Title: Multiple kernel learning

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

PageID: 44635680

Categories: Category:Data mining, Category:Machine learning algorithms

---

Supervised learning

Unsupervised learning

Semi-supervised learning

Self-supervised learning

Reinforcement learning

Meta-learning

Online learning

Batch learning

Curriculum learning

Rule-based learning

Neuro-symbolic AI

Neuromorphic engineering

Quantum machine learning

Classification

Generative modeling

Regression

Clustering

Dimensionality reduction

Density estimation

Anomaly detection

Data cleaning

AutoML

Association rules

Semantic analysis

Structured prediction

Feature engineering

Feature learning

Learning to rank

Grammar induction

Ontology learning

Multimodal learning

Apprenticeship learning

Decision trees

Ensembles Bagging Boosting Random forest

Bagging

Boosting

Random forest

k -NN

Linear regression

Naive Bayes

Artificial neural networks

Logistic regression

Perceptron

Relevance vector machine (RVM)

Support vector machine (SVM)

BIRCH

CURE

Hierarchical

k -means

Fuzzy

Expectation–maximization (EM)

DBSCAN

OPTICS

Mean shift

Factor analysis

CCA

ICA

LDA

NMF

PCA

PGD

t-SNE

SDL

Graphical models Bayes net Conditional random field Hidden Markov

Bayes net

Conditional random field

Hidden Markov

RANSAC

k -NN

Local outlier factor

Isolation forest

Autoencoder

Deep learning

Feedforward neural network

Recurrent neural network LSTM GRU ESN reservoir computing

LSTM

GRU

ESN

reservoir computing

Boltzmann machine Restricted

Restricted

GAN

Diffusion model

SOM

Convolutional neural network U-Net LeNet AlexNet DeepDream

U-Net

LeNet

AlexNet

DeepDream

Neural field Neural radiance field Physics-informed neural networks

Neural radiance field

Physics-informed neural networks

Transformer Vision

Vision

Mamba

Spiking neural network

Memtransistor

Electrochemical RAM (ECRAM)

Q-learning

Policy gradient

SARSA

Temporal difference (TD)

Multi-agent Self-play

Self-play

Active learning

Crowdsourcing

Human-in-the-loop

Mechanistic interpretability

RLHF

Coefficient of determination

Confusion matrix

Learning curve

ROC curve

Kernel machines

Bias–variance tradeoff

Computational learning theory

Empirical risk minimization

Occam learning

PAC learning

Statistical learning

VC theory

Topological deep learning

AAAI

ECML PKDD

NeurIPS

ICML

ICLR

IJCAI

ML

JMLR

Glossary of artificial intelligence

List of datasets for machine-learning research List of datasets in computer vision and image processing

List of datasets in computer vision and image processing

Outline of machine learning

v

t

e

Multiple kernel learning refers to a set of machine learning methods that use a predefined set of kernels and learn an optimal linear or non-linear combination of kernels as part of the algorithm. Reasons to use multiple kernel learning include a) the ability to select for an optimal kernel and parameters from a larger set of kernels, reducing bias due to kernel selection while allowing for more automated machine learning methods, and b) combining data from different sources (e.g. sound and images from a video) that have different notions of similarity and thus require different kernels. Instead of creating a new kernel, multiple kernel algorithms can be used to combine kernels already established for each individual data source.

Multiple kernel learning approaches have been used in many applications, such as event recognition in video, object recognition in images, and biomedical data fusion.

Algorithms

Multiple kernel learning algorithms have been developed for supervised, semi-supervised, as well as unsupervised learning. Most work has been done on the supervised learning case with linear combinations of kernels, however, many algorithms have been developed. The basic idea behind multiple kernel learning algorithms is to add an extra parameter to the minimization problem of the learning algorithm. As an example, consider the case of supervised learning of a linear combination of a set of n {\displaystyle n} kernels K {\displaystyle K} . We introduce a new kernel K ′ = ∑ i = 1 n β i K i {\displaystyle K'=\sum _{i=1}^{n}\beta _{i}K_{i}} , where β {\displaystyle \beta } is a vector of coefficients for each kernel. Because the kernels are additive (due to properties of reproducing kernel Hilbert spaces), this new function is still a kernel. For a set of data X {\displaystyle X} with labels Y {\displaystyle Y} , the minimization problem can then be written as

