.1, 1, 10, 10^2, 10^3, 10^4\}$).

16-by-8 binary image, *i.e.*, a 128-dimensional vector (our input) and has one of 26 labels (`a`–`z`; the outputs to predict). Like in (Taskar et al., 2003), we address this sequence labeling problem with a structured SVM; however, we *learn* the kernel from the data, via Alg. 3. We use an indicator basis function to represent the correlation between consecutive outputs. Our first experiment (reported in the upper part of Tab. 1) compares linear, quadratic, and Gaussian kernels, either used individually, combined via a simple average, or with MKL. The results show that MKL outperforms the others by 2% or more.

The second experiment aims at showing the ability of Alg. 3 to exploit both *feature* and *kernel* sparsity by learning a combination of a *linear* kernel (explicit features) with a *generalized* B_1*-spline* kernel, given by $K(\mathbf{x}, \mathbf{x}') = \max\{0, 1 - \|\mathbf{x} - \mathbf{x}'\|/h\}$, with h chosen so that the kernel matrix has $\sim 95\%$ zeros. The rationale is to combine the strength of a simple feature-based kernel with that of one depending only on a few nearest neighbors. The results (Tab. 1, bottom part) show that the MKL outperforms by $\sim 10\%$ the individual kernels, and by more than 2% the averaged kernel. Perhaps more importantly, the accuracy is not much worse than the best one obtained in the previous experiment, while the runtime is much faster (15 versus 279 seconds).

Dependency parsing. We trained non-projective dependency parsers for English, using the dataset from the CoNLL-2008 shared task Surdeanu et al. (2008) (39278 training sentences, $\sim 10^6$ tokens, and 2399 test sentences). The output to be predicted from each input sentence is the set of dependency arcs, linking *heads* to *modifiers*, that must define a spanning tree (see example in Fig. 1). We use arc-factored models, where the feature vectors decompose as $\boldsymbol{\phi}(x, y) = \sum_{(h,m) \in y} \boldsymbol{\phi}_{h,m}(x)$. Although they are not the state-of-the-art for this task, exact inference is tractable via minimum spanning tree algorithms (McDonald et al., 2005). We defined 507 feature templates for each candidate arc by conjoining the words, lemmas, and parts-of-speech of the head h and the modifier m, as well as the parts-of-speech of the surrounding words, and the distance and direction of attachment. This yields a large scale problem, with > 50 million features instantiated. The feature vectors associated with each candidate arc, however, are very sparse and this is exploited in the implementation. We ran Alg. 3 with explicit features, with each group standing for a feature template. MKL did not outperform a standard SVM in this experiment (90.67% against 90.92%); however, it showed a good performance

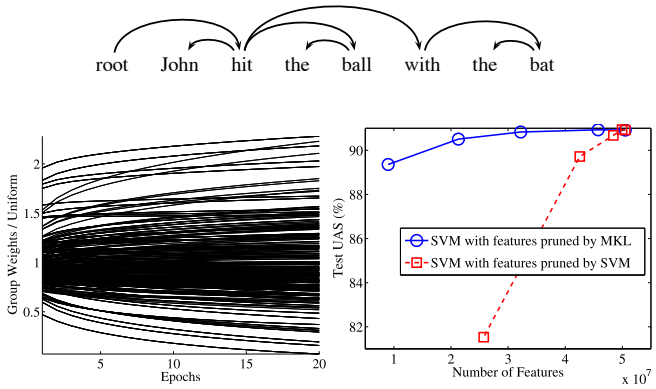

Figure 1: Top: a dependency parse tree (adapted from (McDonald et al., 2005)). Bottom left: group weights along the epochs of Alg. 3. Bottom right: results of standard SVMs trained on sets of feature templates of sizes $\{107, 207, 307, 407, 507\}$, either selected via a standard SVM or by MKL (the UAS—*unlabeled attachment score*—is the fraction of non-punctuation words whose head was correctly assigned.)

at pruning irrelevant feature templates (see Fig. 1, bottom right). Besides *interpretability*, which may be useful for the understanding of the syntax of natural languages, this pruning is also appealing in a two-stage architecture, where a standard learner at a second stage will only need to handle a small fraction of the templates initially hypothesized.

5 Conclusions

We introduced a new class of online proximal algorithms that extends FOBOS and is applicable to many variants of MKL and group-LASSO. We provided regret, convergence, and generalization bounds, and used the algorithm for learning the kernel in large-scale structured prediction tasks.

Our work may impact other problems. In structured prediction, the ability to promote structural sparsity suggests that it is possible to learn simultaneously the structure and the parameters of the graphical models. The ability to learn the kernel online offers a new paradigm for problems in which the underlying geometry (induced by the similarities between objects) evolves over time: algorithms that adapt the kernel while learning are robust to certain kinds of concept drift. We plan to explore these directions in future work.

A Proof of Proposition 1

We have respectively:

