Chapter 13

Structured prediction

Chapter summary

- With appropriate modifications, we can design convex surrogates for output spaces that are arbitrarily complex and with generic loss functions, starting with multi-category classification.
- Like for binary classification, these convex surrogates lead to efficient algorithms that predict optimally given infinite amounts of data (Fisher consistency).
- Quadratic surrogates that extend the square loss lead to simple, intuitive, and consistent estimation procedures with well-defined decoding steps once a score function has been learned. They can be extended to smooth surrogates.
- Non-smooth surrogates can be defined in the general structured prediction framework, that extends the support vector machine.

In most of this book on supervised learning, we have focused on regression or binary classification, which led to estimating real-valued prediction functions directly when predicting a real-valued output (least-squares regression) or indirectly through convex surrogates (support vector machine or logistic regression) where the binary output in $\{-1, 1\}$ was obtained by taking the sign function. As shown in Section 4.1, the use of convex surrogates comes with strong theoretical guarantees in terms of achieving the Bayes error (that is, the optimal performance on unseen data).

In this chapter, we tackle arbitrary output spaces $\mathcal{Y}$, with arbitrary loss functions, which are ubiquitous in practice (see examples in Section 13.2). Most developments from Section 4.1 will extend with appropriate modifications.

We start in Section 13.1 with the natural extension to multi-category classification with the 0-1 loss, which directly extends binary classification, before describing in Section 13.2 a more general class of problems, referred to as structured prediction. We then present surrogate methods in Section 13.3 and their desirable properties before describing the two main classes, that is, smooth surrogates in Section 13.4 and non-smooth surrogates in Section 13.5. We then present generalization bounds in Section 13.6 and experiments in Section 13.7.

13.1 Multi-category classification

We dealt with binary classification with $\mathcal{Y} = \{-1, 1\}$ in Section 4.1.1 by estimating real-valued prediction functions and taking their signs. Going from 2 to $k > 2$ classes requires multi-dimensional vector-space valued functions. To preserve symmetry among classes, we will consider k-dimensional outputs. That is, for $\mathcal{Y} = \{1, \dots, k\}$, we will estimate a function $g : \mathcal{X} \to \mathbb{R}^k$, and predict as $f(x) \in \arg\max_{j \in \{1,\dots,k\}} g_j(x) \subset \mathcal{Y}$.

When $k = 2$, we recover our traditional framework by mapping $\{1, 2\}$ to $\{-1, 1\}$ and taking the sign of $g_2(x) - g_1(x)$, highlighting the general fact (valid for all k) that predictions are invariant by the addition of a constant vector to $g(x) \in \mathbb{R}^k$.

In the binary case, the convex surrogates we considered were all of the form $\Phi(yg(x))$. In the multi-category case, there is significantly more diversity. In Section 13.1.1 below, we describe the most commonly used convex surrogates and, when possible, the corresponding optimal predictors and their relationship with the Bayes predictor for the 0-1 loss, equal to $\arg\max_{z \in \{1,\dots,k\}} \mathbb{P}(y = z|x)$. Generalization bounds will then be derived, first for stochastic gradient descent used on linear models in Section 13.1.2 because it does not require any new developments, and then using Rademacher complexities in Section 13.1.3. In later sections, we show how this can be generalized to general output spaces.

Throughout this section on multi-category classification, we will identify elements y of $\{1, \dots, k\}$ with the corresponding canonical basis vector in $\mathbb{R}^k$, that is, the vector $\bar{y} \in \mathbb{R}^k$ with all zero components except a one at index y.

13.1.1 Extension of classical convex surrogates

All binary convex surrogates presented in Section 4.1.1 have natural extensions that we now present. We consider a label $y \in \{1, \dots, k\}$ (also identified to a canonical basis vector $\bar{y}$), and a vector-valued function $g : \mathcal{X} \to \mathbb{R}^k$. Our goal is to build a convex surrogate $S(y, g(x))$ (which is convex with respect to its second variable).

Softmax loss. We can extend the logistic loss and its relationship with maximum likelihood by considering the conditional model:

$$\mathbb{P}(y = j|x) = \frac{\exp(g_j(x))}{\sum_{i=1}^k \exp(g_i(x))} = \text{softmax}(g(x))_j,$$

by definition of the softmax function from $\mathbb{R}^k$ to the simplex in $\mathbb{R}^k$, defined as $\text{softmax}(u)_j = \exp(u_j)/\sum_{i=1}^k \exp(u_i)$.

The negative log-likelihood (often referred to as the cross-entropy loss) for this model is then equal to

$$\begin{aligned} S(y, g(x)) &= -\log \frac{\exp(g_y(x))}{\sum_{i=1}^k \exp(g_i(x))} = -g_y(x) + \log\Big(\sum_{j=1}^k \exp(g_j(x))\Big) \\ &= -\bar{y}^\top g(x) + \log\Big(\sum_{j=1}^k \exp(g_j(x))\Big). \end{aligned}$$

The minimizer of the surrogate expected risk $\mathbb{E}[S(y, g(x))]$ is then equal to $g^*(x)_j = \log \mathbb{P}(y = j|x) + c(x)$, for any function $c : \mathcal{X} \to \mathbb{R}$; thus, the predictor $\arg\max_{j\in\{1,\dots,k\}} g^*(x)_j$ is the Bayes predictor, which will lead to consistent estimation. A calibration function relating the excess surrogate risk and the 0-1 excess risk will be derived in Section 13.4 in the more general structured prediction case (with the same square root behavior as for logistic regression).

Square loss. The square loss has a natural extension $S(y, g(x)) = \|\bar{y} - g(x)\|_2^2$, with a minimizer of the surrogate expected risk equal to $g^*(x) = \mathbb{E}[\bar{y}|x]$. Again, the predictor $\arg\max_{j\in\{1,\dots,k\}} g^*(x)_j$ is the Bayes predictor (note that this is an instance of the "one vs. all" framework presented below), with a calibration function derived in Section 13.4, also with a square root behavior typical of smooth surrogates.

Hinge loss. The maximum-margin framework presented in Section 4.1.2 can be extended in several ways. We present the one that has natural extensions in structured prediction. The goal is to make $g_y(x)$ strictly larger than all others $g_j(x)$, for $j \neq y$, with potential slack, that is, we aim at finding the lowest $\xi \geqslant 0$, such that

$$\forall j \neq y, \ g_y(x) \geqslant g_j(x) + 1 - \xi.$$

The lowest such ξ can obtained in closed form, leading to the surrogate:

$$S(y, g(x)) = \sup_{j\in\{1,\dots,k\}} \big\{ 1_{y\neq j} + g(x)_j - g(x)_y \big\}.$$

Finding the minimizer of the expected surrogate risk is not as easy. We have for a given $x \in \mathcal{X}$,

$$\mathbb{E}[S(y, g(x))|x] = \sum_{i=1}^k \mathbb{P}(y = i|x) \sup_{j\in\{1,\dots,k\}} \big\{ 1_{i\neq j} + g(x)_j - g(x)_i \big\}.$$

Assuming without loss of generality that posterior probabilities given x are non-increasing, the global minimizer of the quantity above can be shown (proof left as an exercise) to be achieved for $g_1(x) \geqslant g_2(x) = \cdots = g_k(x)$, and we thus need to minimize

$$\mathbb{P}(y = 1|x)(1 + g_1(x) - g_2(x))_+ + (1 - \mathbb{P}(y = 1|x))(1 + g_2(x) - g_1(x))_+.$$

If $\mathbb{P}(y = 1|x) > 1/2$, then the optimal $g_1(x) - g_2(x)$ can be shown to be equal to 1, and the prediction is optimal. Otherwise, it is not, hence, we lose consistency in general, as for $k > 2$, it is not always the case that the posterior probability of the most likely class exceeds $1/2$. A consistent version of the non-smooth hinge loss will be discussed in Section 13.5.

One vs. all (♦). This class of techniques essentially solve k different binary optimization problems, by solving *independently* for each $g_j(x)$ predicting $y_j \in \{0,1\}$. Using convex surrogates for these problems and taking into account the mapping from $\{0,1\}$ to $\{-1,1\}$, the overall cost function is:

$$S(y, g(x)) = \sum_{j=1}^{k} \Phi((2y_j - 1) g_j(x)),$$

with Φ one of the convex surrogates from Section 4.1.1. For the square loss, we recover the multivariate square loss, but other losses can be used as well. The expected surrogate risk given $x \in \mathcal{X}$, is then equal to

$$\mathbb{E}[S(y, g(x))|x] = \sum_{j=1}^{k} \Big\{ \mathbb{P}(y=j|x)\Phi(g_j(x)) + (1 - \mathbb{P}(y=j|x))\Phi(-g_j(x)) \Big\}.$$

For a differentiable strictly convex function such that $\Phi(z) < \Phi(-z)$ for $z > 0$ (which excludes the hinge loss), minimizing it is done by setting $\mathbb{P}(y=j|x)\Phi'(g_j(x)) = (1 - \mathbb{P}(y=j|x))\Phi'(-g_j(x))$. One can then show that $g_j(x)$ is a strictly increasing function of $\mathbb{P}(y=j|x)$, and thus we get consistent predictions (in the population case), with also calibration functions (see Zhang, 2004b, Theorem 11).

Beyond. As reviewed by Zhang (2004b), there are many examples of convex surrogates to estimate the k functions $g_1, \dots, g_k : \mathcal{X} \to \mathbb{R}$, based on several principles. Reductions to binary classification problems go beyond one vs. all approaches, for example, by considering several subsets A of $\{1, \dots, k\}$ and solving the binary classification problems of deciding $y \in A$ vs. $y \notin A$. This approach based on error-correcting codes (Dietterich and Bakiri, 1994) will also be considered within the general surrogate framework.