where E {\displaystyle \mathrm {E} } is an error function and R {\displaystyle R} is a regularization term. E {\displaystyle \mathrm {E} } is typically the square loss function (Tikhonov regularization) or the hinge loss function (for SVM algorithms), and R {\displaystyle R} is usually an ■ n {\displaystyle \ell _{n}} norm or some combination of the norms (i.e. elastic net regularization). This optimization problem can then be solved by standard optimization methods. Adaptations of existing techniques such as the Sequential Minimal Optimization have also been developed for multiple kernel SVM-based methods.

Supervised learning

For supervised learning, there are many other algorithms that use different methods to learn the form of the kernel. The following categorization has been proposed by Gonen and Alpayd■n (2011)

Fixed rules approaches

Fixed rules approaches such as the linear combination algorithm described above use rules to set the combination of the kernels. These do not require parameterization and use rules like summation and multiplication to combine the kernels. The weighting is learned in the algorithm. Other examples of fixed rules include pairwise kernels, which are of the form

These pairwise approaches have been used in predicting protein-protein interactions.

Heuristic approaches

These algorithms use a combination function that is parameterized. The parameters are generally defined for each individual kernel based on single-kernel performance or some computation from the kernel matrix. Examples of these include the kernel from Tenabe et al. (2008). Letting π m {\displaystyle \pi _{m}} be the accuracy obtained using only K m {\displaystyle K_{m}} , and letting δ {\displaystyle \delta } be a threshold less than the minimum of the single-kernel accuracies, we can define

Other approaches use a definition of kernel similarity, such as

Using this measure, Qui and Lane (2009) used the following heuristic to define

Optimization approaches

These approaches solve an optimization problem to determine parameters for the kernel combination function. This has been done with similarity measures and structural risk minimization approaches. For similarity measures such as the one defined above, the problem can be formulated as follows:

where K t r a ′ {\displaystyle K'_{tra}} is the kernel of the training set.

Structural risk minimization approaches that have been used include linear approaches, such as that used by Lanckriet et al. (2002). We can define the implausibility of a kernel ω (K) {\displaystyle \omega (K)} to be the value of the objective function after solving a canonical SVM problem. We can then solve the following minimization problem:

where c {\displaystyle c} is a positive constant.

Many other variations exist on the same idea, with different methods of refining and solving the problem, e.g. with nonnegative weights for individual kernels and using non-linear combinations of kernels.

Bayesian approaches

Bayesian approaches put priors on the kernel parameters and learn the parameter values from the priors and the base algorithm. For example, the decision function can be written as

η {\displaystyle \eta } can be modeled with a Dirichlet prior and α {\displaystyle \alpha } can be modeled with a zero-mean Gaussian and an inverse gamma variance prior. This model is then optimized using a customized multinomial probit approach with a Gibbs sampler .

These methods have been used successfully in applications such as protein fold recognition and protein homology problems

Boosting approaches

Boosting approaches add new kernels iteratively until some stopping criteria that is a function of performance is reached. An example of this is the MARK model developed by Bennett et al. (2002)

The parameters α i m {\displaystyle \alpha _{i}^{m}} and b {\displaystyle b} are learned by gradient descent on a coordinate basis. In this way, each iteration of the descent algorithm identifies the best kernel column to choose at each particular iteration and adds that to the combined kernel. The model is then rerun to generate the optimal weights α i {\displaystyle \alpha _{i}} and b {\displaystyle b} .

Semisupervised learning

Semisupervised learning approaches to multiple kernel learning are similar to other extensions of supervised learning approaches. An inductive procedure has been developed that uses a log-likelihood empirical loss and group LASSO regularization with conditional expectation consensus on unlabeled data for image categorization. We can define the problem as follows. Let L = (x i , y i) {\displaystyle L={(x_{i},y_{i})}} be the labeled data, and let U = x i {\displaystyle U={x_{i}}} be the set of unlabeled data. Then, we can write the decision function as follows.

