



WIKIPEDIA  
The Free Encyclopedia

[Main page](#)  
[Contents](#)  
[Featured content](#)  
[Current events](#)  
[Random article](#)  
[Donate to Wikipedia](#)  
[Wikipedia store](#)

Interaction  
[Help](#)  
[About Wikipedia](#)  
[Community portal](#)  
[Recent changes](#)  
[Contact page](#)

Tools  
[What links here](#)  
[Related changes](#)  
[Upload file](#)  
[Special pages](#)  
[Permanent link](#)  
[Page information](#)  
[Wikidata item](#)  
[Cite this page](#)

Print/export  
[Create a book](#)  
[Download as PDF](#)  
[Printable version](#)

Languages  
[Add links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Search

# Structured support vector machine

From Wikipedia, the free encyclopedia  
(Redirected from [Structured SVM](#))

The **structured support vector machine** is a [machine learning](#) algorithm that generalizes the [Support Vector Machine](#) (SVM) classifier. Whereas the SVM classifier supports [binary classification](#), [multiclass classification](#) and [regression](#), the structured SVM allows training of a classifier for general [structured output labels](#).

As an example, a sample instance might be a natural language sentence, and the output label is an annotated [parse tree](#). Training a classifier consists of showing pairs of correct sample and output label pairs. After training, the structured SVM model allows one to predict for new sample instances the corresponding output label; that is, given a natural language sentence, the classifier can produce the most likely parse tree.

**Contents** [\[hide\]](#)

- [1 Training](#)
- [2 Inference](#)
- [3 Separation](#)
- [4 References](#)

## Training [\[edit\]](#)

For a set of  $\ell$  training instances  $(\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathcal{Y}, n = 1, \dots, \ell$  from a sample space  $\mathcal{X}$  and label space  $\mathcal{Y}$ , the structured SVM minimizes the following regularized risk function.

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 + C \sum_{n=1}^{\ell} \max_{y \in \mathcal{Y}} (\Delta(y_n, y) + \mathbf{w}'\Psi(\mathbf{x}_n, y) - \mathbf{w}'\Psi(\mathbf{x}_n, y_n))$$

The function is convex in  $\mathbf{w}$  because the maximum of a set of affine functions is convex. The function  $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$  measures a distance in label space and is an arbitrary function (not necessarily a [metric](#)) satisfying  $\Delta(y, z) \geq 0$  and  $\Delta(y, y) = 0 \ \forall y, z \in \mathcal{Y}$ . The function  $\Psi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$  is a feature function, extracting some feature vector from a given sample and label. The design of this function depends very much on the application.

Because the regularized risk function above is non-differentiable, it is often reformulated in terms of a [quadratic program](#) by introducing one slack variable  $\xi_n$  for each sample, each representing the value of the maximum. The standard structured SVM primal formulation is given as follows.

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \|\mathbf{w}\|^2 + C \sum_{n=1}^{\ell} \xi_n \\ \text{s.t.} \quad & \mathbf{w}'\Psi(\mathbf{x}_n, y_n) - \mathbf{w}'\Psi(\mathbf{x}_n, y) + \xi_n \geq \Delta(y_n, y), \quad n = 1, \dots, \ell, \quad \forall y \in \mathcal{Y} \end{aligned}$$

## Inference [\[edit\]](#)

At test time, only a sample  $\mathbf{x} \in \mathcal{X}$  is known, and a prediction function  $f : \mathcal{X} \rightarrow \mathcal{Y}$  maps it to a predicted label from the label space  $\mathcal{Y}$ . For structured SVMs, given the vector  $\mathbf{w}$  obtained from training, the prediction function is the following.

$$f(\mathbf{x}) = \operatorname{argmax}_{y \in \mathcal{Y}} \mathbf{w}'\Psi(\mathbf{x}, y)$$

Therefore, the maximizer over the label space is the predicted label. Solving for this maximizer is the so-called inference problem and similar to making a maximum a-posteriori (MAP) prediction in probabilistic models. Depending on the structure of the function  $\Psi$ , solving for the maximizer can be a hard problem.

## Separation [\[edit\]](#)

The above quadratic program involves a very large, possibly infinite number of linear inequality constraints. In general, the number of inequalities is too large to be optimized over explicitly. Instead the problem is solved by using [delayed constraint generation](#) where only a finite and small subset of the constraints is used. Optimizing over a subset of the constraints enlarges the [feasible set](#) and will yield a solution which provides a lower bound

on the objective. To test whether the solution  $\mathbf{w}$  violates constraints of the complete set inequalities, a [separation](#)<sup>[*disambiguation needed*]</sup> problem needs to be solved. As the inequalities decompose over the samples, for each sample  $(\mathbf{x}_n, y_n)$  the following problem needs to be solved.

$$y_n^* = \operatorname{argmax}_{y \in \mathcal{Y}} (\Delta(y_n, y) + \mathbf{w}'\Psi(\mathbf{x}_n, y) - \mathbf{w}'\Psi(\mathbf{x}_n, y_n) - \xi_n)$$







The right hand side objective to be maximized is composed of the constant  $-\mathbf{w}'\Psi(\mathbf{x}_n, y_n) - \xi_n$  and a term dependent on the variables optimized over, namely  $\Delta(y_n, y) + \mathbf{w}'\Psi(\mathbf{x}_n, y)$ . If the achieved right hand side objective is smaller or equal to zero, no violated constraints for this sample exist. If it is strictly larger than zero, the most violated constraint with respect to this sample has been identified. The problem is enlarged by this constraint and resolved. The process continues until no violated inequalities can be identified.

If the constants are dropped from the above problem, we obtain the following problem to be solved.

$$y_n^* = \operatorname{argmax}_{y \in \mathcal{Y}} (\Delta(y_n, y) + \mathbf{w}'\Psi(\mathbf{x}_n, y))$$

This problem looks very similar to the inference problem. The only difference is the addition of the term  $\Delta(y_n, y)$ . Most often, it is chosen such that it has a natural decomposition in label space. In that case, the influence of  $\Delta$  can be encoded into the inference problem and solving for the most violating constraint is equivalent to solving the inference problem.

## References [\[edit\]](#)

- Ioannis Tsochantaridis, **Thorsten Joachims**, Thomas Hofmann and Yasemin Altun (2005), [Large Margin Methods for Structured and Interdependent Output Variables](#) , JMLR, Vol. 6, pages 1453-1484.
- Thomas Finley and Thorsten Joachims (2008), [Training Structural SVMs when Exact Inference is Intractable](#) , ICML 2008.
- Sunita Sarawagi and Rahul Gupta (2008), [Accurate Max-margin Training for Structured Output Spaces](#) , ICML 2008.
- Gökhan Bakır, Ben Taskar, Thomas Hofmann, Bernhard Schölkopf, Alex Smola and SVN Vishwanathan (2007), [Predicting Structured Data](#) , MIT Press.
- Vojtěch Franc and Bogdan Savchynskyy [Discriminative Learning of Max-Sum Classifiers](#) , Journal of Machine Learning Research, 9(Jan):67—104, 2008, Microtome Publishing
- Kevin Murphy [\[1\]](#)  Machine Learning, Mit Press

Categories: [Structured prediction](#) | [Support vector machines](#)

This page was last modified on 1 March 2015, at 00:20.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