$$\begin{aligned}
M_\varphi(\mathbf{x}_1,\ldots,\mathbf{x}_p) &= \min_{\mathbf{y}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2 + \varphi(\mathbf{y}) \\
&= \min_{\mathbf{y}_1,\ldots,\mathbf{y}_p} \frac{1}{2}\sum_{k=1}^{p}\|\mathbf{y}_k-\mathbf{x}_k\|^2 + \psi(\|\mathbf{y}_1\|,\ldots,\|\mathbf{y}_p\|) \\
&= \min_{\mathbf{u}\in\mathbb{R}_+^p} \psi(u_1,\ldots,u_p) + \min_{\mathbf{y}:\|\mathbf{y}_k\|=u_k,\forall k} \frac{1}{2}\sum_{k=1}^{p}\|\mathbf{y}_k-\mathbf{x}_k\|^2 \\
&= \min_{\mathbf{u}\in\mathbb{R}_+^p} \psi(u_1,\ldots,u_p) + \frac{1}{2}\sum_{k=1}^{p}\min_{\mathbf{y}_k:\|\mathbf{y}_k\|=u_k} \|\mathbf{y}_k-\mathbf{x}_k\|^2 \quad (^*) \\
&= \min_{\mathbf{u}\in\mathbb{R}_+^p} \psi(u_1,\ldots,u_p) + \frac{1}{2}\sum_{k=1}^{p}\left\|\frac{u_k}{\|\mathbf{x}_k\|}\mathbf{x}_k-\mathbf{x}_k\right\|^2 \\
&= \min_{\mathbf{u}\in\mathbb{R}_+^p} \psi(u_1,\ldots,u_p) + \frac{1}{2}\sum_{k=1}^{p}(u_k-\|\mathbf{x}_k\|)^2 \\
&= M_\psi(\|\mathbf{x}_1\|,\ldots,\|\mathbf{x}_p\|), \qquad (18)
\end{aligned}$$

where the solution of the innermost minimization problem in $(^*)$ is $\mathbf{y}_k = \frac{u_k}{\|\mathbf{x}_k\|}\mathbf{x}_k$, and therefore $[\mathrm{prox}_\varphi(\mathbf{x}_1,\ldots,\mathbf{x}_p)]_k = [\mathrm{prox}_\psi(\|\mathbf{x}_1\|,\ldots,\|\mathbf{x}_p\|)]_k \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|}$.

B Proof of Corollary 3

We start by stating and proving the following lemma:

Lemma 7 *Let $\varphi : \mathbb{R}^p \to \bar{\mathbb{R}}$ be as in Prop. 2, and let $\bar{\mathbf{x}} \triangleq \mathrm{prox}_\varphi(\mathbf{x})$. Then, any $\mathbf{y} \in \mathbb{R}^p$ satisfies*

$$(\bar{\mathbf{x}}-\mathbf{y})^\top(\bar{\mathbf{x}}-\mathbf{x}) \le \varphi(\mathbf{y}) - \varphi(\bar{\mathbf{x}}) \qquad (19)$$

Proof: From (8), we have that

$$\begin{aligned}
\frac{1}{2}\|\mathbf{x}\|^2 &= \frac{1}{2}\|\bar{\mathbf{x}}-\mathbf{x}\|^2 + \varphi(\bar{\mathbf{x}}) + \frac{1}{2}\|\bar{\mathbf{x}}\|^2 + \varphi^*(\mathbf{x}-\bar{\mathbf{x}}) \\
&= \frac{1}{2}\|\bar{\mathbf{x}}-\mathbf{x}\|^2 + \varphi(\bar{\mathbf{x}}) + \frac{1}{2}\|\bar{\mathbf{x}}\|^2 + \sup_{\mathbf{u}\in\mathbb{R}^p}\Big(\mathbf{u}^\top(\mathbf{x}-\bar{\mathbf{x}}) - \varphi(\mathbf{u})\Big) \\
&\ge \frac{1}{2}\|\bar{\mathbf{x}}-\mathbf{x}\|^2 + \varphi(\bar{\mathbf{x}}) + \frac{1}{2}\|\bar{\mathbf{x}}\|^2 + \mathbf{y}^\top(\mathbf{x}-\bar{\mathbf{x}}) - \varphi(\mathbf{y}) \\
&= \frac{1}{2}\|\mathbf{x}\|^2 + \bar{\mathbf{x}}^\top(\bar{\mathbf{x}}-\mathbf{x}) + \mathbf{y}^\top(\mathbf{x}-\bar{\mathbf{x}}) - \varphi(\mathbf{y}) + \varphi(\bar{\mathbf{x}}) \\
&= \frac{1}{2}\|\mathbf{x}\|^2 + (\bar{\mathbf{x}}-\mathbf{y})^\top(\bar{\mathbf{x}}-\mathbf{x}) - \varphi(\mathbf{y}) + \varphi(\bar{\mathbf{x}}),
\end{aligned}$$

from which (19) follows. ∎

Now, take Lemma 7 and bound the left hand side as:

$$\begin{aligned}(\bar{\mathbf{x}}-\mathbf{y})^\top(\bar{\mathbf{x}}-\mathbf{x}) &\geq (\bar{\mathbf{x}}-\mathbf{y})^\top(\bar{\mathbf{x}}-\mathbf{x}) - \frac{1}{2}\|\bar{\mathbf{x}}-\mathbf{x}\|^2 \\ &= (\bar{\mathbf{x}}-\mathbf{y})^\top(\bar{\mathbf{x}}-\mathbf{x}) - \frac{1}{2}\|\bar{\mathbf{x}}\|^2 - \frac{1}{2}\|\mathbf{x}\|^2 + \bar{\mathbf{x}}^\top\mathbf{x} \\ &= \frac{1}{2}\|\bar{\mathbf{x}}\|^2 - \mathbf{y}^\top(\bar{\mathbf{x}}-\mathbf{x}) - \frac{1}{2}\|\mathbf{x}\|^2 \\ &= \frac{1}{2}\|\mathbf{y}-\bar{\mathbf{x}}\|^2 - \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|^2.\end{aligned}$$

This concludes the proof.

