In machine learning , Manifold regularization is a technique for using the shape of a dataset to constrain the functions that should be learned on that dataset. In many machine learning problems, the data to be learned do not cover the entire input space. For example, a facial recognition system may not need to classify any possible image, but only the subset of images that contain faces. The technique of manifold learning assumes that the relevant subset of data comes from a manifold , a mathematical structure with useful properties. The technique also assumes that the function to be learned is smooth : data with different labels are not likely to be close together, and so the labeling function should not change quickly in areas where there are likely to be many data points. Because of this assumption, a manifold regularization algorithm can use unlabeled data to inform where the learned function is allowed to change quickly and where it is not, using an extension of the technique of Tikhonov regularization . Manifold regularization algorithms can extend supervised learning algorithms in semi-supervised learning and transductive learning settings, where unlabeled data are available. The technique has been used for applications including medical imaging, geographical imaging, and object recognition.

Manifold regularizer

Motivation

Manifold regularization is a type of regularization , a family of techniques that reduces overfitting and ensures that a problem is well-posed by penalizing complex solutions. In particular, manifold regularization extends the technique of Tikhonov regularization as applied to Reproducing kernel Hilbert spaces (RKHSs). Under standard Tikhonov regularization on RKHSs, a learning algorithm attempts to learn a function f from among a hypothesis space of functions $\mathcal{H}$. The hypothesis space is an RKHS, meaning that it is associated with a kernel K , and so every candidate function f has a norm $\left\|f\right\|_{K}$, which represents the complexity of the candidate function in the hypothesis space. When the algorithm considers a candidate function, it takes its norm into account in order to penalize complex functions.

Formally, given a set of labeled training data $(x_{1},y_{1}),\ldots ,(x_{\ell },y_{\ell })$ with $x_{i}\in X, y_{i}\in Y$ and a loss function V , a learning algorithm using Tikhonov regularization will attempt to solve the expression

where γ is a hyperparameter that controls how much the algorithm will prefer simpler functions over functions that fit the data better.

Manifold regularization adds a second regularization term, the intrinsic regularizer , to the ambient regularizer used in standard Tikhonov regularization. Under the manifold assumption in machine learning, the data in question do not come from the entire input space X , but instead from a nonlinear manifold $M\subset X$. The geometry of this manifold, the intrinsic space, is used to determine the regularization norm. [1]

Laplacian norm

There are many possible choices for the intrinsic regularizer $\left\|f\right\|_{I}$. Many natural choices involve the gradient on the manifold $\nabla _{M}$, which can provide a measure of how smooth a target function is. A smooth function should change slowly where the input data are dense; that is, the gradient $\nabla _{M}f(x)$ should be small where the marginal probability density $\mathcal{P}_{X}(x)$, the

probability density of a randomly drawn data point appearing at x, is large. This gives one appropriate choice for the intrinsic regularizer:

In practice, this norm cannot be computed directly because the marginal distribution $\mathcal{P}_X$ is unknown, but it can be estimated from the provided data.

Graph-based approach of the Laplacian norm

When the distances between input points are interpreted as a graph, then the Laplacian matrix of the graph can help to estimate the marginal distribution. Suppose that the input data include ℓ labeled examples (pairs of an input x and a label y) and u unlabeled examples (inputs without associated labels). Define W to be a matrix of edge weights for a graph, where W_{ij} is a distance measure between the data points x_i and x_j. Define D to be a diagonal matrix with $D_{ii}=\sum_{j=1}^{\ell+u}W_{ij}$ and L to be the Laplacian matrix $D-W$. Then, as the number of data points $\ell+u$ increases, L converges to the Laplace–Beltrami operator Δ_{M}, which is the divergence of the gradient ∇_{M}. [2][3] Then, if $\mathbf{f}$ is a vector of the values of f at the data, $\mathbf{f}=[f(x_{1}),\ldots,f(x_{\ell+u})]^{\mathrm{T}}$, the intrinsic norm can be estimated:

As the number of data points $\ell+u$ increases, this empirical definition of $\left\|f\right\|_{I}^{2}$ converges to the definition when $\mathcal{P}_X$ is known. [1]

Solving the regularization problem with graph-based approach

Using the weights γ_{A} and γ_{I} for the ambient and intrinsic regularizers, the final expression to be solved becomes:

As with other kernel methods , $\mathcal{H}$ may be an infinite-dimensional space, so if the regularization expression cannot be solved explicitly, it is impossible to search the entire space for a solution. Instead, a representer theorem shows that under certain conditions on the choice of the norm $\left\|f\right\|_{I}$, the optimal solution f^{*} must be a linear combination of the kernel centered at each of the input points: for some weights α_{i},

Using this result, it is possible to search for the optimal solution f^{*} by searching the finite-dimensional space defined by the possible choices of α_{i}. [1]

Functional approach of the Laplacian norm

The idea beyond the graph-Laplacian is to use neighbors to estimate the Laplacian.

This method is akin to local averaging methods , that are known to scale poorly in high-dimensional problems.

Indeed, the graph Laplacian is known to suffer from the curse of dimensionality . [2] Luckily, it is possible to leverage expected smoothness of the function to estimate thanks to more advanced functional analysis.

This method consists of estimating the Laplacian operator using derivatives of the kernel reading $\partial_{1,j}K(x_{i},x)$ where $\partial_{1,j}$ denotes the partial derivatives according to the j -th coordinate of the first variable. [4] This second approach to the Laplacian norm is to put in relation with meshfree methods , that contrast with the finite difference method in PDE.

