Title: Matrix regularization

URL: https://en.wikipedia.org/wiki/Matrix_regularization

PageID: 44628821

Categories: Category:Estimation theory, Category:Machine learning, Category:Matrices (mathematics)

In the field of statistical learning theory , matrix regularization generalizes notions of vector regularization to cases where the object to be learned is a matrix. The purpose of regularization is to enforce conditions, for example sparsity or smoothness, that can produce stable predictive functions. For example, in the more common vector framework, Tikhonov regularization optimizes over min x ■ A x – y ■ 2 + λ ■ x ■ 2 {\displaystyle \min _{x}\left\|Ax-y\right\|^{2}+\lambda \left\|x\right\|^{2}} to find a vector x {\displaystyle x} that is a stable solution to the regression problem. When the system is described by a matrix rather than a vector, this problem can be written as min X ■ A X – Y ■ 2 + λ ■ X ■ 2 , {\displaystyle \min _{X}\left\|AX-Y\right\|^{2}+\lambda \left\|X\right\|^{2},} where the vector norm enforcing a regularization penalty on x {\displaystyle x} has been extended to a matrix norm on X {\displaystyle X} .

Matrix regularization has applications in matrix completion , multivariate regression , and multi-task learning . Ideas of feature and group selection can also be extended to matrices, and these can be generalized to the nonparametric case of multiple kernel learning .

Basic definition

Consider a matrix W {\displaystyle W} to be learned from a set of examples, S = (X i t , y i t) {\displaystyle S=(X_{i}^{t},y_{i}^{t})} , where i {\displaystyle i} goes from 1 {\displaystyle 1} to n {\displaystyle n} , and t {\displaystyle t} goes from 1 {\displaystyle 1} to T {\displaystyle T} . Let each input matrix X i {\displaystyle X_{i}} be ∈ R D T {\displaystyle \in \mathbb {R} ^{DT}} , and let W {\displaystyle W} be of size D × T {\displaystyle D\times T} . A general model for the output y {\displaystyle y} can be posed as y i t = ■ W , X i t ■ F , {\displaystyle y_{i}^{t}=\left\langle W,X_{i}^{t}\right\rangle _{F},} where the inner product is the Frobenius inner product . For different applications the matrices X i {\displaystyle X_{i}} will have different forms, [1] but for each of these the optimization problem to infer W {\displaystyle W} can be written as min W ∈ H E (W) + R (W) , {\displaystyle \min _{W\in {\mathcal {H}}}E(W)+R(W),} where E {\displaystyle E} defines the empirical error for a given W {\displaystyle W} , and R (W) {\displaystyle R(W)} is a matrix regularization penalty. The function R (W) {\displaystyle R(W)} is typically chosen to be convex and is often selected to enforce sparsity (using ■ 1 {\displaystyle \ell ^{1}} -norms) and/or smoothness (using ■ 2 {\displaystyle \ell ^{2}} -norms). Finally, W {\displaystyle W} is in the space of matrices H {\displaystyle {\mathcal {H}}} with Frobenius inner product ■ … ■ F {\displaystyle \langle \dots \rangle _{F}} .

General applications

Matrix completion

In the problem of matrix completion , the matrix X i t {\displaystyle X_{i}^{t}} takes the form X i t = e t ⊗ e i ′ , {\displaystyle X_{i}^{t}=e_{t}\otimes e_{i}',} where (e t) t {\displaystyle (e_{t})_{t}} and (e i ′) i {\displaystyle (e_{i}')_{i}} are the canonical basis in R T {\displaystyle \mathbb {R} ^{T}} and R D {\displaystyle \mathbb {R} ^{D}} . In this case the role of the Frobenius inner product is to select individual elements w i t {\displaystyle w_{i}^{t}} from the matrix W {\displaystyle W} . Thus, the output y {\displaystyle y} is a sampling of entries from the matrix W {\displaystyle W} .

The problem of reconstructing W {\displaystyle W} from a small set of sampled entries is possible only under certain restrictions on the matrix, and these restrictions can be enforced by a regularization function. For example, it might be assumed that W {\displaystyle W} is low-rank, in which case the regularization penalty can take the form of a nuclear norm. [2] R (W) = λ ■ W ■ ∗

= λ ∑ i | σ i | , {\displaystyle R(W)=\lambda \left\|W\right\|_{*}=\lambda \sum _{i}\left|\sigma _{i}\right|,} where σ i {\displaystyle \sigma _{i}} , with i {\displaystyle i} from 1 {\displaystyle 1} to min D , T {\displaystyle \min D,T} , are the singular values of W {\displaystyle W} .