C Proof of Lemma 4

Let $u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}) \triangleq \lambda R(\bar{\boldsymbol{\theta}}) - \lambda R(\boldsymbol{\theta})$. We have successively:

$$\begin{aligned}\|\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_{t+1}\|^2 &\leq^{\text{(i)}} \|\bar{\boldsymbol{\theta}}-\tilde{\boldsymbol{\theta}}_{t+1}\|^2 \\ &\leq^{\text{(ii)}} \|\bar{\boldsymbol{\theta}}-\tilde{\boldsymbol{\theta}}_{t}\|^2 + 2\eta_t\lambda\sum_{j=1}^{J}(R_j(\bar{\boldsymbol{\theta}}) - R_j(\tilde{\boldsymbol{\theta}}_{t+j/J})) \\ &\leq^{\text{(iii)}} \|\bar{\boldsymbol{\theta}}-\tilde{\boldsymbol{\theta}}_{t}\|^2 + 2\eta_t u(\bar{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}_{t+1}) \\ &\leq^{\text{(iv)}} \|\bar{\boldsymbol{\theta}}-\tilde{\boldsymbol{\theta}}_{t}\|^2 + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\ &= \|\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_{t}\|^2 + \|\boldsymbol{\theta}_t-\tilde{\boldsymbol{\theta}}_{t}\|^2 + 2(\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_t)^\top(\boldsymbol{\theta}_t-\tilde{\boldsymbol{\theta}}_t) + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\ &= \|\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_{t}\|^2 + \eta_t^2\|\mathbf{g}\|^2 + 2\eta_t(\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_t)^\top\mathbf{g} + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\ &\leq^{\text{(v)}} \|\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_{t}\|^2 + \eta_t^2\|\mathbf{g}\|^2 + 2\eta_t(L(\bar{\boldsymbol{\theta}}) - L(\boldsymbol{\theta}_t)) + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}) \\ &\leq \|\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_{t}\|^2 + \eta_t^2 G^2 + 2\eta_t(L(\bar{\boldsymbol{\theta}}) - L(\boldsymbol{\theta}_t)) + 2\eta_t u(\bar{\boldsymbol{\theta}}, \boldsymbol{\theta}_{t+1}), \qquad (20)\end{aligned}$$

where the inequality (i) is due to the nonexpansiveness of the projection operator, (ii) follows from applying Corollary 3 J times, (iii) follows from applying the inequality $R_j(\tilde{\boldsymbol{\theta}}_{t+l/J}) \geq R_j(\tilde{\boldsymbol{\theta}}_{t+(l+1)/J})$ for $l = j, \ldots, J-1$, (iv) results from the fact that $R(\tilde{\boldsymbol{\theta}}_{t+1}) \geq R(\Pi_\Theta(\tilde{\boldsymbol{\theta}}_{t+1}))$, and (v) results from the subgradient inequality of convex functions, which has an extra term $\frac{\sigma}{2}\|\bar{\boldsymbol{\theta}}-\boldsymbol{\theta}_t\|^2$ if L is σ-strongly convex.

D Proof of Proposition 5

Invoke Lemma 4 and sum for $t = 1, \ldots, T$, which gives

$$
\begin{aligned}
& \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda R(\boldsymbol{\theta}_t)\right) \\
= & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda R(\boldsymbol{\theta}_{t+1})\right) - \lambda(R(\boldsymbol{\theta}_{m+1}) - R(\boldsymbol{\theta}_1)) \\
\leq^{\text{(i)}} & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda R(\boldsymbol{\theta}_{t+1})\right) \\
\leq & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)\right) + \frac{G^2}{2} \sum_{t=1}^{T} \eta_t + \sum_{t=1}^{T} \frac{\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_t\|^2 - \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{t+1}\|^2}{2\eta_t} \\
= & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)\right) + \frac{G^2}{2} \sum_{t=1}^{T} \eta_t + \frac{1}{2} \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) \cdot \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_t\|^2 \\
& + \frac{1}{2\eta_1} \cdot \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_1\|^2 - \frac{1}{2\eta_T} \cdot \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{T+1}\|^2
\end{aligned} \tag{21}
$$

where the inequality (i) is due to the fact that $\boldsymbol{\theta}_1 = \mathbf{0}$. Noting that the third term vanishes for a constant learning rate and that the last term is non-positive suffices to prove the first part. For the second part, we continue as:

$$
\begin{aligned}
& \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda R(\boldsymbol{\theta}_t)\right) \\
\leq & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)\right) + \frac{G^2}{2} \sum_{t=1}^{T} \eta_t + \frac{F^2}{2} \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right) + \frac{F^2}{2\eta_1} \\
= & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)\right) + \frac{G^2}{2} \sum_{t=1}^{T} \eta_t + \frac{F^2}{2\eta_T} \\
\leq^{\text{(ii)}} & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)\right) + G^2 \eta_0 (\sqrt{T} - 1/2) + \frac{F^2 \sqrt{T}}{2\eta_0} \\
\leq & \sum_{t=1}^{T} \left(L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)\right) + \left(G^2 \eta_0 + \frac{F^2}{2\eta_0}\right) \sqrt{T},
\end{aligned} \tag{22}
$$

where equality (ii) is due to the fact that $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$. For the third part, continue after inequality (i) as:

$$\begin{aligned}
& \sum_{t=1}^{T} (L(\boldsymbol{\theta}_t; x_t, y_t) + \lambda R(\boldsymbol{\theta}_t)) \\
\leq \quad & \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)) + \frac{G^2}{2} \sum_{t=1}^{T} \eta_t + \frac{1}{2} \sum_{t=2}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \sigma \right) \cdot \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_t\|^2 \\
& + \frac{1}{2} \left(\frac{1}{\eta_1} - \sigma \right) \cdot \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_1\|^2 - \frac{1}{2\eta_T} \cdot \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{T+1}\|^2 \\
= \quad & \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)) + \frac{G^2}{2\sigma} \sum_{t=1}^{T} \frac{1}{t} - \frac{\sigma T}{2} \cdot \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{T+1}\|^2 \\
\leq \quad & \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)) + \frac{G^2}{2\sigma} \sum_{t=1}^{T} \frac{1}{t} \\
\leq^{\text{(iii)}} \quad & \sum_{t=1}^{T} (L(\boldsymbol{\theta}^*; x_t, y_t) + \lambda R(\boldsymbol{\theta}^*)) + \frac{G^2}{2\sigma} (1 + \log T), \qquad (23)
\end{aligned}$$

where the equality (iii) is due to the fact that $\sum_{t=1}^{T} \frac{1}{t} \leq 1 + \log T$.