Exercise 13.1 *Consider the following surrogate $S(y, g(x)) = \sum_{i \neq y} \Phi(-g_i(x))$, with the additional constraint that $\sum_{i=1}^{k} g_i(x) = 0$, with a strictly convex decreasing function Φ. Show that if g^* is the minimizer of $\mathbb{E}[S(y, g(x)]$, then for all $i, j \in \{1, \dots, k\}$, $\mathbb{P}(y=i|x) > \mathbb{P}(y=j|x) \Rightarrow g_i^*(x) > g_j^*(x)$. Conclude on the consistency of this convex surrogate.*

13.1.2 Generalization bound I: stochastic gradient descent

In these following two sections, we will consider generalization bounds for Lipschitz-continuous losses (all losses in the section above are Lipschitz-continuous, potentially once restricted to a bounded set), like done in Chapter 4 with estimation errors controlled by Rademacher complexities, and Chapter 5 using single-pass stochastic gradient descent (SGD). We start with SGD because the analysis can be done without the need for new tools.

Linear models. Since we will leverage convergence results for convex optimization algorithms, we need to consider linear models. We thus assume that we have a feature vector $\varphi : \mathcal{X} \to \mathbb{R}^d$ almost surely bounded by R in ℓ_2-norm, and that the vector-valued function $g : \mathcal{X} \to \mathbb{R}^k$ is parameterized linearly as $g^{(\theta)}(x) = \theta^\top \varphi(x)$, with $\theta = (\theta_1, \dots, \theta_k) \in \mathbb{R}^{d\times k}$. We aim to estimate θ restricted to a ball of radius D for a certain norm Ω.

Several candidates are natural for the norm Ω: the simplest one is the Frobenius norm defined through its square $\|\theta\|_{\mathrm{F}}^2 = \sum_{i=1}^k \|\theta_i\|_2^2 = \sum_{i=1}^k \sum_{j=1}^d \theta_{ji}^2$, which corresponds to the Euclidean norm for the matrix θ seen as a vector, and for which all results related to SGD will apply. Another classical norm is the nuclear norm (a.k.a. trace norm) defined as the sum of singular values of θ, which will push for low-rank θ with similar properties as the ℓ_1-norm in Section 8.3. Other norms, such as the ℓ_1-norm or "grouped" norms, could also be considered for variable selection. For these norms, optimization tools such as stochastic mirror descent from Section 11.1.3 are needed (see Exercise 13.2 for the nuclear norm).

Note finally that we could use (positive definite) kernel methods with the kernel trick from Section 7.4.5 to deal with infinite-dimensional feature vectors.

Sampling assumptions. Following the rest of the book, we assume that we have n i.i.d. pairs of observations $(x_i, y_i) \in \mathcal{X} \times \{1, \dots, k\}$, $i = 1, \dots, n$. Given the function $g^{(\theta)} : \mathcal{X} \to \mathbb{R}^k$ defined through the linear model above, we consider the expected risk $\mathcal{R}(f)$ for the predictor $f : \mathcal{X} \to \{1, \dots, k\}$ defined as $f(x) \in \arg\max_{i\in\{1,\dots,k\}} g^{(\theta)}(x)_i$ and the 0-1 loss. As already mentioned and shown in the subsequent section, the excess risk $\mathcal{R}(f) - \mathcal{R}^*$ (where $\mathcal{R}^*$ is the minimum risk overall measurable functions, not only the ones obtained from the model), will be bounded by an increasing function of the excess surrogate risk $\mathcal{R}_S(g) - \mathcal{R}_S^*$ (again $\mathcal{R}_S^*$ is the minimal value across all measurable functions). We thus focus on the excess surrogate risk.

Using the same decomposition as in Section 4.5.4, we consider an estimator $\hat\theta \in \mathbb{R}^{d\times k}$ that depends on the observations and subject to the constraint $\|\theta\|_{\mathrm{F}} \leqslant D$ (other norms could also be considered) for some real value D. A bound on the expected excess risk is obtained by the sum of the approximation error $\inf_{\|\theta\|_{\mathrm{F}}\leqslant D} \mathcal{R}_S(g^{(\theta)}) - \mathcal{R}_S^*$ and the estimation error $\mathcal{R}_S(g^{(\hat\theta)}) - \inf_{\|\theta\|_{\mathrm{F}}\leqslant D} \mathcal{R}_S(g^{(\theta)})$. We focus on the latter quantity, a random quantity we bound through its expectation.

We assume that the convex surrogate $S : \mathcal{Y} \times \mathbb{R}^k \to \mathbb{R}$ is Lipschitz-continuous with respect to its second variable, that is, its subgradients $S'(y, u) \in \mathbb{R}^k$ are bounded by G in ℓ_2-norm.

Single-pass SGD. We consider the following SGD iteration, for $t \in \{1, \dots, n\}$, with an arbitrary subgradient of the surrogate S:

$$\theta_t = \Pi_D\big(\theta_{t-1} - \gamma_t \varphi(x_t) S'(y_t, \theta_{t-1}^\top \varphi(x_t))^\top\big).$$

The analysis of Section 5.4 exactly applies, and with the choice $\gamma_t = \frac{D}{RG\sqrt{n}}$, we obtain the following generalization bound:

$$\mathbb{E}\big[\mathcal{R}_S(g_{(\theta_n)}) - \inf_{\|\theta\|_{\mathrm{F}} \leqslant D} \mathcal{R}_S(g^{(\theta)})\big] \leqslant \frac{DRG}{\sqrt{n}}, \tag{13.1}$$

which is exactly the same as for real-valued predictions.

Exercise 13.2 (Mirror descent for trace-norm penalty (♦)) *We consider the following mirror map on $\mathbb{R}^{d\times k}$, $\Psi(\theta) = \frac{1}{2}\|\sigma(\theta)\|_p^2$, where $\sigma(\theta)$ is the vector of singular values of θ. Show that for $p \in (1,2)$ is $(p-1)$-strongly-convex with respect to the norm $\|\sigma(\cdot)\|_p$. Show how to apply stochastic mirror descent on a nuclear norm ball and provide a convergence rate.*

13.1.3 Generalization bound II: Rademacher complexities (♦)

Another approach we followed in this book is to assume we can compute a minimizer of the empirical risk beyond linear models (in particular for neural networks). We thus assume a generic space of functions $\mathcal{G}$ of functions from $\mathcal{X} \to \mathbb{R}^k$, and define $\hat{g} \in \mathcal{G}$ as the minimizer of the empirical surrogate risk $\widehat{\mathcal{R}}(g) = \frac{1}{n}\sum_{i=1}^n S(y_i, g(x_i))$ over $g \in \mathcal{G}$. Following Section 4.2 and Section 4.5, the expected estimation error $\mathcal{R}_S(\hat{g}) - \inf_{g\in\mathcal{G}} \mathcal{R}_S(g)$ is then less than twice the Rademacher complexity

$$\mathrm{R}_n(S, \mathcal{G}) = \mathbb{E}\Big[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \varepsilon_i S(y_i, g(x_i))\Big],$$

where the expectation is taken with respect to the data and the Rademacher random variables $\varepsilon_1, \dots, \varepsilon_n \in \{-1, 1\}$. In the real-valued case, we used a contraction principle (Prop. 4.3) that allows us to get rid of the surrogate cost S as long as it is Lipschitz-continuous. Such a contraction principle also exists for vector-valued prediction functions (Maurer, 2016) and is presented in Prop. 13.1 below. Its application leads to

$$\mathrm{R}_n(S, \mathcal{G}) \leqslant \sqrt{2}G \cdot \mathbb{E}\Big[\sup_{g\in\mathcal{G}} \frac{1}{n}\sum_{i=1}^n (\varepsilon_i')^\top g(x_i)\Big], \tag{13.2}$$

where now each $\varepsilon_i' \in \{-1,1\}^k$, $i = 1, \dots, n$, is a *vector* of independent Rademacher random variables. This bound allows us to obtain a bound for all the function spaces that we considered in this book. We consider linear models below (and compare them to the SGD bound from above) and leave neural networks as an exercise.

Linear models. In the set-up of the previous section (linear models with Frobenius bound), we can further upper-bound in Eq. (13.2) as:

$$\begin{aligned}
\mathrm{R}_n(S,\mathcal{G}) &\leqslant \sqrt{2}G\cdot\mathbb{E}\Big[\sup_{g\in\mathcal{G}}\frac{1}{n}\sum_{i=1}^n(\varepsilon'_i)^\top g(x_i)\Big] = \sqrt{2}G\cdot\mathbb{E}\Big[\sup_{\|\theta\|_{\mathrm{F}}\leqslant D}\frac{1}{n}\sum_{i=1}^n(\varepsilon'_i)^\top\theta^\top\varphi(x_i)\Big]\\
&= \frac{D}{n}\sqrt{2}G\cdot\mathbb{E}\Big[\Big\|\sum_{i=1}^n\varphi(x_i)(\varepsilon'_i)^\top\Big\|_{\mathrm{F}}\Big] \leqslant \frac{DG}{n}\sqrt{2}\cdot\bigg(\mathbb{E}\Big[\Big\|\sum_{i=1}^n\varphi(x_i)(\varepsilon'_i)^\top\Big\|_{\mathrm{F}}^2\Big]\bigg)^{1/2}\\
&= \frac{DG}{n}\sqrt{2}\cdot\bigg(\mathbb{E}\Big[\sum_{i=1}^n\|\varphi(x_i)\|_2^2\|\varepsilon'_i\|_2^2\|_{\mathrm{F}}^2\Big]\bigg)^{1/2} \leqslant \frac{DGR\sqrt{2k}}{\sqrt{n}}.
\end{aligned}$$

We thus obtain the same bound as in Eq. (13.1), but with an extra factor of $\sqrt{k}$, showing that the Rademacher average technique, as opposed to the real-valued case, does not lead to the same result as SGD. However, it applies more generally to non-convex loss functions (as long as they are Lipschitz-continuous) and non-linear predictors.

Contraction principle (♦). We now provide a proof for the vector-valued contraction principle, taken from Maurer (2016). The proof follows the same structure as the proof of Prop. 4.3 for $k=1$, and we start from a key lemma.