Multivariate regression

Models used in multivariate regression are parameterized by a matrix of coefficients. In the Frobenius inner product above, each matrix X {\displaystyle X} is X i t = e t ⊗ x i {\displaystyle X_{i}^{t}=e_{t}\otimes x_{i}} such that the output of the inner product is the dot product of one row of the input with one column of the coefficient matrix. The familiar form of such models is Y = X W + b {\displaystyle Y=XW+b}

Many of the vector norms used in single variable regression can be extended to the multivariate case. One example is the squared Frobenius norm, which can be viewed as an ■ 2 {\displaystyle \ell ^{2}} -norm acting either entrywise, or on the singular values of the matrix: R (W) = λ ■ W ■ F 2 = λ ∑ i ∑ j | w i j | 2 = λ Tr ■ (W ∗ W) = λ ∑ i σ i 2 . {\displaystyle R(W)=\lambda \left\|W\right\|_{F}^{2}=\lambda \sum _{i}\sum _{j}\left|w_{ij}\right|^{2}=\lambda \operatorname {Tr} \left(W^{*}W\right)=\lambda \sum _{i}\sigma _{i}^{2}.}

In the multivariate case the effect of regularizing with the Frobenius norm is the same as the vector case; very complex models will have larger norms, and, thus, will be penalized more.

Multi-task learning

The setup for multi-task learning is almost the same as the setup for multivariate regression. The primary difference is that the input variables are also indexed by task (columns of Y {\displaystyle Y}). The representation with the Frobenius inner product is then X i t = e t ⊗ x i t . {\displaystyle X_{i}^{t}=e_{t}\otimes x_{i}^{t}.}

The role of matrix regularization in this setting can be the same as in multivariate regression, but matrix norms can also be used to couple learning problems across tasks. In particular, note that for the optimization problem min W ■ X W − Y ■ 2 2 + λ ■ W ■ 2 2 {\displaystyle \min _{W}\left\|XW-Y\right\|_{2}^{2}+\lambda \left\|W\right\|_{2}^{2}} the solutions corresponding to each column of Y {\displaystyle Y} are decoupled. That is, the same solution can be found by solving the joint problem, or by solving an isolated regression problem for each column. The problems can be coupled by adding an additional regularization penalty on the covariance of solutions min W , Ω ■ X W − Y ■ 2 2 + λ 1 ■ W ■ 2 2 + λ 2 Tr ■ (W T Ω − 1 W) {\displaystyle \min _{W,\Omega }\left\|XW-Y\right\|_{2}^{2}+\lambda _{1}\left\|W\right\|_{2}^{2}+\lambda _{2}\operatorname {Tr} \left(W^{T}\Omega ^{-1}W\right)} where Ω {\displaystyle \Omega } models the relationship between tasks. This scheme can be used to both enforce similarity of solutions across tasks, and to learn the specific structure of task similarity by alternating between optimizations of W {\displaystyle W} and Ω {\displaystyle \Omega } . [3] When the relationship between tasks is known to lie on a graph, the Laplacian matrix of the graph can be used to couple the learning problems.

Spectral regularization

Regularization by spectral filtering has been used to find stable solutions to problems such as those discussed above by addressing ill-posed matrix inversions (see for example Filter function for Tikhonov regularization). In many cases the regularization function acts on the input (or kernel) to ensure a bounded inverse by eliminating small singular values, but it can also be useful to have spectral norms that act on the matrix that is to be learned.

There are a number of matrix norms that act on the singular values of the matrix. Frequently used examples include the Schatten p-norms , with p = 1 or 2. For example, matrix regularization with a Schatten 1-norm, also called the nuclear norm, can be used to enforce sparsity in the spectrum of a matrix. This has been used in the context of matrix completion when the matrix in question is believed to have a restricted rank. [2] In this case the optimization problem becomes: min ■ W ■ ∗ subject to W i , j = Y i j . {\displaystyle \min \left\|W\right\|_{*}~~{\text{ subject to }}~~W_{i,j}=Y_{ij}.}

Spectral Regularization is also used to enforce a reduced rank coefficient matrix in multivariate regression. [4] In this setting, a reduced rank coefficient matrix can be found by keeping just the top n {\displaystyle n} singular values, but this can be extended to keep any reduced set of singular

values and vectors.

Structured sparsity

Sparse optimization has become the focus of much research interest as a way to find solutions that depend on a small number of variables (see e.g. the Lasso method). In principle, entry-wise sparsity can be enforced by penalizing the entry-wise ■ 0 {\displaystyle \ell ^{0}} -norm of the matrix, but the ■ 0 {\displaystyle \ell ^{0}} -norm is not convex. In practice this can be implemented by convex relaxation to the ■ 1 {\displaystyle \ell ^{1}} -norm. While entry-wise regularization with an ■ 1 {\displaystyle \ell ^{1}} -norm will find solutions with a small number of nonzero elements, applying an ■ 1 {\displaystyle \ell ^{1}} -norm to different groups of variables can enforce structure in the sparsity of solutions. [5]