E Lipschitz Constants of Some Loss Functions

Let $\boldsymbol{\theta}^*$ be a solution of the problem (10) with $\Theta = \mathbb{R}^d$. For certain loss functions, we may obtain bounds of the form $\|\boldsymbol{\theta}^*\| \leq \gamma$ for some $\gamma > 0$, as the next proposition illustrates. Therefore, we may redefine $\Theta = \{\boldsymbol{\theta} \in \mathbb{R}^d \mid \|\boldsymbol{\theta}\| \leq \gamma\}$ (a vacuous constraint) without affecting the solution of (10).

Proposition 8 *Let $R(\boldsymbol{\theta}) = \frac{1}{2}(\sum_{k=1}^{p} \|\boldsymbol{\theta}_k\|)^2$. Let L_{SVM} and L_{CRF} be the structured hinge and logistic losses (4). Assume that the average cost function (in the SVM case) or the average entropy (in the CRF case) are bounded by some $\Lambda \geq 0$, i.e.,*[4]

$$\frac{1}{m} \sum_{i=1}^{m} \max_{y_i' \in \in \mathcal{Y}(x_t)} \ell(y_i'; y_i) \leq \Lambda \quad \textit{or} \quad \frac{1}{m} \sum_{i=1}^{m} H(Y_i) \leq \Lambda. \qquad (24)$$

Then:

1. *The solution of (10) with $\Theta = \mathbb{R}^d$ satisfies $\|\boldsymbol{\theta}^*\| \leq \sqrt{2\Lambda/\lambda}$.*
2. *L is G-Lipschitz on $\mathbb{R}^d$, with $G = 2 \max_{u \in \mathcal{U}} \|\boldsymbol{\phi}(u)\|$.*
3. *Consider the following problem obtained from (10) by adding a quadratic term:*

$$\min_{\boldsymbol{\theta}} \frac{\sigma}{2} \|\boldsymbol{\theta}\|^2 + \lambda R(\boldsymbol{\theta}) + \frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{\theta}; x_i, y_i). \qquad (25)$$

The solution of this problem satisfies $\|\boldsymbol{\theta}^*\| \leq \sqrt{2\Lambda/(\lambda + \sigma)}$.

[4] In sequence binary labeling, we have $\Lambda = \bar{N}$ for the CRF case and for the SVM case with a Hamming cost function, where $\bar{N}$ is the average sequence length. Observe that the entropy of a distribution over labelings of a sequence of length N is upper bounded by $\log 2^N = N$.

Algorithm 4 Moreau projection for the squared weighted ℓ_1-norm

Input: A vector $\mathbf{x}_0 \in \mathbb{R}^p$, a weight vector $\mathbf{d} \geq 0$, and a parameter $\lambda > 0$

Set $u_{0r} = |x_{0r}|/d_r$ and $a_r = d_r^2$ for each $r = 1, \ldots, p$

Sort $\mathbf{u}_0$: $u_{0(1)} \geq \ldots \geq u_{0(p)}$

Find $\rho = \max \left\{ j \in \{1, \ldots, p\} \mid u_{0(j)} - \frac{\lambda}{1+\lambda \sum_{r=1}^{j} a_{(r)}} \sum_{r=1}^{j} a_{(r)} u_{0(r)} > 0 \right\}$

Compute $\mathbf{u} = \text{soft}(\mathbf{u}_0, \tau)$, where $\tau = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}$

Output: $\mathbf{x}$ s.t. $x_r = \text{sign}(x_{0r}) d_r u_r$.

4. *The modified loss* $\tilde{L} = L + \frac{\sigma}{2}\|.\|^2$ *is* $\tilde{G}$*-Lipschitz on* $\left\{ \boldsymbol{\theta} \mid \|\boldsymbol{\theta}\| \leq \sqrt{2\Lambda/(\lambda+\sigma)} \right\}$*, where* $\tilde{G} = G + \sqrt{2\sigma^2\Lambda/(\lambda+\sigma)}$.

Proof: Let $F_{\text{SVM}}(\boldsymbol{\theta})$ and $F_{\text{CRF}}(\boldsymbol{\theta})$ be the objectives of (10) for the SVM and CRF cases. We have

$$F_{\text{SVM}}(\mathbf{0}) = \lambda R(\mathbf{0}) + \frac{1}{m}\sum_{i=1}^{m} L_{\text{SVM}}(\mathbf{0}; x_i, y_i) = \frac{1}{m}\sum_{i=1}^{m} \max_{y_i' \in \mathcal{Y}(x_i)} \ell(y_i'; y_i) \leq \Lambda_{\text{SVM}} \tag{26}$$

$$F_{\text{CRF}}(\mathbf{0}) = \lambda R(\mathbf{0}) + \frac{1}{m}\sum_{i=1}^{m} L_{\text{CRF}}(\mathbf{0}; x_i, y_i) = \frac{1}{m}\sum_{i=1}^{m} \log |\mathcal{Y}(x_i)| \leq \Lambda_{\text{CRF}} \tag{27}$$

Using the facts that $F(\boldsymbol{\theta}^*) \leq F(\mathbf{0})$, that the losses are non-negative, and that $(\sum_i |x_i|)^2 \geq \sum_i x_i^2$, we obtain $\frac{\lambda}{2}\|\boldsymbol{\theta}^*\|^2 \leq \lambda R(\boldsymbol{\theta}^*) \leq F(\boldsymbol{\theta}^*) \leq F(\mathbf{0})$, which proves the first statement.