Applications

Manifold regularization can extend a variety of algorithms that can be expressed using Tikhonov regularization, by choosing an appropriate loss function V and hypothesis space $\mathcal{H}$. Two commonly used examples are the families of support vector

machines and regularized least squares algorithms. (Regularized least squares includes the ridge regression algorithm; the related algorithms of LASSO and elastic net regularization can be expressed as support vector machines. [5] [6]) The extended versions of these algorithms are called Laplacian Regularized Least Squares (abbreviated LapRLS) and Laplacian Support Vector Machines (LapSVM), respectively. [1]

Laplacian Regularized Least Squares (LapRLS)

Regularized least squares (RLS) is a family of regression algorithms : algorithms that predict a value $y=f(x)$ {\displaystyle y=f(x)} for its inputs x {\displaystyle x} , with the goal that the predicted values should be close to the true labels for the data. In particular, RLS is designed to minimize the mean squared error between the predicted values and the true labels, subject to regularization. Ridge regression is one form of RLS; in general, RLS is the same as ridge regression combined with the kernel method . [citation needed] The problem statement for RLS results from choosing the loss function V {\displaystyle V} in Tikhonov regularization to be the mean squared error:

Thanks to the representer theorem , the solution can be written as a weighted sum of the kernel evaluated at the data points:

and solving for α^{*} {\displaystyle \alpha ^{*}} gives:

where K {\displaystyle K} is defined to be the kernel matrix, with $K_{ij}=K(x_i,x_j)$ {\displaystyle K_{ij}=K(x_{i},x_{j})} , and Y {\displaystyle Y} is the vector of data labels.

Adding a Laplacian term for manifold regularization gives the Laplacian RLS statement:

The representer theorem for manifold regularization again gives

and this yields an expression for the vector α^{*} {\displaystyle \alpha ^{*}} . Letting K {\displaystyle K} be the kernel matrix as above, Y {\displaystyle Y} be the vector of data labels, and J {\displaystyle J} be the $(\blacksquare +u)\times (\blacksquare +u)$ {\displaystyle (\ell +u)\times (\ell +u)} block matrix $\begin{bmatrix}I_\blacksquare &0\\0&0_u\end{bmatrix}$ {\displaystyle {\begin{bmatrix}I_{\ell }&0\\0&0_{u}\end{bmatrix}}} :

with a solution of

LapRLS has been applied to problems including sensor networks, [7] medical imaging , [8] [9] object detection, [10] spectroscopy , [11] document classification , [12] drug-protein interactions, [13] and compressing images and videos. [14]

Laplacian Support Vector Machines (LapSVM)

Support vector machines (SVMs) are a family of algorithms often used for classifying data into two or more groups, or classes . Intuitively, an SVM draws a boundary between classes so that the closest labeled examples to the boundary are as far away as possible. This can be directly expressed as a linear program , but it is also equivalent to Tikhonov regularization with the hinge loss function, $V(f(x),y)=\max(0,1-yf(x))$ {\displaystyle V(f(x),y)=\max(0,1-yf(x))} :

Adding the intrinsic regularization term to this expression gives the LapSVM problem statement:

Again, the representer theorem allows the solution to be expressed in terms of the kernel evaluated at the data points:

α {\displaystyle \alpha } can be found by writing the problem as a linear program and solving the dual problem . Again letting K {\displaystyle K} be the kernel matrix and J {\displaystyle J} be the block matrix $\begin{bmatrix}I_\blacksquare &0\\0&0_u\end{bmatrix}$ {\displaystyle {\begin{bmatrix}I_{\ell }&0\\0&0_{u}\end{bmatrix}}} , the solution can be shown to be

where β^{*} {\displaystyle \beta ^{*}} is the solution to the dual problem

and Q {\displaystyle Q} is defined by

LapSVM has been applied to problems including geographical imaging, [17] [18] [19] medical imaging, [20] [21] [22] face recognition, [23] machine maintenance, [24] and brain–computer interfaces . [25]

Limitations

Manifold regularization assumes that data with different labels are not likely to be close together. This assumption is what allows the technique to draw information from unlabeled data, but it only applies to some problem domains. Depending on the structure of the data, it may be necessary to use a different semi-supervised or transductive learning algorithm. [26]

In some datasets, the intrinsic norm of a function $\left\|f\right\|_{I}$ can be very close to the ambient norm $\left\|f\right\|_{K}$: for example, if the data consist of two classes that lie on perpendicular lines, the intrinsic norm will be equal to the ambient norm. In this case, unlabeled data have no effect on the solution learned by manifold regularization, even if the data fit the algorithm's assumption that the separator should be smooth. Approaches related to co-training have been proposed to address this limitation. [27]

If there are a very large number of unlabeled examples, the kernel matrix K becomes very large, and a manifold regularization algorithm may become prohibitively slow to compute. Online algorithms and sparse approximations of the manifold may help in this case. [28]

See also

Manifold learning

Manifold hypothesis

Semi-supervised learning

Transduction (machine learning)

Spectral graph theory

Reproducing kernel Hilbert space

Tikhonov regularization

Differential geometry

References

External links

Software

The ManifoldLearn library and the Primal LapSVM library implement LapRLS and LapSVM in MATLAB .

The Dlib library for C++ includes a linear manifold regularization function.