Lemma 13.1 *Given any functions $b:\Theta\to\mathbb{R}$, $a:\Theta\to\mathbb{R}^k$ (no assumption) and $c:\mathbb{R}^k\to\mathbb{R}$ any 1-Lipschitz-functions (with respect to the ℓ_2-norm), we have, for $\varepsilon\in\{1,1\}$ a Rademacher random variable, and $\varepsilon'\in\mathbb{R}^n$ a vector of independent Rademacher random variables:*

$$\mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\varepsilon c(a(\theta))\Big\}\Big] \leqslant \mathbb{E}\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\sqrt{2}\sum_{j=1}^k\varepsilon'_j a_j(\theta)\Big\}\Big].$$

Proof (♦♦) Writing down explicitly the expectation with respect to ε, we get:

$$\begin{aligned}
\mathbb{E}_\varepsilon\Big[\sup_{\theta\in\Theta}\Big\{b(\theta)+\varepsilon c(a(\theta))\Big\}\Big] &= \frac{1}{2}\sup_{\theta\in\Theta}\Big\{b(\theta)+c(a(\theta))\Big\}+\frac{1}{2}\sup_{\theta\in\Theta}\Big\{b(\theta)-c(a(\theta))\Big\}\\
&= \sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{c(a(\theta))-c(a(\theta'))}{2}.
\end{aligned}$$

By taking the supremum over (θ,θ') and (θ',θ), and using Lipschitz-continuity of c, we further get the bound

$$\sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{|c(a(\theta))-c(a(\theta'))|}{2} \leqslant \sup_{\theta,\theta'\in\Theta}\frac{b(\theta)+b(\theta')}{2}+\frac{\|a(\theta)-a(\theta')\|_2}{2}.$$

In the proof of Prop. 4.3 (for $k=1$), we then applied the same set of equalities to obtain the desired result without the constant $\sqrt{2}$. Here, we can use Khintchine's inequality from Lemma 11.1, that is, for any vector $v\in\mathbb{R}^k$, $\|v\|_2\leqslant\sqrt{2}\cdot\mathbb{E}\big[\big|\sum_{j=1}^k\varepsilon'_j v_j\big|\big]$ for any vector $v \in \mathbb{R}^k$ and independent Rademacher random variables $\varepsilon'_1, \dots, \varepsilon'_n$. This leads to the bound:

$$\begin{aligned}
& \sup_{\theta,\theta' \in \Theta} \frac{b(\theta)+b(\theta')}{2} + \frac{\sqrt{2}}{2} \mathbb{E}\Big[\Big| \sum_{j=1}^k \varepsilon'_j (a_j(\theta) - a_j(\theta'))\Big|\Big] \\
\leqslant\ & \mathbb{E}\Big[\sup_{\theta,\theta' \in \Theta} \frac{b(\theta)+b(\theta')}{2} + \frac{\sqrt{2}}{2} \Big| \sum_{j=1}^k \varepsilon'_j (a_j(\theta) - a_j(\theta'))\Big|\Big] \\
& \qquad \text{using properties of expectations and suprema,} \\
=\ & \mathbb{E}\Big[\sup_{\theta,\theta' \in \Theta} \frac{b(\theta)+b(\theta')}{2} + \frac{\sqrt{2}}{2} \sum_{j=1}^k \varepsilon'_j (a_j(\theta) - a_j(\theta'))\Big] \text{ by symmetry,} \\
\leqslant\ & \mathbb{E}\Big[\sup_{\theta \in \Theta} \Big\{ b(\theta) + \sqrt{2} \sum_{j=1}^k \varepsilon'_j a_j(\theta) \Big\}\Big] \text{ which is the desired result.}
\end{aligned}$$

■

Proposition 13.1 (Vector-valued contraction principle) *Given any functions $b : \Theta \to \mathbb{R}$, $a_i : \Theta \to \mathbb{R}^k$ (no assumption) and $c_i : \mathbb{R}^k \to \mathbb{R}$ any 1-Lipschitz-functions (with respect to the ℓ_2-norm), for $i = 1, \dots, n$, we have, for $\varepsilon \in \mathbb{R}^n$ a vector of independent Rademacher random variables, and $\varepsilon' \in \mathbb{R}^{n \times k}$ a matrix of independent Rademacher random variables:*

$$\mathbb{E}\Big[\sup_{\theta \in \Theta} \Big\{ b(\theta) + \sum_{i=1}^n \varepsilon_i c_i(a_i(\theta)) \Big\}\Big] \leqslant \mathbb{E}\Big[\sup_{\theta \in \Theta} \Big\{ b(\theta) + \sqrt{2} \sum_{i=1}^n \sum_{j=1}^k \varepsilon'_{ij} a_{ij}(\theta) \Big\}\Big].$$

Proof (♦♦) We consider a proof by induction on n. The case $n = 0$ is trivial, and we show how to go from $n \geqslant 0$ to $n+1$. We thus consider $\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_{n+1}} \Big[\sup_{\theta \in \Theta} \Big\{ b(\theta) + \sum_{i=1}^{n+1} \varepsilon_i c_i(a_i(\theta)) \Big\}\Big]$ and apply Lemma 13.1 with fixed $\varepsilon_1, \dots, \varepsilon_n$, leading to the bound

$$\mathbb{E}_{\varepsilon_1,\dots,\varepsilon_n,\varepsilon'_{n+1}} \Big[\sup_{\theta \in \Theta} \Big\{ b(\theta) + \sum_{j=1}^k \varepsilon'_{n+1,j} a_{n+1,j}(\theta) + \sum_{i=1}^n \varepsilon_i c_i(a_i(\theta)) \Big\}\Big],$$

and apply the induction hypothesis with ε'_{n+1} fixed to obtain the desired result. ■

Exercise 13.3 (Multi-cateagory classification with neural networks (♦)) *We consider the neural network with outputs in $\mathbb{R}^k$, that is, using the notations from Section 9.2, $f(x) = \sum_{j=1}^m \sigma(w_j^\top x + b_j)\eta_j$, with now $\eta_j \in \mathbb{R}^k$. Extend the estimation error analysis from Section 9.2.3 by imposing a constraint on $\sum_{j=1}^m \|\eta_j\|_2$.*

13.2 General set-up and examples

Now that the multi-category classification has been presented, we consider the same general set-up presented earlier in Section 2.2, that is, we want to predict a variable

$y \in \mathcal{Y}$ from some $x \in \mathcal{X}$, and given a prediction $z \in \mathcal{Y}$, we incur the loss $\ell(y, z)$, with the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Like in Section 2.2, given a test distribution p on $\mathcal{X} \times \mathcal{Y}$, we can define the Bayes predictor

$$f^*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \int_{\mathcal{Y}} \ell(y, z) dp(y|x) \tag{13.3}$$

in the usual way. While it led to simple closed-form formulas for the 0–1 loss and binary classification, this will not always be the case. Nevertheless, our goal will still be to achieve its (optimal) performance at a reasonable computational cost.

13.2.1 Examples

We now consider classic examples with their applicative motivations in natural language processing, biology, or computer vision—see more examples by Nowak et al. (2019) and Ciliberto et al. (2020):

- **Multi-category classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k \times k}$, with $\ell(i, j) = L_{ij}$. The usual 0-1 loss from Section 13.1 corresponds to $L_{ij} = 1_{i \neq j}$, but in most applications, errors do not have the same cost (for example, in spam prediction, classifying a legitimate email as spam costs much more than the opposite).

- **Robust regression**: $\mathcal{Y} = \mathbb{R}$, with $\ell(y, z) = \rho(y - z)$ and typically ρ non convex. When ρ is convex, such as $\rho(\delta) = |\delta|$ or $\rho(\delta) = \delta^2$, there is no need for a surrogate framework, but then regression may be non-robust to strong outlier perturbations. Having a non-convex ρ, such as, $\rho(\delta) = 1 - \exp(-\delta^2)$ leads to robust regression.

- **Ordinal regression**: this is a particular case of the situation above, where the loss matrix has a specific structure where the loss L_{ij} is increasing in $|i - j|$. This is common when using a rating system with a few discrete levels. One possibility is to ignore the discrete structure of the loss and use least-squares regression together with rounding, but this does not lead to optimal predictions.

- **Multiple labels**: $\mathcal{Y} = \{-1, 1\}^k$, with cardinality 2^k, with the traditional Hamming loss $\ell(y, z) = \frac{1}{2}\|y - z\|_1 = \frac{1}{4}\|y - x\|_2^2$, which counts the number of mistakes and will be a running example in this chapter. Other scores, such as precision/recall[1] or F-scores[2], are typically used (and may not be symmetric) and can be treated as well with the frameworks presented in this chapter. These are detailed by Nowak et al. (2019).

  Multiple label prediction is common in multimedia applications, where there are potentially k objects in a document, and one wants to predict which ones are present.

- **Permutations**: $\mathcal{Y}$ is the set of permutations among m elements, that is, y a bijection from $\{1, \dots, m\}$ to $\{1, \dots, m\}$. We have then $|\mathcal{Y}| = m!$. A common loss function is the "pairwise disagreement", equal to $\ell(y, z) = \sum_{i,j=1}^m 1_{y(i) > y(j)} 1_{z(i) < z(j)}$,

[1] See https://en.wikipedia.org/wiki/Precision_and_recall.

[2] See https://en.wikipedia.org/wiki/F-score.

but other losses such as the discounted cumulative gain[3] can be used, or losses of the form $\sum_{i=1}^{m} \ell_i(a_{y(i)} - b_{(y(i)})$ for vectors $a, b \in \mathbb{R}^m$ and functions $\ell_i : \mathbb{R} \to \mathbb{R}$. Predicting permutations occurs in information retrieval and ranking problems where the permutation encodes a user's preferences over a set of m items. See Nowak et al. (2019) and references therein for a review of classical losses used in practice.

- **Sequences**: $\mathcal{Y}$ is the set of sequences of potentially arbitrary lengths over some alphabet; this has applications in natural language processing (e.g., translation from one language to another), computational biology (DNA basis or amino-acid sequences), or econometrics/finance (prediction of time series, where the alphabet is usually not finite). The cardinality of y is thus large (or infinite), and the Hamming loss is commonly used.
- **Trees, graphs**: $\mathcal{Y}$ is set of potentially labelled graphs over some vertices. Classic examples include the prediction of molecules (which can be represented as graphs) or the grammatical analysis of sentences in natural language processing.

Why is it difficult? Structured prediction is challenging for two reasons:

- Computationally: we need to predict large structured (often discrete) objects from real-valued outputs.
- Statistically: there is a potential curse of dimensionality in both k (the underlying dimension of the problem, to be defined later precisely) *and* the input d, in addition to a complicated combinatorial structure.