To prove the second statement for the SVM case, note that a subgradient of L_{SVM} at $\boldsymbol{\theta}$ is $\mathbf{g}_{\text{SVM}} = \boldsymbol{\phi}(x, \hat{y}) - \boldsymbol{\phi}(x, y)$, where $\hat{y} = \arg\max_{y' \in \mathcal{Y}(x)} \boldsymbol{\theta}^\top(\boldsymbol{\phi}(x, y') - \boldsymbol{\phi}(x, y)) + \ell(y'; y)$; and that the gradient of L_{CRF} at $\boldsymbol{\theta}$ is $\mathbf{g}_{\text{CRF}} = \mathbb{E}_{\boldsymbol{\theta}}\boldsymbol{\phi}(x, Y) - \boldsymbol{\phi}(x, y)$. Applying Jensen's inequality, we have that $\|\mathbf{g}_{\text{CRF}}\| \leq \mathbb{E}_{\boldsymbol{\theta}}\|\boldsymbol{\phi}(x, Y) - \boldsymbol{\phi}(x, y)\|$. Therefore, both $\|\mathbf{g}_{\text{SVM}}\|$ and $\|\mathbf{g}_{\text{CRF}}\|$ are upper bounded by $\max_{x \in \mathcal{X}, y, y' \in \mathcal{Y}(x)} \|\boldsymbol{\phi}(x, y') - \boldsymbol{\phi}(x, y)\| \leq 2 \max_{u \in \mathcal{U}} \|\boldsymbol{\phi}(u)\|$.

The same rationale can be used to prove the third and fourth statements. ■

F Computing the proximity operator of the (non-separable) squared ℓ_1

We present an algorithm (Alg. 4) that computes the Moreau projection of the *squared*, *weighted* ℓ_1-norm. Denote by $\odot$ the Hadamard product, $[\mathbf{a} \odot \mathbf{b}]_k = a_k b_k$. Letting $\lambda, \mathbf{d} \geq 0$, and $\phi_{\mathbf{d}}(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{d} \odot \mathbf{x}\|_1^2$, the underlying optimization problem is:

$$M_{\lambda\phi_{\mathbf{d}}}(\mathbf{x}_0) \triangleq \min_{\mathbf{x} \in \mathbb{R}^p} \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|^2 + \frac{\lambda}{2}\left(\sum_{i=1}^{p} d_i |x_i|\right)^2. \tag{28}$$

This includes the squared ℓ_1-norm as a particular case, when $\mathbf{d} = \mathbf{1}$ (the case addressed in Alg. 2). The proof is somewhat technical and follows the same procedure employed by Duchi et al. (2008) to derive an algorithm for projecting onto the ℓ_1-ball. The runtime is $O(p \log p)$ (the amount of time that is necessary to sort the vector), but a similar trick as the one described in (Duchi et al., 2008) can be employed to yield $O(p)$ runtime.

Lemma 9 *Let* $\mathbf{x}^* = \text{prox}_{\lambda\phi_{\mathbf{d}}}(\mathbf{x}_0)$ *be the solution of* (28)*. Then:*

1. $\mathbf{x}^*$ *agrees in sign with* $\mathbf{x}_0$, i.e., *each component satisfies* $x_{0i} \cdot x_i^* \geq 0$.

2. *Let* $\boldsymbol{\sigma} \in \{-1, 1\}^p$. *Then* $\operatorname{prox}_{\lambda\phi_{\mathbf{d}}}(\boldsymbol{\sigma} \odot \mathbf{x}_0) = \boldsymbol{\sigma} \odot \operatorname{prox}_{\lambda\phi_{\mathbf{d}}}(\mathbf{x}_0)$, i.e., *flipping a sign in* $\mathbf{x}_0$ *produces a* $\mathbf{x}^*$ *with the same sign flipped.*

Proof: Suppose that $x_{0i} \cdot x_i^* < 0$ for some i. Then, $\mathbf{x}$ defined by $x_j = x_j^*$ for $j \neq i$ and $x_i = -x_i^*$ achieves a lower objective value than $\mathbf{x}^*$, since $\phi_{\mathbf{d}}(\mathbf{x}) = \phi_{\mathbf{d}}(\mathbf{x}^*)$ and $(x_i - x_{0i})^2 < (x_i^* - x_{0i})^2$; this contradicts the optimality of $\mathbf{x}^*$. The second statement is a simple consequence of the first one and that $\phi_{\mathbf{d},\lambda}(\boldsymbol{\sigma} \odot \mathbf{x}) = \phi_{\mathbf{d},\lambda}(\boldsymbol{\sigma} \odot \mathbf{x}^*)$. ■

Lemma 9 enables reducing the problem to the non-negative orthant, by writing $\mathbf{x}_0 = \boldsymbol{\sigma} \cdot \tilde{\mathbf{x}}_0$, with $\tilde{\mathbf{x}}_0 \geq \mathbf{0}$, obtaining a solution $\tilde{\mathbf{x}}^*$ and then recovering the true solution as $\mathbf{x}^* = \boldsymbol{\sigma} \cdot \tilde{\mathbf{x}}^*$. It therefore suffices to solve (28) with the constraint $\mathbf{x} \geq \mathbf{0}$, which in turn can be transformed into:

$$\min_{\mathbf{u} \geq \mathbf{0}} F(\mathbf{u}) \triangleq \frac{1}{2} \sum_{r=1}^{p} a_r (u_r - u_{0r})^2 + \frac{\lambda}{2} \left(\sum_{r=1}^{p} a_r u_r \right)^2, \tag{29}$$

where we made the change of variables $a_i \triangleq d_i^2$, $u_{0i} \triangleq x_{0i}/d_i$ and $u_i \triangleq x_i/d_i$.

