Sobolev space

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

Definition

A supplier of a high-risk artificial intelligence system (high-risk AI system), such as a classifier that screens medical images, must be able to state how much a prediction can change when the feature vector of a data point is perturbed slightly. Sobolev spaces make such a statement precise: they consist of the functions whose derivatives, in a generalized sense, have bounded size (Evans, 2010, Ch. 5).

The generalization is needed because the functions of interest are typically not differentiable everywhere. A function g_j is a weak partial derivative of $f : \Omega \to \mathbb{R}$, on an open set $\Omega \subseteq \mathbb{R}^d$, if

$$\int_{\Omega} f(\mathbf{x}) \frac{\partial \varphi(\mathbf{x})}{\partial x_j} \, \mathrm{d}\mathbf{x} = - \int_{\Omega} g_j(\mathbf{x}) \varphi(\mathbf{x}) \, \mathrm{d}\mathbf{x}$$

holds for every infinitely differentiable φ that vanishes outside a bounded subset of Ω (Evans, 2010, Sect. 5.2). The identity is integration by parts, with all derivatives moved onto φ. Both sides are Lebesgue integrals, so f and g_j need only be integrable on bounded subsets of Ω. For an integer $k \geq 1$ and $p \in [1, \infty]$, the Sobolev space $W^{k,p}(\Omega)$ consists of the functions whose weak derivatives up to order k exist and have finite $\|\cdot\|_p$; the case $p = 2$ gives the Hilbert space $H^k(\Omega) = W^{k,2}(\Omega)$.

The case that matters most for machine learning (ML) is $W^{1,\infty}$, the functions with a bounded weak gradient. A function f belongs to it, with

$$\sup_{\mathbf{x} \in \Omega} \|\nabla f(\mathbf{x})\|_2 \leq L,$$

precisely when it is Lipschitz continuous with constant L (Evans, 2010, Sect. 5.8). The bound therefore reads as a robustness guarantee: perturbing a feature vector by $\boldsymbol{\delta}$ changes the value of f by at most $L\|\boldsymbol{\delta}\|_2$. Fig. 1 contrasts a function with a bounded weak gradient and one without.

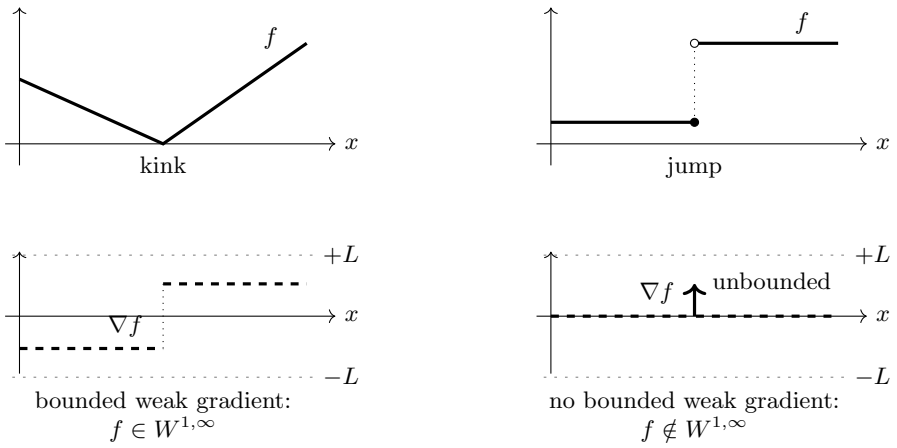

Figure 1: Two functions on $\mathbb{R}$ and their weak derivatives (dashed). Left: a kinked function is not differentiable at the kink, yet its weak derivative exists and stays within $\pm L$, so the function lies in $W^{1,\infty}$ and is Lipschitz continuous with constant L. Right: a function with a jump has no weak derivative that is a function (filled and open circles mark the two one-sided values); no bound L exists, and an arbitrarily small perturbation of x changes the value by the height of the jump

The bound is what turns robustness into a number that can be computed and declared. Consider a binary classification hypothesis $\hat{h}(\mathbf{x}) = \operatorname{sign}(f(\mathbf{x}))$ with $f \in W^{1,\infty}$ and constant L. If $|f(\mathbf{x})| = \gamma$, then every perturbation $\boldsymbol{\delta}$ with $\|\boldsymbol{\delta}\|_2 < \gamma/L$ leaves $\operatorname{sign}(f(\mathbf{x}+\boldsymbol{\delta}))$ unchanged, since the value of f moves by less than γ. The radius γ/L is a certified robustness guarantee against an adversarial attack (Tsuzuku et al., 2018). For a linear classifier $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$ the weak gradient is the constant $\mathbf{w}$, and γ/L is the distance of the feature vector from the decision boundary (see support vector machine (SVM)).