Our goal is to obtain polynomial-time algorithms in k, n, and d to attain the optimal prediction, that is, we aim to obtain:

1. Computational tractability by introducing convex surrogates (to use convex optimization) and efficient decoding steps (often dedicated algorithms).
2. Fisher consistency (excess risk goes to zero in the population case) and calibration (sub-optimality for the convex surrogate leads to sub-optimality for the true risk).

Following the rest of the book, we will always go through vector-space valued prediction functions. Thus, there will always be two components:

1. Learning some scores from data, implicitly and explicitly, in a Hilbert space $\mathcal{H}$ or $\mathbb{R}^k$, where k is the (potentially implicit) "affine dimension" of $\mathcal{Y}$.
2. Decoding step to go from score functions to predictions (obvious and somewhat overlooked in the binary classification case, as this was simply the sign).

From one learning framework per situation to a general framework. The development of structured prediction methods has seen two streams of work: first, methods dedicated to specific instances (in particular, cost-sensitive multi-category classification, ranking, or learning with multiple labels), then generic frameworks that encompass all the particular cases. In this book, we focus on the latter set of techniques.

[3] See https://en.wikipedia.org/wiki/Discounted_cumulative_gain.

13.2.2 Structure encoding loss functions

To achieve guaranteed predictive performance, we will need to impose some low-dimension vectorial structure, which in turn imposes some specific structure within $\mathcal{Y}$, hence the name "structured prediction". More precisely, we will assume that we have two embeddings of the label space $\mathcal{Y}$ into the same Hilbert space $\mathcal{H}$, that is, two maps $\chi, \psi : \mathcal{Y} \to \mathcal{H}$ and a constant $c \in \mathbb{R}$, such that

$$\forall (y,z) \in \mathcal{Y} \times \mathcal{Y},\ \ell(y,z) = c + \langle \chi(z), \psi(y) \rangle. \tag{13.4}$$

This assumption is called "structure encoding loss function" (SELF) (Ciliberto et al., 2020). This can be an implicit or explicit embedding (see examples below). Note that the representation is not unique as given a pair (χ,ψ), any pair $(V\chi, V^{-*}\psi)$ is valid for any invertible operator.

⚠ There are *two* different embeddings of outputs in $\mathcal{Y}$, while typically one for the inputs in $\mathcal{X}$.

Bayes predictor. With the assumption in Eq. (13.4) above, we can now express the optimal predictor in Eq. (13.3) as:

$$f^*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \Big\langle \chi(z), \int_{\mathcal{Y}} \psi(y) dp(y|x) \Big\rangle. \tag{13.5}$$

Thus, to obtain Fisher consistency, it is sufficient to estimate well the conditional expectation $\int_{\mathcal{Y}} \psi(y)dp(y|x) \in \mathcal{H}$; this is what smooth surrogates will do in Section 13.4. However, what is only needed is, in fact, sufficient knowledge of this conditional expectation to perform the computation of $f^*(x)$ above. This will lead to non-smooth surrogates in Section 13.5.

Examples. We can now revisit the list of losses described in Section 13.2.1 to check if a SELF decomposition exists. In our analysis, we will need a bound on $R_\ell = \sup_{z\in\mathcal{Y}} \|\chi(z)\|$, which we also provide here.

- **Binary classification**, with $\mathcal{Y} \in \{-1,1\}$ and the 0-1 loss: $\mathcal{H} = \mathbb{R}$, $\chi(y) = -y/2$ and $\psi(z) = z$, since $\ell(y,z) = 1_{y \neq z} = \frac{1}{2} - \frac{yz}{2}$, with $R_\ell = 1/2$.
- **Multi-category classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k \times k}$, with $\ell(i,j) = L_{ij}$. This corresponds to the usual "one-hot" encoding of discrete distributions, where $\psi(i) \in \mathbb{R}^k$ is the i-th element of the canonical basis. We then have $\ell(i,j) = L_{ij} = \psi(i)^\top L \psi(j)$, that is, $\chi(j) = L\psi(j)$. For this case, we have $R_\ell = \sup_{j \in \{1,\dots,k\}} \|L(:,j)\|_2$. In particular, for the 0-1 loss, we have $L_{ij} = 1_{i \neq j}$, and we can write $\ell(i,j) = 1 - \psi(i)^\top \psi(j)$, and we can take $\chi(i) = -\psi(i)$, with $R_\ell = 1$.

  We can also choose to have a feature map ψ with values in $\{-1,1\}$ instead of $\{0,1\}$, in particular for the general reduction to binary classification problems.
- **Robust regression**: $\mathcal{Y} = \mathbb{R}$, with the loss $\ell(y,z) = 1 - \exp\big[-(y-z)^2\big]$, which can be written as, using the Fourier transform, $\ell(y,z) = 1 - \frac{1}{2\sqrt{\pi}} \int_{-\infty}^{\infty} \exp(-\omega^2/4) \cos\omega(x-z) d\omega$, which leads to the existence of an infinite-dimensional $\mathcal{H}$.

Indeed, we can select $\mathcal{H}$ to be the set of square integrable functions from $\mathbb{R}$ to $\mathbb{R}^2$, with $\psi(y)(\omega) = e^{-\omega^2/8}\binom{\cos \omega y}{\sin \omega y}$, and $\chi(z)(\omega) = -\frac{1}{2\sqrt{\pi}} e^{-\omega^2/8}\binom{\cos \omega z}{\sin \omega z}$, leading to $R_\ell^2 = \frac{1}{4\pi}\int_{-\infty}^{\infty} \exp(-\omega^2/4) = \frac{1}{2\sqrt{\pi}}$.

- **Multiple labels**: for $\mathcal{Y} = \{-1,1\}^k$, the traditional Hamming loss can be rewritten as $\ell(y,z) = \frac{k}{2} - \frac{1}{2} y^\top z$. We then have, $\psi(y) = y$ and $\chi(z) = -z/2$, and $R_\ell = \sqrt{m}$.

  Note here that choosing the 0-1 loss is not advocated as it is non-zero as soon as a single mistake is made, and it requires a lot of observations in practice (as can be seen by having a constant R_ℓ that grows exponentially in k.

- **Permutations**: for the pairwise disagreement, we have directly $\mathcal{H} = \mathbb{R}^k$ with $k = m(m-1)$, with $\psi(y)_{ij} = 1_{y(i) > y(j)}$ and $\chi(z)_{ij} = 1_{z(i) < z(j)}$ for $i \neq j$, and $R_\ell \leqslant m$. For the loss $\ell(y,z) = \sum_{i=1}^m (a_{y(i)} - b_{(y(i))})^2$, we have $\psi(y) = y$ and $\chi(z) = -2 * z$, with $R_\ell \leqslant \sqrt{2(m+1)}$.

- **Sequences**: we consider binary sequences for simplicity, that is, $\mathcal{Y} = \{-1,1\}^m$, but it extends more generally to all factor graphs (Wainwright and Jordan, 2008) and types of labels. Using the Hamming loss like for multiple labels ignores the sequential structure and does not enforce any notion of consistency between two successive elements of the sequence. On top of the feature $y_1, \dots, y_m \in \{-1,1\}$, we can add the features $y_1 y_2, y_2 y_3, \dots, y_{m-1} y_m \in \{-1,1\}$, that allows to consider losses that encourages perfectly predicted sequence of size 2, for example, by considering the loss $\ell(y,z) = \sum_{j=1}^{m-1} 1_{y_j \neq z_j \text{ or } y_{j+1} \neq z_{j+1}} = \sum_{j=1}^{m-1} \big\{ y_j z_j + y_{j+1} z_{j+1} - y_j z_j y_{j+1} z_{j+1} \big\}$.

⚠ Like for binary classification or regression, the loss choice is independent of the function space considered (local averaging, kernels, neural nets).

Reduction to binary problems. We have encountered several examples above, where the feature map Ψ has binary values in $\{-1,1\}^m$. We will see below that natural convex surrogates end up simply considering each of the m labels independently (ignoring their potential dependency, i.e., in the ranking case, where components of $\Psi(y)$ are $1_{y(i) < y(j)}$, not all values are possible). This can be useful in structured cases like sequence models or ranking, but also in multi-category classification with the 0-1 loss.

13.3 Surrogate methods

In this section, our main concern will be to obtain consistent convex surrogates, convex so that we can run efficient algorithms from Chapter 5, consistent so that we are sure that given sufficient amounts of data and sufficiently flexible models, predictions are optimal. In particular, this will allow us to derive generalization bounds in Section 13.6.

13.3.1 Score functions and decoding step

Binary classification. In this book, we have performed binary classification by learning a real-valued function $g : \mathcal{X} \to \mathbb{R}$, and then predicting with $f(x) = \text{sign}(g(x)) \in \{-1, 1\}$. In the language of this chapter, we have learned a real-valued score function and applied a specific decoding step from $\mathbb{R}$ to $\{-1, 1\}$ (the sign function). We now present the general surrogate framework.

General surrogate framework. In this chapter, we will consider functions $f : \mathcal{X} \to \mathcal{Y}$ that can be written as:

$$f(x) = \text{dec} \circ g(x),$$

where

- $g : \mathcal{X} \to \mathcal{H}$ is a function with values in the vector space $\mathcal{H}$, referred to as a score function.[4]
- $\text{dec} : \mathcal{H} \to \mathcal{Y}$ is the decoding function.

We then need a surrogate loss $S : \mathcal{Y} \times \mathcal{H} \to \mathbb{R}$, that will be used to form empirical and expected surrogate risks:

$$\hat{\mathcal{R}}_S(g) = \frac{1}{n} \sum_{i=1}^{n} S(y_i, g(x_i)) \quad \text{and} \quad \mathcal{R}_S(g) = \mathbb{E}\big[S(y, g(x))\big].$$

For binary classification where $\mathcal{Y} = \{-1, 1\}$, we had $S(y, g(x)) = \Phi(yg(x))$ for Φ a convex function.