The Lagrangian of (29) is $\mathcal{L}(\mathbf{u}, \boldsymbol{\xi}) = \frac{1}{2} \sum_{r=1}^{p} a_r (u_r - u_{0r})^2 + \frac{\lambda}{2} \left(\sum_{r=1}^{p} a_r u_r \right)^2 - \boldsymbol{\xi}^\top \mathbf{u}$, where $\boldsymbol{\xi} \geq \mathbf{0}$ are Lagrange multipliers. Equating the gradient (w.r.t. $\mathbf{u}$) to zero gives

$$\mathbf{a} \odot (\mathbf{u} - \mathbf{u}_0) + \lambda \sum_{r=1}^{p} a_r u_r \mathbf{a} - \boldsymbol{\xi} = \mathbf{0}. \tag{30}$$

From the complementary slackness condition, $u_j > 0$ implies $\xi_j = 0$, which in turn implies

$$a_j (u_j - u_{0j}) + \lambda a_j \sum_{r=1}^{p} a_r u_r = 0. \tag{31}$$

Thus, if $u_j > 0$, the solution is of the form $u_j = u_{0j} - \tau$, with $\tau = \lambda \sum_{r=1}^{p} a_r u_r$. The next lemma shows the existence of a split point below which some coordinates vanish.

Lemma 10 *Let* $\mathbf{u}^*$ *be the solution of* (29). *If* $u_k^* = 0$ *and* $u_{0j} < u_{0k}$, *then we must have* $u_j^* = 0$.

Proof: Suppose that $u_j^* = \epsilon > 0$. We will construct a $\tilde{\mathbf{u}}$ whose objective value is lower than $F(\mathbf{u}^*)$, which contradicts the optimality of $\mathbf{u}^*$: set $\tilde{u}_l = u_l^*$ for $l \notin \{j, k\}$, $\tilde{u}_k = \epsilon c$, and $\tilde{u}_j = \epsilon (1 - c a_k/a_j)$, where $c = \min\{a_j/a_k, 1\}$. We have $\sum_{r=1}^{p} a_r u_r^* = \sum_{r=1}^{p} a_r \tilde{u}_r$, and therefore

$$\begin{aligned} 2(F(\tilde{\mathbf{u}}) - F(\mathbf{u}^*)) &= \sum_{r=1}^{p} a_r (\tilde{u}_r - u_{0r})^2 - \sum_{r=1}^{p} a_r (u_r^* - u_{0r})^2 \\ &= a_j (\tilde{u}_j - u_{0j})^2 - a_j (u_j^* - u_{0j})^2 + a_k (\tilde{u}_k - u_{0k})^2 - a_k (u_k^* - u_{0k})^2. \end{aligned} \tag{32}$$

Consider the following two cases: (i) if $a_j \leq a_k$, then $\tilde{u}_k = \epsilon a_j/a_k$ and $\tilde{u}_j = 0$. Substituting in (32), we obtain $2(F(\tilde{\mathbf{u}}) - F(\mathbf{u}^*)) = \epsilon^2 \left(a_j^2/a_k - a_j \right) \leq 0$, which leads to the contradiction $F(\tilde{\mathbf{u}}) \leq F(\mathbf{u}^*)$. If (ii) $a_j > a_k$, then $\tilde{u}_k = \epsilon$ and $\tilde{u}_j = \epsilon (1 - a_k/a_j)$. Substituting in (32), we obtain $2(F(\tilde{\mathbf{u}}) - F(\mathbf{u}^*)) = a_j \epsilon^2 (1 - a_k/a_j)^2 + 2a_k \epsilon u_{0j} - 2a_k \epsilon u_{0k} + a_k \epsilon^2 - a_j \epsilon^2 < a_k^2/a_j \epsilon^2 - 2a_k \epsilon^2 + a_k \epsilon^2 = \epsilon^2 \left(a_k^2/a_j - a_k \right) < 0$, which also leads to a contradiction. ■

Let $u_{0(1)} \geq \ldots \geq u_{0(p)}$ be the entries of $\mathbf{u}_0$ sorted in decreasing order, and let $u^*_{(1)}, \ldots, u^*_{(p)}$ be the entries of $\mathbf{u}^*$ under the same permutation. Let ρ be the number of nonzero entries in $\mathbf{u}^*$, i.e., $u^*_{(\rho)} > 0$, and, if $\rho < p$, $u^*_{(\rho+1)} = 0$. Summing (31) for $(j) = 1, \ldots, \rho$, we get

$$\sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} - \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} + \left(\sum_{r=1}^{\rho} a_{(r)} \right) \lambda \sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} = 0, \tag{33}$$

which implies

$$\sum_{r=1}^{p} u^*_r = \sum_{r=1}^{\rho} u^*_{(r)} = \frac{1}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}, \tag{34}$$

and therefore $\tau = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}$. The complementary slackness conditions for $r = \rho$ and $r = \rho + 1$ imply

$$u^*_{(\rho)} - u_{0(\rho)} + \lambda \sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} = 0 \quad \text{and} \quad - u^*_{0(\rho+1)} + \lambda \sum_{r=1}^{\rho} a_{(r)} u^*_{(r)} = \xi_{(\rho+1)} \geq 0; \tag{35}$$

therefore $u_{0(\rho)} > u_{0(\rho)} - u^*_{(\rho)} = \tau \geq u_{0(\rho+1)}$. This implies that ρ is such that

$$u_{0(\rho)} > \frac{\lambda}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} \geq u_{0(\rho+1)}. \tag{36}$$

The next proposition goes farther by exactly determining ρ.