An artificial neural network (ANN) with rectified linear unit (ReLU) activation functions is the case that makes the weak derivative indispensable: such a hypothesis is piecewise linear, so its classical gradient does not exist along the kinks, while its weak gradient exists and is bounded by the product of the largest singular values of the weight matrices. For a two-layer ReLU ANN on $\mathbb{R}^2$, that product gives $L \leq 4.94$ while the smallest valid constant is 2.12: the bound holds but is loose. Drawing 60000 perturbations inside the certified radius γ/L produced no change of the prediction (`pythondemos/sobolevspace.py`). By contrast, a decision tree is piecewise constant with jumps and has no bounded weak gradient at all: in the same experiment, a stump changes its prediction across a perturbation of Euclidean norm $2 \cdot 10^{-9}$, so no certified radius exists for it, however large the margin.

A certified radius is also of regulatory interest. Article 15 of the EU AI Act requires a high-risk AI system to reach an appropriate level of robustness and to perform consistently, to be as resilient as possible to errors and inconsistencies,

and to have its accuracy levels and metrics declared in the instructions for use (European Parliament and Council of the European Union, 2024). A certified radius is such a metric: it states, as a single number, how large a perturbation of a feature vector the prediction provably survives.

Sobolev spaces also underlie two regularizers used in ML. Penalizing the squared H^1 seminorm

$$\int_\Omega \|\nabla f(\mathbf{x})\|_2^2 \, \mathrm{d}\mathbf{x}$$

expresses the smoothness assumption; its counterpart for a function defined on the nodes of a graph replaces the gradient by differences across edges, which is the form used by generalized total variation (GTV) and generalized total variation minimization (GTVMin) (SarcheshmehPour et al., 2023). Penalizing the gradient in L^1 instead gives total variation regularization, which allows jumps: as a transition of width ε sharpens, its squared H^1 seminorm grows without bound, while its total variation stays at the height of the limiting jump (`pythondemos/sobolevspace.py`). This is why total variation regularization is used in image denoising (Rudin et al., 1992). Kernels connect to Sobolev spaces as well. For a Matérn kernel with smoothness parameter $\nu > 0$ on a domain $\Omega \subseteq \mathbb{R}^d$ with Lipschitz boundary, and for $s := \nu + d/2$ an integer, the reproducing kernel Hilbert space (RKHS) of the kernel is $W^{s,2}(\Omega)$ as a set of functions, and the two norms are equivalent: there are $c_1, c_2 > 0$ with $c_1\|f\|_{W^{s,2}(\Omega)} \leq \|f\|_{\mathcal{H}_k} \leq c_2\|f\|_{W^{s,2}(\Omega)}$ for every f in the RKHS (Rasmussen and Williams, 2006, Eq. 4.15; Wendland, 2005, Cor. 10.48; Kanagawa et al., 2018, Example 2.6). The penalty term of the corresponding regularized empirical risk minimization (RERM) is therefore a Sobolev norm, and the smoothness it enforces is again weak: the functions of this RKHS have weak derivatives up to order s, while they are guaranteed to be differentiable in the classical sense only up to order ν (Kanagawa et al., 2018, Remark 2.11).

Cross References

Lipschitz, robustness, smoothness assumption, GTV, GTVMin, RKHS, Hilbert space, differentiable, Lebesgue integral

References

1. SarcheshmehPour et al. (2023). Clustered Federated Learning via Generalized Total Variation Minimization. IEEE Trans. Signal Process., 71, pp. 4240–4256.

2. European Parliament and Council of the European Union (2024). Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intel-

ligence Act) (Text with EEA relevance). Official Journal of the European Union, L series, Jul. 12. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

3. Rasmussen and Williams (2006). Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA, USA.

4. Evans (2010). Partial Differential Equations. 2nd ed., American Mathematical Society, Providence, RI, USA.

5. Kanagawa et al. (2018). Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences. URL: https://arxiv.org/abs/1807.02582.

6. Rudin et al. (1992). Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1–4), pp. 259–268.

7. Tsuzuku et al. (2018). Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks. In: Adv. Neural Inf. Process. Syst. (NeurIPS), pp. 6541–6550.

8. Wendland (2005). Scattered Data Approximation. Cambridge University Press, Cambridge, UK.