13.3.2 Fisher consistency and calibration functions

Following the same definition as in Section 4.1, we denote $\mathcal{R}_S^*$ the minimim S-risk, that is the infimum over all functions from $\mathcal{X}$ to $\mathcal{H}$ of $\mathcal{R}_S(g) = \mathbb{E}\big[S(y, g(x))\big]$. It is equal to:

$$\mathcal{R}_S^* = \mathbb{E}\Big[\inf_{h \in \mathcal{H}} \mathbb{E}\big[S(y, h)|x\big]\Big].$$

The loss is said "Fisher-consistent" if we can get an arbitrary small excess risk $\mathcal{R}(f) - \mathcal{R}^*$ for $f = \text{dec} \circ g$, as soon as the excess S-risk of g is sufficiently small. In other words, perfectly minimizing the S-risk should lead to the Bayes predictor.

A stronger property that enables the transfer of convergence rates for the excess S-risk to the excess risk is the existence of a *calibration function*, that is, an increasing function $H : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\mathcal{R}(\text{dec} \circ g) - \mathcal{R}^* \leqslant H\big[\mathcal{R}_S(g) - \mathcal{R}_S^*\big]$. Note that, like in the binary classification case in Section 4.1.3, a more refined notion of consistency can be defined and studied, see, e.g., Long and Servedio (2013).

[4] In statistics, the score function often refers to the gradient of the log-density with respect to parameters. There is no link between these two definitions.

13.3.3 Main surrogate frameworks

As described in Section 4.1, for binary classification, we saw two classes of convex surrogates:

- Smooth surrogates, where the predictor minimizing the expected surrogate risk led to a complete description of the conditional distribution of y given x, that is, since we had only two outcomes, knowledge of $\mathbb{E}[y|x]$. Classic examples were the square loss and the logistic loss. Then, when going from the excess surrogate risk to the true excess risk, the calibration function was the square root.
- Non-smooth surrogates, where the predictor minimizing the expected surrogate risk already provided a thresholded version, that is, $\text{sign}(\mathbb{E}[y|x])$. The calibration function, however, did not exhibit a square root behavior but rather a (better) linear behavior.

In this chapter, we will present extensions of these two sets of surrogates: (1) least-squares (or more generally smooth surrogates), (2) max-margin (non-smooth that estimates the discrete estimator directly), as they come with efficient algorithms and guarantees. But there are other related frameworks that we will not study (Osokin et al., 2017; Lee et al., 2004; Blondel et al., 2020). In particular, probabilistic graphical models in the form of conditional random fields are popular (Sutton and McCallum, 2012).

13.4 Smooth/quadratic surrogates

We first look at a class of techniques that extends the square and logistic losses beyond binary classification for the whole class of structure encoding loss functions. We first start with quadratic surrogates, following Ciliberto et al. (2020), where the analysis is the simplest and most elegant.

13.4.1 Quadratic surrogate

Given the SELF decomposition in Eq. (13.4), we consider estimating a score function $g : \mathcal{X} \to \mathcal{H}$ with the following surrogate function:

$$S(y, g(x)) = \|\psi(y) - g(x)\|^2,$$

for the Hilbert norm $\|\cdot\|$. In other words, we aim at directly estimating $\mathbb{E}\big[\psi(y)|x\big]$ for every $x \in \mathcal{X}$. The decoding function is then naturally

$$\text{dec}(s) \in \arg\min_{z\in\mathcal{Y}} \langle \chi(z), g(x)\rangle,$$

since, when $g(x) = \mathbb{E}\big[\psi(y)|x\big]$, it leads to $\arg\min_{z\in\mathcal{Y}} \mathbb{E}\big[\langle\chi(z), \psi(y)\rangle|x\big] = \arg\min_{z\in\mathcal{Y}} \mathbb{E}\big[\ell(y,z)|x\big]$, which is the optimal predictor.

For the binary classification case, it leads to the square loss framework from Section 4.1.1, but in the general case, it extends to the many situations alluded to earlier. The decoding steps will be described in Section 13.4.3.

When the loss function is induced by a positive-definite kernel, then this framework is also referred to as output kernel regression (see, e.g., Brouard et al., 2016), or kernel dependency estimation (Weston et al., 2002).

13.4.2 Theoretical guarantees

For the framework proposed above, we can prove a precise calibration result by leveraging the properties of the square loss. We first notice that

$$\mathcal{R}_S(g) - \mathcal{R}_S^* = \mathbb{E}\Big[\big\|g(x) - \mathbb{E}[\psi(y)|x]\big\|^2\Big]. \tag{13.6}$$

Moreover, by construction, the function defined by $g^*(x) = \mathbb{E}[\psi(y)|x]$ is the minimizer of the expected S-risk, and the Bayes predictor is indeed $f^* = \mathrm{dec} \circ g^*$.

We can then express the excess risk using the decomposition of the loss as:

$$\begin{aligned}
& \mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}^* \\
= \; & \mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}(\mathrm{dec} \circ g^*) \\
= \; & \mathbb{E}\Big[\mathbb{E}\big[\ell(y, \mathrm{dec} \circ g(x)) - \ell(y, \mathrm{dec} \circ g^*(x))\big|x\big]\Big] \\
= \; & \mathbb{E}\Big[\mathbb{E}\big[\big\langle \psi(y), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big|x\big]\Big] \text{ by the SELF decomposition,} \\
= \; & \mathbb{E}\Big[\big\langle \mathbb{E}\big[\psi(y)|x\big], \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big] \text{ by moving expectations,} \\
= \; & \mathbb{E}\Big[\big\langle \mathbb{E}\big[\psi(y)|x\big] - g(x), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big] \\
& \qquad\qquad + \mathbb{E}\Big[\big\langle g(x), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big],
\end{aligned}$$

by adding and subtracting $g(x)$. The definition of the decoding function implies the negativity of the second term. Thus, we get:

$$\begin{aligned}
\mathcal{R}(\mathrm{dec} \circ g) - \mathcal{R}^* \; & \leqslant \; \mathbb{E}\Big[\big\langle \mathbb{E}\big[\psi(y)|x\big] - g(x), \chi(\mathrm{dec} \circ g(x)) - \chi(\mathrm{dec} \circ g^*(x))\big\rangle\big]\Big] \\
& \leqslant \; 2 \sup_{z \in \mathcal{Y}} \|\chi(z)\| \cdot \mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|\Big] \text{ using Cauchy-Schwarz inequality,} \\
& \leqslant \; 2 \sup_{z \in \mathcal{Y}} \|\chi(z)\| \cdot \sqrt{\cdot \mathbb{E}\Big[\big\|\big\langle \mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|^2\Big]} \text{ using Jensen's inequality,} \\
& = \; 2R_\ell \cdot \sqrt{\mathcal{R}_S(g) - \mathcal{R}_S^*} \text{ because of Eq. (13.6),}
\end{aligned}$$

which is precisely a calibration function result. A key feature of this result is that the constant R_ℓ typically does not explode, even for sets $\mathcal{Y}$ with large cardinality (see examples in Section 13.2.2). To get a learning bound, we then need to use learning bounds for multivariate least-squares regression, which behave similarly to univariate least-squares regression (see Section 13.6). For example, if we assume that the target function $g^*(x) = \mathbb{E}[\psi(y)|x]$ from $\mathcal{X} \to \mathcal{H}$ is in the space of functions that we are using for learning, then penalized least-squares with the proper choice of regularization parameter will lead to explicit convergence rates. Otherwise, we need to let the parameter go to zero to obtain universal consistency. See Ciliberto et al. (2020) for more details.

13.4.3 Linear estimators and decoding steps

When the function g is linear in the observations $\psi(y_i)$, $i = 1, \dots, n$ (e.g., local averaging methods from Section 6.2.1, or kernel methods from Section 7.6.1), that is,

$$g(x) = \sum_{i=1}^{n} w_i(x)\psi(y_i),$$

for well-defined functions $w_i : \mathcal{X} \to \mathbb{R}$, we see that the decoding step is

$$\mathrm{dec}(s) \in \arg\min_{z \in \mathcal{Y}} \Big\langle \chi(z), \sum_{i=1}^{n} w_i(x)\psi(y_i) \Big\rangle = \arg\min_{z \in \mathcal{Y}} \sum_{i=1}^{n} w_i(x)\ell(y_i, z), \tag{13.7}$$

that is, there is no need to know the loss decomposition to run the algorithm. This makes the decoding step even easier with the following examples:

- **Robust regression**: $\mathcal{Y} = \mathbb{R}$, with the loss $\ell(y, z) = 1 - \exp\big[-(y - z)^2\big]$. Eq. (13.7) then leads to
$$\arg\max_{z \in \mathbb{R}} \sum_{i=1}^{n} w_i(x) \exp\big[-(y_i - z)^2\big],$$
which is a one-dimensional optimization problem that can be solved by grid search.
- **Multi-category classification**: $\mathcal{Y} = \{1, \dots, k\}$ and a loss matrix $L \in \mathbb{R}^{k \times k}$, with $\ell(i, j) = L_{ij}$. Eq. (13.7) then leads to $\arg\max_{z \in \{1,\dots,k\}} \sum_{i=1}^{n} w_i(x) L_{iz}$.
- **Multiple labels**: $\mathcal{Y} = \{-1, 1\}^k$ with $\ell(y, z) = \frac{k}{2} - \frac{1}{2} y^\top z$. Eq. (13.7) then leads to $\arg\max_{z \in \{-1,1\}^k} z^\top \sum_{i=1}^{n} w_i(x) y_i$, which leads to a closed-form formula for z.
- **Permutations**: for the pairwise disagreement, the optimization problem does not have a closed form anymore and is an instance of a hard combinatorial problem ("minimum weighted feedback arc set"), which can be solved for small m, and with simple approximation algorithms otherwise (see Ciliberto et al., 2020).
- **Sequences**: When using separable loss functions, we return to the classical multiple-label setups. However, when using losses over consecutive pairs, we need to minimize with respect to $z \in \{-1, 1\}^m$ a function of the form $\sum_{j=1}^{m} u_j z_j + \sum_{j=1}^{m-1} v_j z_j z_{j+1}$, which can be done in time $O(m)$ by message passing algorithms (see, e.g., Murphy, 2012).