Proposition 11 *The quantity ρ can be determined via:*

$$\rho = \max \left\{ j \in [p] \;\middle|\; u_{0(j)} - \frac{\lambda}{1 + \lambda \sum_{r=1}^{j} a_{(r)}} \sum_{r=1}^{j} a_{(r)} u_{0(r)} > 0 \right\}. \tag{37}$$

Proof: Let $\rho^* = \max\{j | u^*_{(j)} > 0\}$. We have that $u^*_{(r)} = u_{0(r)} - \tau^*$ for $r \leq \rho^*$, where $\tau^* = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho^*} a_{(r)}} \sum_{r=1}^{\rho^*} a_{(r)} u_{0(r)}$, and therefore $\rho \geq \rho^*$. We need to prove that $\rho \leq \rho^*$, which we will do by contradiction. Assume that $\rho > \rho^*$. Let $\mathbf{u}$ be the vector induced by the choice of ρ, i.e., $u_{(r)} = 0$ for $r > \rho$ and $u_{(r)} = u_{0(r)} - \tau$ for $r \leq \rho$, where $\tau = \frac{\lambda}{1+\lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)}$. From the definition of ρ, we have $u_{(\rho)} = u_{0(\rho)} - \tau > 0$, which implies $u_{(r)} = u_{0(r)} - \tau > 0$ for each $r \leq \rho$. In addition,

$$\begin{aligned} \sum_{r=1}^{p} a_r u_r &= \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} - \sum_{r=1}^{\rho} a_{(r)} \tau = \left(1 - \frac{\lambda \sum_{r=1}^{\rho} a_{(r)}}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}} \right) \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} \\ &= \frac{1}{1 + \lambda \sum_{r=1}^{\rho} a_{(r)}} \sum_{r=1}^{\rho} a_{(r)} u_{0(r)} = \frac{\tau}{\lambda}, \end{aligned} \tag{38}$$

$$\begin{aligned} \sum_{r=1}^{p} a_r (u_r - u_{0r})^2 &= \sum_{r=1}^{\rho^*} a_{(r)} \tau^2 + \sum_{r=\rho^*+1}^{\rho} a_{(r)} \tau^2 + \sum_{r=\rho+1}^{p} a_{(r)} u^2_{0(r)} \\ &< \sum_{r=1}^{\rho^*} a_{(r)} \tau^2 + \sum_{r=\rho^*+1}^{p} a_{(r)} u^2_{0(r)}. \end{aligned} \tag{39}$$

We next consider two cases:

$\boxed{\tau^* \geq \tau.}$ From (39), we have that $\sum_{r=1}^{p} a_r(u_r - u_{0r})^2 < \sum_{r=1}^{\rho^*} a_{(r)}\tau^2 + \sum_{r=\rho^*+1}^{p} a_{(r)}u_{0(r)}^2 \leq \sum_{r=1}^{\rho^*} a_{(r)}(\tau^*)^2 + \sum_{r=\rho^*+1}^{p} a_{(r)}u_{0(r)}^2 = \sum_{r=1}^{p} a_r(u_r^* - u_{0r})^2$. From (38), we have that $\left(\sum_{r=1}^{p} a_r u_r\right)^2 = \tau^2/\lambda^2 \leq (\tau^*)^2/\lambda^2$. Summing the two inequalities, we get $F(\mathbf{u}) < F(\mathbf{u}^*)$, which leads to a contradiction.

$\boxed{\tau^* < \tau.}$ We will construct a vector $\tilde{\mathbf{u}}$ from $\mathbf{u}^*$ and show that $F(\tilde{\mathbf{u}}) < F(\mathbf{u}^*)$. Define

$$\tilde{u}_{(r)} = \begin{cases} u^*_{(\rho^*)} - \frac{2a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\epsilon, & \text{if } r = \rho^* \\ \frac{2a_{(\rho^*)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\epsilon, & \text{if } r = \rho^*+1 \\ u^*_{(r)} & \text{otherwise,} \end{cases} \tag{40}$$

where $\epsilon = (u_{0(\rho^*+1)} - \tau^*)/2$. Note that $\sum_{r=1}^{p} a_r\tilde{u}_r = \sum_{r=1}^{p} a_r u_r^*$. From the assumptions that $\tau^* < \tau$ and $\rho^* < \rho$, we have that $u^*_{(\rho^*+1)} = u_{0(\rho^*+1)} - \tau > 0$, which implies that $\tilde{u}_{(\rho^*+1)} = \frac{a_{(\rho^*)}(u_{0(\rho^*+1)}-\tau^*)}{a_{(\rho^*)}+a_{(\rho^*+1)}} > \frac{a_{(\rho^*)}(u_{0(\rho^*+1)}-\tau)}{a_{(\rho^*)}+a_{(\rho^*+1)}} = \frac{a_{(\rho^*)}u^*_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}} > 0$, and that $u^*_{(\rho^*)} = u_{0(\rho^*)} - \tau^* - \frac{a_{(\rho^*+1)}(u_{0(\rho^*+1)}-\tau^*)}{a_{(\rho^*)}+a_{(\rho^*+1)}} = u_{0(\rho^*)} - \frac{a_{(\rho^*+1)}u_{0(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}} - \left(1 - \frac{a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)\tau^* >^{\text{(i)}} \left(1 - \frac{a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)(u_{0(\rho^*+1)} - \tau) = \left(1 - \frac{a_{(\rho^*+1)}}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)(u^*_{(\rho^*+1)}) > 0$, where inequality (i) is justified by the facts that $u_{0(\rho^*)} \geq u_{0(\rho^*+1)}$ and $\tau > \tau^*$. This ensures that $\tilde{\mathbf{u}}$ is well defined. We have:

$$\begin{aligned} 2(F(\mathbf{u}^*) - F(\tilde{\mathbf{u}})) &= \sum_{r=1}^{p} a_r(u_r^* - u_{0r})^2 - \sum_{r=1}^{p} a_r(\tilde{u}_r - u_{0r})^2 \\ &= a_{(\rho^*)}(\tau^*)^2 + a_{(\rho^*+1)}u_{0(\rho^*+1)}^2 - a_{(\rho^*)}\left(\tau^* + \frac{2a_{(\rho^*+1)}\epsilon}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)^2 \\ &\quad - a_{(\rho^*+1)}\left(u_{0(\rho^*+1)} - \frac{2a_{(\rho^*)}\epsilon}{a_{(\rho^*)}+a_{(\rho^*+1)}}\right)^2 \\ &= -\frac{4a_{(\rho^*)}a_{(\rho^*+1)}\epsilon}{a_{(\rho^*)}+a_{(\rho^*+1)}}\underbrace{(\tau^* - u_{0(\rho^*+1)})}_{-2\epsilon} - \frac{4a_{(\rho^*)}a_{(\rho^*+1)}^2\epsilon^2}{\left(a_{(\rho^*)}+a_{(\rho^*+1)}\right)^2} - \frac{4a_{(\rho^*)}^2a_{(\rho^*+1)}\epsilon^2}{\left(a_{(\rho^*)}+a_{(\rho^*+1)}\right)^2} \\ &= \frac{4a_{(\rho^*)}a_{(\rho^*+1)}\epsilon^2}{a_{(\rho^*)}+a_{(\rho^*+1)}} \geq 0, \end{aligned} \tag{41}$$

which leads to a contradiction and completes the proof. ■

References

Bach, F. (2008a). Consistency of the group Lasso and multiple kernel learning. *JMLR*, 9:1179–1225.

Bach, F. (2008b). Exploring large feature spaces with hierarchical multiple kernel learning. *NIPS*, 21.

Bach, F., Lanckriet, G., and Jordan, M. (2004). Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*.

Bottou, L. (1991). Stochastic gradient learning in neural networks. In *Proc. of Neuro-Nîmes*.

Cesa-Bianchi, N., Conconi, A., and Gentile, C. (2004). On the generalization ability of on-line learning algorithms. *IEEE Trans. on Inf. Theory*, 50(9):2050–2057.

Chapelle, O. and Rakotomamonjy, A. (2008). Second order optimization of kernel parameters. In *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*.

Collins, M. (2002). Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proc. of EMNLP*.

Collins, M., Globerson, A., Koo, T., Carreras, X., and Bartlett, P. (2008). Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *JMLR*.

Combettes, P. and Wajs, V. (2006). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). Efficient projections onto the L1-ball for learning in high dimensions. In *ICML*.

Duchi, J. and Singer, Y. (2009). Efficient online and batch learning using forward backward splitting. *JMLR*, 10:2873–2908.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso.

Hazan, E., Agarwal, A., and Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192.

Hofmann, T., Scholkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171.

Jenatton, R., Audibert, J.-Y., and Bach, F. (2009). Structured variable selection with sparsity-inducing norms. Technical report, arXiv:0904.3523.

Kloft, M., Brefeld, U., Sonnenburg, S., and Zien, A. (2010). Non-Sparse Regularization and Efficient Training with Multiple Kernels. *Arxiv preprint arXiv:1003.0079*.

Kowalski, M. and Torrésani, B. (2009). Structured sparsity: From mixed norms to structured shrinkage. In *Workshop on Signal Processing with Adaptive Sparse Structured Representations*.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. (2004). Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72.

Langford, J., Li, L., and Zhang, T. (2009). Sparse online learning via truncated gradient. *JMLR*, 10:777–801.

McDonald, R. T., Pereira, F., Ribarov, K., and Hajic, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proc. of HLT-EMNLP*.

Moreau, J. (1962). Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math*, 255:2897–2899.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2008). SimpleMKL. *JMLR*, 9:2491–2521.

Ratliff, N., Bagnell, J., and Zinkevich, M. (2006). Subgradient methods for maximum margin structured learning. In *ICML Workshop on Learning in Structured Outputs Spaces*.

Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*.

Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *JMLR*, 7:1565.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. *Proc. of CoNLL*.

Suzuki, T. and Tomioka, R. (2009). SpicyMKL. *Arxiv preprint arXiv:0909.5026*.

Taskar, B., Guestrin, C., and Koller, D. (2003). Max-margin Markov networks. In *NIPS*.

Tomioka, R. and Suzuki, T. (2010). Sparsity-accuracy trade-off in MKL. *Arxiv*.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *ICML*.

Wright, S., Nowak, R., and Figueiredo, M. (2009). Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493.

Xu, Z., Jin, R., King, I., and Lyu, M. (2009). An extended level method for efficient multiple kernel learning. *NIPS*, 21:1825–1832.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 68(1):49.

Zhao, P., Rocha, G., and Yu, B. (2008). Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*.

Zhou, Y., Jin, R., and Hoi, S. (2010). Exclusive Lasso for Multi-task Feature Selection. *JMLR*, 9:988–995.

Zien, A. and Ong, C. (2007). Multiclass multiple kernel learning. In *ICML*.

Zinkevich, M. (2003). Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *ICML*.