The problem can be written as

where L {\displaystyle L} is the loss function (weighted negative log-likelihood in this case), R {\displaystyle R} is the regularization parameter (Group LASSO in this case), and Θ {\displaystyle \Theta } is the conditional expectation consensus (CEC) penalty on unlabeled data. The CEC penalty is defined as follows. Let the marginal kernel density for all the data be

where ψ m (x) = [K m (x 1 , x) , … , K m (x L , x)] T {\displaystyle \psi _{m}(x)=[K_{m}(x_{1},x),\ldots ,K_{m}(x_{L},x)]^{T}} (the kernel distance between the labeled data and all of the labeled and unlabeled data) and φ m π {\displaystyle \phi _{m}^{\pi }} is a non-negative random vector with a 2-norm of 1. The value of Π {\displaystyle \Pi } is the number of times each kernel is projected. Expectation regularization is then performed on the MKD, resulting in a reference expectation q m p i (y | g m π (x)) {\displaystyle q_{m}^{pi}(y|g_{m}^{\pi }(x))} and model expectation p m π (f (x) | g m π (x)) {\displaystyle p_{m}^{\pi }(f(x)|g_{m}^{\pi }(x))} . Then, we define

where D (Q | | P) = ∑ i Q (i) ln ■ Q (i) P (i) {\displaystyle D(Q||P)=\sum _{i}Q(i)\ln {\frac {Q(i)}{P(i)}}} is the Kullback–Leibler divergence . The combined minimization problem is optimized using a modified block gradient descent algorithm. For more information, see Wang et al.

Unsupervised learning

Unsupervised multiple kernel learning algorithms have also been proposed by Zhuang et al. The problem is defined as follows. Let U = x i {\displaystyle U={x_{i}}} be a set of unlabeled data. The kernel definition is the linear combined kernel K ′ = ∑ i = 1 M β i K m {\displaystyle K'=\sum _{i=1}^{M}\beta _{i}K_{m}} . In this problem, the data needs to be "clustered" into groups based on the kernel distances. Let B i {\displaystyle B_{i}} be a group or cluster of which x i {\displaystyle x_{i}} is a member. We define the loss function as ∑ i = 1 n ■ x i – ∑ x j ∈ B i K (x i , x j) x j ■ 2

{\displaystyle \sum _{i=1}^{n}\left\Vert x_{i}-\sum _{x_{j}\in B_{i}}K(x_{i},x_{j})x_{j}\right\Vert ^{2}} . Furthermore, we minimize the distortion by minimizing ∑ i = 1 n ∑ x j ∈ B i K (x i , x j) ■ x i – x j ■ 2 {\displaystyle \sum _{i=1}^{n}\sum _{x_{j}\in B_{i}}K(x_{i},x_{j})\left\Vert x_{i}-x_{j}\right\Vert ^{2}} . Finally, we add a regularization term to avoid overfitting. Combining these terms, we can write the minimization problem as follows.

where . One formulation of this is defined as follows. Let D ∈ 0 , 1 n × n {\displaystyle D\in {0,1}^{n\times n}} be a matrix such that D i j = 1 {\displaystyle D_{ij}=1} means that x i {\displaystyle x_{i}} and x j {\displaystyle x_{j}} are neighbors. Then, B i = x j : D i j = 1 {\displaystyle B_{i}={x_{j}:D_{ij}=1}} . Note that these groups must be learned as well. Zhuang et al. solve this problem by an alternating minimization method for K {\displaystyle K} and the groups B i {\displaystyle B_{i}} . For more information, see Zhuang et al.

Libraries

Available MKL libraries include

SPG-GMKL : A scalable C++ MKL SVM library that can handle a million kernels.

GMKL : Generalized Multiple Kernel Learning code in MATLAB , does ■ 1 {\displaystyle \ell _{1}} and ■ 2 {\displaystyle \ell _{2}} regularization for supervised learning.

(Another) GMKL : A different MATLAB MKL code that can also perform elastic net regularization

SMO-MKL : C++ source code for a Sequential Minimal Optimization MKL algorithm. Does p {\displaystyle p} -n orm regularization.

SimpleMKL : A MATLAB code based on the SimpleMKL algorithm for MKL SVM.

MKLPy : A Python framework for MKL and kernel machines scikit-compliant with different algorithms, e.g. EasyMKL and others.

References