13.4.4 Smooth surrogates (♦)

Following Nowak-Vila et al. (2019) and as done in Section 4.1, we can also consider smooth surrogate functions of the form:

$$S(y, g(x)) = c(y) - 2\langle \psi(y), g(x) \rangle + 2a(g(x)),$$

where $a : \mathcal{H} \to \mathbb{R}$ is convex and β-smooth, that is, for any $h, h' \in \mathcal{H}$, $a(h') \leqslant a(h) + \langle a'(h), h' - h\rangle + \frac{\beta}{2}\|h - h'\|^2$. We also assume that $a(0) = 0$ and that the domain of its Fenchel conjugate includes all $\psi(y)$ for $y \in \mathcal{Y}$. The square loss corresponds to $a(h) = \frac{1}{2}\|h\|^2$ and $c(y) = \|\psi(y)\|^2$.

We consider the decoding function $\mathrm{dec} : \mathcal{H} \to \mathcal{Y}$ equal to

$$\mathrm{dec}(h) \in \arg\min_{z \in \mathcal{H}} \ \chi(z)^\top a'(h).$$

For the square loss, we recover exactly the quadratic surrogate.

Examples. Three examples are particularly interesting:

- **Softmax regression:** for multi-category classification, we can always take $\psi(y) = \bar{y} \in \mathbb{R}^k$ the "one-hot" encoding of $y \in \{1, \dots, k\}$. The convex hull of all $\psi(y)$ for $y \in \mathcal{Y}$ is then the simplex in $\mathbb{R}^k$. Softmax regression corresponds to $a(y) = \log\big(\sum_{j=1}^k \exp(x_j)\big)$.
- **Reduction to binary logistic regression:** when $\psi(y) \in \{-1, 1\}^m$, we can consider $a(h) = \sum_{i=1}^m \log \frac{1}{2}(\exp(h_i/2) - \exp(-h_i/2))$, leading to independent logistic regression
- **Graphical models (♦):** The examples above can be made more general using the graphical model framework. We consider sequences in $\{-1, 1\}^m$ for simplicity, but this extends to more general situations, that is, more complex graphical models (see, e.g., Murphy, 2012). For $\psi(y)$ composed of all y_j and $y_j y_{j+1}$. To build the function a, we consider the convex hull of all $\psi(y)$, for $y \in \mathcal{Y}$, and for any elements of this convex hull (which corresponds to a probability distribution on $\mathcal{Y}$), we consider its negative entropy which is a convex function b. We then take a to be the Fenchel conjugate of b.

  For $\psi(y) = y$, we recover independent logistic regression, while for the sequence models, we recover conditional random fields (Sutton and McCallum, 2012).
- **"Perturb-and-MAP":** In situations where one can efficiently maximize linear functions of $\psi(y)$ with respect to $y \in \mathcal{Y}$. In other words, we can compute the convex function $a_0(z) = \max_{y \in \mathcal{Y}} \psi(y)^\top z$. Then we can make it smooth using stochastic smoothing, as presented in Section 11.2, that is, define $a_\sigma = \mathbb{E}\big[a_0(z + \sigma u)\big]$ for a random variable $u \in \mathbb{R}^k$ (typically a Gaussian). When we use Gumbel distributions for u, and $\mathcal{Y} = \{1, \dots, k\}$ with $\psi(y) = \bar{y}$, we recover the softmax function, but the framework is mode generally applicable (see Berthet et al., 2020).

Calibration function. We then have, by definition of the Fenchel-conjugate $a^*(u) = \sup_{h \in \mathcal{H}} \langle u, h\rangle - a(h)$:

$$\begin{aligned}
\mathcal{R}_S(g) &= \mathbb{E}\Big[\mathbb{E}[c(y)|x] - 2\langle \mathbb{E}[\psi(y)|y], g(x)\rangle + 2a(g(x))\Big] \\
\mathcal{R}_S^* &= \mathbb{E}\Big[\mathbb{E}[c(y)|x] + \inf_{h \in \mathcal{H}} -2\langle \mathbb{E}[\psi(y)|x], h\rangle + 2a(h)\Big] \\
&= \mathbb{E}\big[\mathbb{E}[c(y)|x] - 2a^*(\mathbb{E}[\psi(y)|x])\big], \text{ by definition of } a^*,
\end{aligned}$$

leading to a compact expression of the excess S-risk and a lower bound:

$$\begin{array}{rcl}\mathcal{R}_S(g) - \mathcal{R}_S^* & = & \mathbb{E}\big[- 2\langle \mathbb{E}[\psi(y)|x], g(x)\rangle + 2a(g(x)) + 2a^*(\mathbb{E}[\psi(y)|x])\big] \\ & \geqslant & \dfrac{1}{\beta}\mathbb{E}\big[\| a'(g(x)) - \mathbb{E}[\psi(y)|x]\|^2\big],\end{array}$$

where we have used the $(1/\beta)$-strong-convexity of a^*.

Moreover, like in the previous section, we can express the excess risk as:

$$\begin{array}{rcl}\mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}^* & = & \mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}(\mathrm{dec}\circ g^*) \\ & = & \mathbb{E}\Big[\mathbb{E}\big[\ell(y, \mathrm{dec}\circ g(x)) - \ell(y, \mathrm{dec}\circ g^*(x))\big|x\big]\Big] \\ & = & \mathbb{E}\Big[\mathbb{E}\big[\big\langle\psi(y), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big|x\big]\Big] \\ & = & \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big], \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big] \\ & = & \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - a'(g(x)), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big] \\ & & \quad + \mathbb{E}\Big[\big\langle a'(g(x)), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big].\end{array}$$

By definition of the decoding step, we get:

$$\begin{array}{rcl}\mathcal{R}(\mathrm{dec}\circ g) - \mathcal{R}^* & \leqslant & \mathbb{E}\Big[\big\langle\mathbb{E}\big[\psi(y)|x\big] - a'(g(x)), \chi(\mathrm{dec}\circ g(x)) - \chi(\mathrm{dec}\circ g^*(x))\big\rangle\big]\Big] \\ & \leqslant & 2\sup_{z\in\mathcal{Y}}\|\chi(z)\| \cdot \mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - a'(g(x))\big\|\Big] \\ & \leqslant & 2\sup_{z\in\mathcal{Y}}\|\chi(z)\| \cdot \sqrt{\cdot\mathbb{E}\Big[\big\|\mathbb{E}\big[\psi(y)|x\big] - g(x)\big\|^2\Big]} = 2\sqrt{\beta}R_\ell \cdot \sqrt{\mathcal{R}_S(g) - \mathcal{R}_S^*},\end{array}$$

We thus have the same calibration function as for the quadratic surrogate but with an extra factor of $\sqrt{\beta}$. For example, this applies to softmax regression.

Comparison with quadratic surrogates. The comparison between quadratic and smooth surrogates for structured prediction mimics the one for binary classification from Section 4.1. While both lead to consistent predictions and similar calibration functions, they differ in their Bayes predictors, with typically smooth surrogates leading to more natural assumptions (see, e.g., Section 13.7.2).

13.5 Max-margin formulations

Rather than extending the square or logistic loss from binary classification to structured prediction, we can also extend the hinge loss, leading to "max-margin formulations" in reference to the geometric interpretation from Section 4.1.2. In this section, we assume that for any $y \in \mathcal{Y}$, $z \mapsto \ell(z, y)$ is minimized at y, that is, the loss provides a measure of dissimilarity with y.

13.5.1 Structured SVM

Following Taskar et al. (2005); Tsochantaridis et al. (2005), we consider a traditional extension of the support vector machine with a simple interpretation.

We consider a score function, which is a function of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, with the decoder $\arg\max_{z \in \mathcal{Y}} h(x, z)$. The score $S(y, h(x, \cdot))$ is defined as the minimal $\xi \in \mathbb{R}_+$ such that for all $z \in \mathcal{Y}$,

$$h(x, y) \geqslant h(x, z) + \ell(z, y) - \ell(y, y) - \xi.$$

The intuition behind this definition is that we aim at making $h(x, y)$ larger for the observed y than for the other $h(x, z)$, with a difference that is stronger when y and z are further apart, as measured by the loss.

If we take the particular form $h(x, z) = \langle \psi(z), g(x) \rangle$ for $g : \mathcal{X} \to \mathcal{H}$, then the constraint becomes

$$\langle \psi(y), g(x) \rangle \geqslant \langle \psi(z), g(x) \rangle + \langle \chi(y), \psi(z) \rangle - \langle \chi(y), \psi(y) \rangle - \xi,$$

which is equivalent to

$$\xi \geqslant \langle \psi(z) - \psi(y), \chi(y) + g(x) \rangle,$$

and thus the score function is:

$$S(y, g(x)) = \max_{z \in \mathcal{Y}} \langle \psi(z) - \psi(y), \chi(y) + g(x) \rangle. \tag{13.8}$$

For binary classification with the 0-1 loss, this recovers exactly the SVM. Moreover, this convex loss is computable as soon as we can maximize linear functions of $\chi(z)$; thus, this applies to many combinatorial problems, in particular those described earlier.

However, this approach is not consistent; that is, even in the population case where the test distribution is known, it does not lead to the optimal predictor in general; note that there are subcases, such as multi-category classification with the 0-1 loss and a "majority class", where the approach is consistent (Liu, 2007) (see exercise below).

Exercise 13.4 *(♦) For the multi-category classification with the 0–1 loss, show that the structural SVM is Fisher-consistent if for all* $x \in \mathcal{X}$, $\max_{j \in \{1, \dots, k\}} \mathbb{P}(y = j | x) > \frac{1}{2}$.

13.5.2 Max-min formulations (♦♦)

Following Fathony et al. (2016); Nowak-Vila et al. (2020), we can provide a non-smooth surrogate, which is both consistent and comes with a calibration function that does not have a square root. In the binary case, the support vector machine led to a target surrogate function, which was exactly the Bayes predictor, in values in $\{-1, 1\}$. We will see that it is possible to reproduce in the general case. We still assume that for any $y \in \mathcal{Y}$, $z \mapsto \ell(z, y)$ is minimized at y, that is, the loss provides a measure of dissimilarity with y.