The most straightforward example of structured sparsity uses the ■ p , q {\displaystyle \ell _{p,q}} norm with p = 2 {\displaystyle p=2} and q = 1 {\displaystyle q=1} : ■ W ■ 2 , 1 = ∑ i ■ w i ■ 2 . {\displaystyle \left\|W\right\|_{2,1}=\sum _{i}\left\|w_{i}\right\|_{2}.}

For example, the ■ 2 , 1 {\displaystyle \ell _{2,1}} norm is used in multi-task learning to group features across tasks, such that all the elements in a given row of the coefficient matrix can be forced to zero as a group. [6] The grouping effect is achieved by taking the ■ 2 {\displaystyle \ell ^{2}} -norm of each row, and then taking the total penalty to be the sum of these row-wise norms. This regularization results in rows that will tend to be all zeros, or dense. The same type of regularization can be used to enforce sparsity column-wise by taking the ■ 2 {\displaystyle \ell ^{2}} -norms of each column.

More generally, the ■ 2 , 1 {\displaystyle \ell _{2,1}} norm can be applied to arbitrary groups of variables: R (W) = λ ∑ g G ∑ j | G g | | w g j | 2 = λ ∑ g G ■ w g ■ g {\displaystyle R(W)=\lambda \sum _{g}^{G}{\sqrt {\sum _{j}^{|G_{g}|}\left|w_{g}^{j}\right|^{2}}}=\lambda \sum _{g}^{G}\left\|w_{g}\right\|_{g}} where the index g {\displaystyle g} is across groups of variables, and | G g | {\displaystyle |G_{g}|} indicates the cardinality of group g {\displaystyle g} .

Algorithms for solving these group sparsity problems extend the more well-known Lasso and group Lasso methods by allowing overlapping groups, for example, and have been implemented via matching pursuit : [7] and proximal gradient methods . [8] By writing the proximal gradient with respect to a given coefficient, w g i {\displaystyle w_{g}^{i}} , it can be seen that this norm enforces a group-wise soft threshold [1] prox λ , R g ■ (w g) i = (w g i – λ w g i ■ w g ■ g) 1 ■ w g ■ g ≥ λ . {\displaystyle \operatorname {prox} _{\lambda ,R_{g}}\left(w_{g}\right)^{i}=\left(w_{g}^{i}-\lambda {\frac {w_{g}^{i}}{\left\|w_{g}\right\|_{g}}}\right)\mathbf {1} _{\|w_{g}\|_{g}\geq \lambda }.} where 1 ■ w g ■ g ≥ λ {\displaystyle \mathbf {1} _{\|w_{g}\|_{g}\geq \lambda }} is the indicator function for group norms ≥ λ {\displaystyle \geq \lambda } .

Thus, using ■ 2 , 1 {\displaystyle \ell _{2,1}} norms it is straightforward to enforce structure in the sparsity of a matrix either row-wise, column-wise, or in arbitrary blocks. By enforcing group norms on blocks in multivariate or multi-task regression, for example, it is possible to find groups of input and output variables, such that defined subsets of output variables (columns in the matrix Y {\displaystyle Y}) will depend on the same sparse set of input variables.

Multiple kernel selection

The ideas of structured sparsity and feature selection can be extended to the nonparametric case of multiple kernel learning . [9] This can be useful when there are multiple types of input data (color and texture, for example) with different appropriate kernels for each, or when the appropriate kernel is unknown. If there are two kernels, for example, with feature maps A {\displaystyle A} and B {\displaystyle B} that lie in corresponding reproducing kernel Hilbert spaces H A , H B {\displaystyle {\mathcal {H_{A}}},{\mathcal {H_{B}}}} , then a larger space, H D {\displaystyle {\mathcal {H_{D}}}} , can be created as the sum of two spaces: H D : f = h + h ′ ; h ∈ H A , h ′ ∈ H B {\displaystyle {\mathcal {H_{D}}}:f=h+h';h\in {\mathcal {H_{A}}},h'\in {\mathcal {H_{B}}}} assuming linear independence in A {\displaystyle A} and B {\displaystyle B} . In this case the ■ 2 , 1 {\displaystyle \ell _{2,1}} -norm is again the sum of norms: ■ f ■ H D , 1 = ■ h ■ H A + ■ h ′ ■ H B {\displaystyle \left\|f\right\|_{{\mathcal {H_{D}}},1}=\left\|h\right\|_{\mathcal {H_{A}}}+\left\|h'\right\|_{\mathcal {H_{B}}}}

Thus, by choosing a matrix regularization function as this type of norm, it is possible to find a solution that is sparse in terms of which kernels are used, but dense in the coefficient of each used kernel. Multiple kernel learning can also be used as a form of nonlinear variable selection, or as a model aggregation technique (e.g. by taking the sum of squared norms and relaxing sparsity constraints). For example, each kernel can be taken to be the Gaussian kernel with a different width.

See also

Regularization (mathematics)

References