Given the expression of the Bayes predictor in Eq. (13.5), that is,

$$f^*(x) \ \in \ \arg\min_{z \in \mathcal{Y}} \big\langle \chi(z), \mathbb{E}[\psi(y)|x] \big\rangle,$$

we can define the function $g^*(x) = \chi(f^*(x)) \in \mathcal{H}$, which is defined as

$$g^*(x) \in \arg \min_{h \in \chi(\mathcal{Y})} \big\langle h, \mathbb{E}[\psi(y)|x] \big\rangle,$$

which happens to be a subgradient at $\mathbb{E}[\psi(y)|x]$ of the function

$$b : \mu \mapsto - \min_{h \in \chi(\mathcal{Y})} \langle h, \mu \rangle = - \min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle.$$

Thus, if we can design a surrogate function S so that $\mathbb{E}[S(y, g(x))|x]$ has minimizer $g^*(x)$ defined above, we can consider the decoder function with our desired consistent behavior:

$$\text{dec} \circ g(x) \in \arg \min_{y' \in \mathcal{Y}} \ \psi(y')^\top g(x),$$

for which

$$\begin{aligned} \text{dec} \circ g^*(x) \quad &\in \quad \arg \min_{y' \in \mathcal{Y}} \ \psi(y')^\top g^*(x) = \arg \min_{y' \in \mathcal{Y}} \ \psi(y')^\top \chi(f^*(x)) \\ &= \quad \arg \min_{y' \in \mathcal{Y}} \ell(y', f^*(x)) = f^*(x), \end{aligned}$$

because of our assumption on the loss $\ell(y, z)$ being minimizer with respect to y at z.

To have a subgradient $g^*(x)$ of b at $\mathbb{E}[\psi(y)|x]$, to be a minimizer of $\mathbb{E}[S(y, g(x))|x]$, it is sufficient to consider (this is not the only choice):

$$S(y, g(x)) = b^*(g(x)) - \langle g(x), \psi(y) \rangle,$$

where b^* is the Fenchel conjugate of b restricted to $\mathcal{M}(\psi) \subset \mathcal{H}$ the closure of the convex hull of all $\psi(z), z \in \mathcal{Y}$, that is,

$$b^*(h) = \max_{\mu \in \mathcal{M}(\psi)} \langle \mu, h \rangle - b(\mu) = \max_{\mu \in \mathcal{M}(\psi)} \langle \mu, h \rangle + \min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle.$$

We thus have:

$$\begin{aligned} S(y, g(x)) \quad &= \quad \max_{\mu \in \mathcal{M}(\psi)} \ \langle g(x), \mu \rangle + \min_{y' \in \mathcal{Y}} \langle \chi(y'), \mu \rangle - \langle g(x), \psi(y) \rangle \qquad (13.9) \\ &= \quad \max_{\mu \in \mathcal{M}(\psi)} \min_{y' \in \mathcal{Y}} \ \langle g(x) + \chi(y'), \mu - \psi(y) \rangle + \ell(y, y'). \end{aligned}$$

Note the similarity with the maximum-margin SVM loss in Eq. (13.8), which considers $y' = y$ instead of the minimization with respect to $y' \in \mathcal{Y}$. This extra minimization makes the surrogate loss function more complicated to minimize (though it is still convex) but leads to a Fisher-consistent estimator.

Fisher consistency. We now prove that any minimizer g^* of $\mathbb{E}\big[S(y, g(x))\big]$ over all measurable functions from $\mathcal{X}$ to $\mathcal{H}$ leads to the optimal prediction, with

$$\text{dec} \circ g^*(x) = \arg \max_{y' \in \mathcal{Y}} \ \psi(y')^\top g^*(x) = f^*(x).$$

As in Section 13.4.4, for $x \in \mathcal{X}$, any minimizer g^* has a value $g^*(x)$ that minimizes

$$\mathbb{E}\big[S(y, g(x))|x\big] = a(g(x)) - \big\langle g(x), \mathbb{E}[\psi(y)|x]\big\rangle.$$

By definition of a, $g^*(x)$ is a minimizer of $h \mapsto \langle h, \mathbb{E}[\psi(y)|x]\rangle$ over $h \in \mathcal{M}(\chi)$. Thus, given the expression of the Bayes predictor in Eq. (13.5), we get $g^*(x) = \chi(f^*(x)) \in \mathcal{H}$. This leads to $\operatorname{dec} \circ g^*(x) = f^*(x)$ because of the assumption that $z \mapsto \ell(z, y)$ is minimized at y. We can also get a linear calibration function in generic situations; see Nowak-Vila et al. (2020) for details.

Optimization algorithms. In this book, we have focused on optimization methods based on subgradients. For the loss defined in Eq. (13.9), this requires to have an optimizer $\mu \in \mathcal{M}(\psi)$, which requires solving a min-max problem in general. Below, we consider the multi-category classification problem with 0-1 loss, where this can be achieved, and refer to Nowak-Vila et al. (2020) for algorithms based on primal-dual formulations.

Binary classification with the 0-1 loss. In this situation, we can compute $a^*(\mu) = \frac{1}{2}|\mu|$ with domain $[-1, 1]$, leading to $a(v) = (|v| - 1/2)_+$, and a formulation that is close to the binary SVM (but non-identical).

Multi-category classification. for $\mathcal{Y} = \{1 \dots, k\}$ and the 0-1 loss, a short exercise shows that we obtain

$$S(y, g(x)) = \max_{A \subset \{1,\dots,k\},\ A \neq \varnothing} \frac{\sum_{j \in A} g_j(x) + |A| - 1}{|A|},$$

where the maximizers in A can be obtained in closed form (together with the subgradient).

13.6 Generalization bounds (♦)

In this section, we provide generalization bounds for the structured prediction problem with smooth convex surrogates defined in Section 13.4. For simplicity, we will assume a linear model with feature vector $\varphi : \mathcal{X} \to \mathbb{R}^d$, and that the feature vector is flexible enough so that the minimizer of the expected surrogate risk is indeed a linear function of φ, that is, $g^{(\theta)}(x) = \theta^\top \varphi(x)$. Taking care of an approximation error would lead to developments similar to Section 7.5.1.

For real-valued prediction functions and linear models, we looked at two frameworks to obtain generalization bounds: one based on Rademacher complexities and one based on stochastic gradient descent. For multi-category classification in Section 13.1, we considered both and only considered SGD in this section as it leads to better bounds. Moreover, we could use kernel methods when φ is only known through the associated kernel function (using in particular Section 7.4.5), but we stick to explicit feature maps for simplicity. Finally, for least-squares, we could directly extend the ridge regression analysis from Section 7.6, which does not use Rademacher complexities. We instead focus on

Lipschitz-continuous losses, that is, we assume that a has gradients bounded by G in ℓ_2-norm.

We thus assume that there exists $\theta_* \in \mathbb{R}^{d\times k}$ such that $a'(\mathbb{E}[\psi(y)|x)]) = \theta_*^\top \varphi(x)$, and thus $\mathcal{R}_S^* = \mathbb{E}\big[S(y, \theta_*^\top \varphi(x))\big]$. We assume i.i.d. observations $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, and the single pass gradient descent recursion, initialized at 0 and defined for $t \in \{1, \dots, n\}$, as

$$\theta_t = \theta_{t-1} - \gamma_t \varphi(x_t) S'(y_t, \theta_{t-1}^\top \varphi(x_t))^\top M,$$

for a positive definite matrix $M^{1/2}$, which will allow to include some penalty function of the form $\mathrm{tr}[\theta\theta^\top M^{-1}]$.

The analysis of Section 5.4 exactly applies, and with the choice $\gamma_t = \gamma$, we obtain the following generalization bound:

$$\mathbb{E}\big[\mathcal{R}_S(g_{(\theta_n)})\big] - \mathcal{R}_S(g^{(\theta_*)}) \leqslant \frac{1}{2\gamma n} \mathrm{tr}[\theta_*^\top M^{-1} \theta_*] + \frac{\gamma G^2 R^2}{2}.$$

With the optimal choice of γ, we get:

$$\mathbb{E}\big[\mathcal{R}_S(g_{(\theta_n)})\big] - \mathcal{R}_S(g^{(\theta_*)}) \leqslant \frac{GR\big(\mathrm{tr}[\theta_*^\top M^{-1} \theta_*]\big)^{1/2}}{\sqrt{n}}.$$

We can then use the calibration result to obtain a consistency result for the structured prediction problem with a bound in $n^{-1/4}$.

Examples of structured regularization. The prediction function is characterized by a matrix $\theta \in \mathbb{R}^{d\times k}$, and without further knowledge, it is natural to use the Frobenius norm or the nuclear norm as regularization or constraint. In particular, set-ups where the k columns have a specific structure, some particular squared norm $\mathrm{tr}[\theta\theta^\top M^{-1}]$ can be natural.

For example, in the ranking problem with the pairwise disagreement loss, where $\psi(y)$ is indexed by two indices $i, j \in \{1, \dots, m\}$, it is natural to consider $\theta_{ij} = \eta_i - \eta_j$ for a matrix $\eta \in \mathbb{R}^{d\times m}$ (see Section 13.7.2 for more details).

13.7 Experiments

In this section, we present two experiments highlighting the benefits of structured prediction and illustrating the results from this chapter.

13.7.1 Robust regression

We consider a toy robust regression problem to illustrate the use of the quadratic surrogates presented in Section 13.4.1. We use a simple one-dimensional robust regression problem, where we compare the square loss and the loss $\ell(y, z) = 1 - \exp(-(y - z)^2)$. We

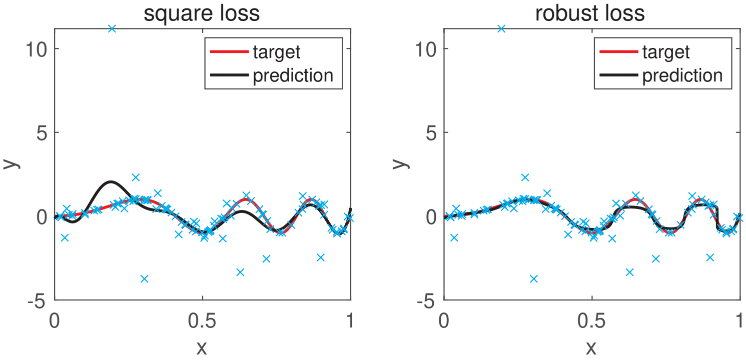

Figure 13.1: Robust regression in one dimension, with heavy-tail noise (fifth power of Gaussian noise): regular square loss (left), vs. robust loss (right).

generate data with heavy-tail additive noise and plot below the best performance for kernel ridge regression with the Gaussian kernel, with the optimal regularization parameter (selected for test performance).

Since we use a kernel method, we can use Section 13.4.3, that is, once the n data-dependent weight functions $w_1(x), \dots, w_n(x)$ are estimated using ridge regression, we need at test time to compute $\arg\min_{z\in\mathbb{R}} \sum_{i=1}^n w_i(x)\ell(y_i,z) = \arg\min_{z\in\mathbb{R}} \sum_{i=1}^n w_i(x)(1-\exp(-(y_i-z)^2))$, which can be done by grid search. See results in Figure 13.1, where we see that the robust loss is indeed more robust to outliers.

13.7.2 Ranking

We illustrate structured prediction on a ranking problem, where $\mathcal{Y}$ is the set of permutations from $\{1,\dots,m\}$ to $\{1,\dots,m\}$. We consider two different loss functions:

- **Square loss:** $\ell(y,x) = \sum_{i=1}^m (y(i) - z(i))^2$, with $\psi(y) = y \in \{1,\dots,m\}^m$. For this loss, we only consider the square loss. We thus need to fit a function $h : \mathcal{X} \to \mathbb{R}^m$ using least-squares directly on y. The decoding step a for test point x is then simply to sort the m components of $h(x)$.

- **Pairwise disagreement:** $\ell(y,z) = \sum_{i,j=1}^m \big(1_{y(i)<y(j)} - 1_{z(i)<z(j)}\big)^2$, with the feature $\psi(y) \in \{-1,1\}^{m(m-1)/2}$ defined as $\psi(y)_{ij} = 1_{y(i)<y(j)}$ for $i<j$. We thus need to learn a function $g : \mathcal{X} \to \mathbb{R}^{m(m-1)/2}$; we consider the square loss, where we minimize expectations of $\|\psi(y) - h(x)\|_2^2$, as well as the logistic loss, where we minimize the expectation of $\sum_{i<j} \log\big(1+\exp(-\psi(y)_{ij}h_{ij}(x))\big)$.

  In terms of decoding, given the function h, at test point x, we need to maximize $\sum_{i<j} g_{ij}(x)1_{z(i)<z(j)}$ with respect to a permutation z when using the square loss, while we need to maximize $\sum_{i<j} \tanh(g_{ij}(x))1_{z(i)<z(j)}$ when using the logistic loss. This is an instance of the "minimum feedback arc set problem", which is an NP-hard problem with known approximation algorithms (Ailon et al., 2008). When the weights $g_{ij}(x)$ are of the form $u(h_j(x) - h_i(x))$ for a non-decreasing function u and

a function $g : \mathcal{X} \to \mathbb{R}^m$, then it can be solved by sorting h. Thus, when either the square loss or the logistic loss is used, using a specific model $g_{ij} = h_i - h_j$ leads to simpler decoding. For the square loss, using this specific model leads exactly to performing least-squares on $y \in \mathbb{R}^m$.

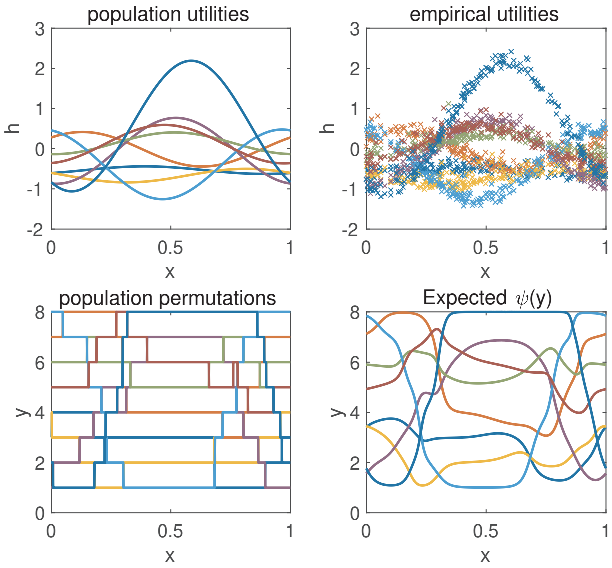

Figure 13.2: Top Left: utilities $h_1, \ldots, h_m$, top right: empirical utilities $h_1 + \eta_1, \ldots, h_m + \eta_m$. Bottom: population permutations $y^*(x) \in \{1, \ldots, m\}$ (left) and conditional expectation $\mathbb{E}[y|x] \in [1, m]$ (right).

Placket-Luce model. We generate data from the Plackett-Luce model (Marden, 1996), that is, from m functions $h_1(x), \ldots, h_m(x)$, with the random permutation obtained by sorting the m real values $h_1(x) + \eta_1, \ldots, h_m(x) + \eta_m$ in ascending order, where each η_i, $i = 1, \ldots, m$, is a Gumbel random variable. Our convention is that $y(i)$ is the position of item i, that is, $y(i) = 1$ if $h_i(x) + \eta_i$ is the lowest, and $y(i) = m$ is $h_i(x) + \eta_i$ is the largest.

If $\pi_i(x) = \text{softmax}(h(x))$, this happens to be equivalent to the model where $y(m)$ is selected with probability vector $\pi(x)$, and then $y(m-1)$ with probability vector proportional to $\pi(x)$ (but without the possibility of taking $y(1)$. In other words, the probability of selecting a permutation z is equal to:

$$\pi(x)_{z(m)} \frac{\pi(x)_{z(m-1)}}{1 - \pi(x)_{z(m)}} \frac{\pi(x)_{z(m-2)}}{1 - \pi(x)_{z(m)} - \pi(x)_{z(m-1)}} \cdots \frac{\pi_{z(2)}}{\pi_{z(1)} + \pi(z(2))}.$$

Moreover, we also have $\mathbb{E}[1_{y(i)<y(j)}|x] = \frac{\pi_j(x)}{\pi_j(x)+\pi_i(x)} = \frac{\exp(h_j(x))}{\exp(h_i(x))+\exp(h_j(x))}$, so that a logistic regression model for predicting $1_{y(i)<y(j)}$ has a target function equal to $h_j - h_i$.

However, the target function for the square loss, $\mathbb{E}[y|x]$, can be expressed as products of softmax functions of subsets of $h_1, \dots, h_m$, and thus are not linear in these functions.

We consider $\mathcal{X} = [0,1]$ and consider functions $h_1, \dots, h_m$ which are linear combinations of cosine functions $\cos(2\pi kx)$ and sine function $\cos(2\pi kx)$ for $k \in \{0,1\}$ for the generating functions. See Figure 13.2 for an illustration.

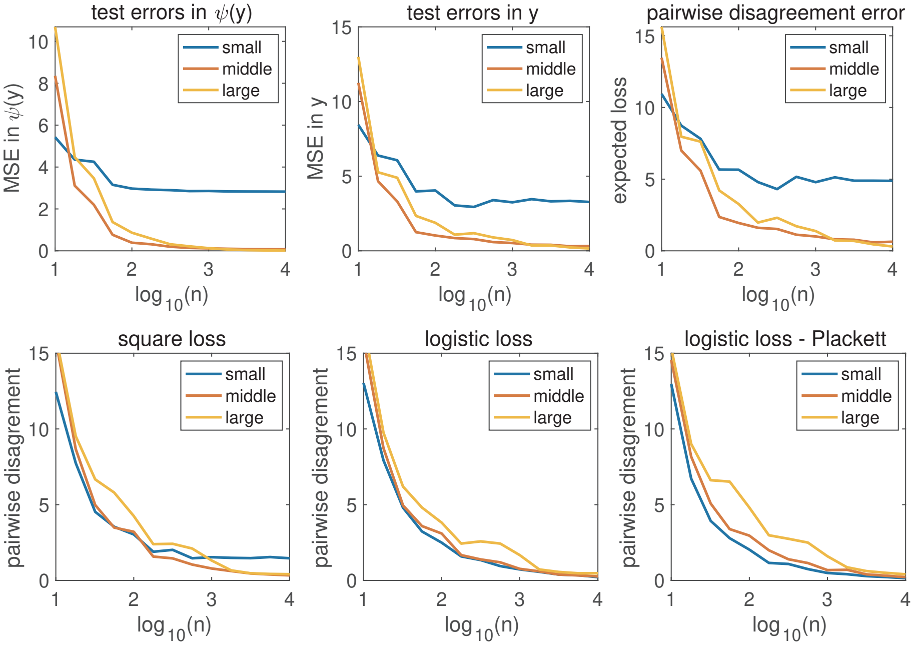

Figure 13.3: Top: using the square loss $\ell(y,z) = \|y - z\|_2^2$ with the square surrogate. Bottom: using the pairwise disagreement loss $\ell(y,z) = \|\psi(y) - \psi(z)\|_2^2$, with the square surrogate, and logistic surrogate.

We consider situations where the prediction model we use for the functions g and h includes the ones generating the data, hence a well-specified model for the logistic loss, but not for the square loss. In Figure 13.3, we provide learning curves where we vary the number n of observations, for three classes of prediction functions based on sines and cosines: for the small model, we use exactly the same model class as for $h_1, \dots, h_m$, that is, $k \in \{0,1\}$, while for the middle model, $k \in \{0, \dots, 3\}$, and for the large model, $k \in \{0, \dots, 15\}$. We use a fixed regularization parameter proportional to $1/n$.

In the top-left plot of Figure 13.3, the quadratic surrogate with $\psi(y) = y$ is considered, and the small model is not well-specified. Thus, when n grows, the testing error does not go down to zero, similarly for the bottom-left plot where the square loss is used with the pairwise disagreement loss function. For the logistic loss, however, where even the small model is well-specified, we obtain a learning curve that goes closer to zero.

13.8 Conclusion

In this chapter, we explored surrogate frameworks beyond binary classification, focusing on convex surrogates. These convex formulations can be used with any prediction functions (linear in the parameter, such as kernel methods, or not, such as neural networks) and come with guarantees for linear models. We presented several principles, e.g., margin-based techniques or frameworks with probabilistic interpretation through maximum likelihood.