Support Vector Machine

ISLR Chapter 9

Yuan YAO

Hong Kong University of Science and Technology

Department of Mathematics

Spring 2024

Best Machine Learning Algorithms in history?

- Boosting (ISLR chapter 8): CART, AdaBoost, Random Forests (Leo Breiman), etc.
- **Support Vector Machines (ISLR chapter 9): or kernel methods (V. Vapnik)**, etc.
- Neural Networks: perceptrons (1950s), deep learning (2010s), CNN/RNN/LSTM ($<$ 2000), ResNet/Transformers ($>$ 2015), etc.

About this chapter

- Vapnik's Support vector machine dominates neural networks during late 1990s and 2000s, more than a decade.
- Empirically successful, with well developed theory (max-margin classification, Vapnik-Chervonenkis Theory, etc.)
- One of the best off-the-shelf methods, based on convex optimization and geometry.

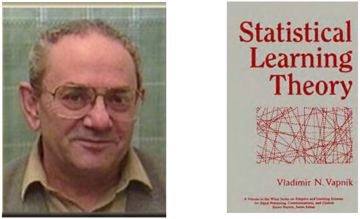


Figure: **Vladimir Naumovich Vapnik** and his classic book

Yann LeCun's MNIST competition in 1998

- *Gradient-Based Learning Applied to Document Recognition*, by Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, *Proceedings of the IEEE*, 86(11):2278-2324, 1998.
- A comprehensive comparisons of machine learning methods in MNIST competition
- Best neural net (Boosted LeNet-4, 0.7%) beats best SVM (V-SVM poly9, 0.8%) by 0.1%, but is hard to tune.

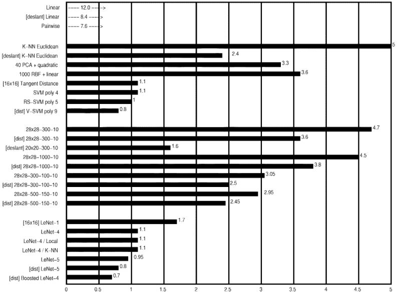


Fig. 9. Error rate on the test set (%) for various classification methods. [deslant] indicates that the classifier was trained and tested on the deslanted version of the database. [dist] indicates that the training set was augmented with artificially distorted examples. [16×16] indicates that the system used the 16×16 pixel images. The uncertainty in the quoted error rates is about 0.1%.

Maximal margin classifier

- Non-separable Support Vector Classifier

Support Vector Machines

- Support vector machines with kernels
- SVMs with more than two classes
- Outlier Detection: 1-class SVM
- Support Vector Regression

* Supplementary Topics

- Kernel Approximation for Large Scale SVM
- Primal-Dual support vector classifiers
- Relationship with logistic regression and AdaBoost

Outline

Maximal margin classifier

- Non-separable Support Vector Classifier

Support Vector Machines

- Support vector machines with kernels
- SVMs with more than two classes
- Outlier Detection: 1-class SVM
- Support Vector Regression

* Supplementary Topics

- Kernel Approximation for Large Scale SVM
- Primal-Dual support vector classifiers
- Relationship with logistic regression and AdaBoost

Hyperplane in R^p

- $\mathbf{x} = (x_1, ..., x_p)^T \in R^p$ (p-dimensional real space).
- Consider all $\mathbf{x}$ satisfying

$$f(\mathbf{x}) = \beta_0 + \beta^T\mathbf{x} = \beta_0 + \langle \beta, \mathbf{x} \rangle = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p = 0.$$

where $\beta = (\beta_1, \ldots, \beta_p)^T$.
All such $\mathbf{x}$ defines a hyperplane: All $\mathbf{x}$ such that its projection on β is

$$\left\langle \frac{\beta}{\|\beta\|}, \mathbf{x} \right\rangle \frac{\beta}{\|\beta\|} = -\beta_0 \frac{\beta}{\|\beta\|^2}.$$

Hyperplane in R^p

- $$f(\mathbf{x}) = 0 \iff \mathbf{x} = -\beta_0 \frac{\beta}{\|\beta\|^2} + y \qquad \text{with } y \perp \beta.$$

- $$f(\mathbf{x}) > 0 \iff \mathbf{x} = \tilde{\beta}_0 \frac{\beta}{\|\beta\|^2} + y \qquad \text{with } y \perp \beta,\ \tilde{\beta}_0 > -\beta_0$$

- $$f(\mathbf{x}) < 0 \iff \mathbf{x} = \tilde{\beta}_0 \frac{\beta}{\|\beta\|^2} + y \qquad \text{with } y \perp \beta,\ \tilde{\beta}_0 < -\beta_0$$

here $\tilde{\beta}_0 = \langle \mathbf{x}, \beta \rangle$ and $f(\mathbf{x}) = \tilde{\beta}_0 - (-\beta_0)$.

- $f(\mathbf{x}) > 0 \iff \mathbf{x}$ is on one side of the hyperplane (at the same direction as β.)
  $f(\mathbf{x}) < 0 \iff \mathbf{x}$ is on the other side of the hyperplane (at the opposite direction as β.)
- For any vector $\mathbf{z} \in R^p$, the signed distance of a point $\mathbf{z} \in R^p$ to this hyperplane is

$$\langle \mathbf{z}, \beta/\|\beta\| \rangle - (-\beta_0/\|\beta\|) = (\langle \mathbf{z}, \beta \rangle + \beta_0)/\|\beta\| = f(\mathbf{z})/\|\beta\|.$$

  - If $\|\beta\| = 1$, $f(\mathbf{z})$ is the signed distance of $\mathbf{z}$ to the hyperplane defined by $f(\mathbf{x}) = 0$.

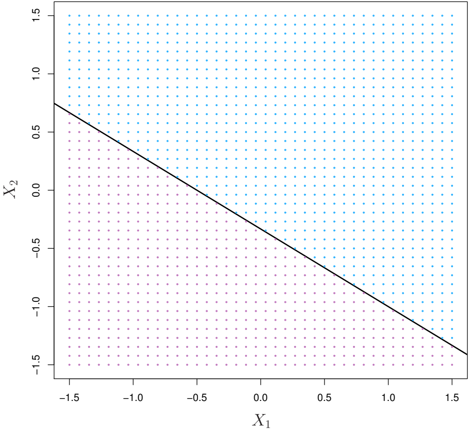


Figure: 9.1. The hyperplane $1 + 2X_1 + 3X_2 = 0$ is shown. The blue region is the set of points for which $1 + 2X_1 + 3X_2 > 0$, and the purple region is the set of points for which $1 + 2X_1 + 3X_2 < 0$.

Separating hyperplane

- Training Data:$(y_i, \mathbf{x}_i), i = 1, ..., n$, with input $\mathbf{x}_i = (x_{i1}, ..., x_{ip})$ and two-class output $y_i = \pm 1$.
- Suppose the two classes are separated by one hyperplane

  $$f(\mathbf{x}) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p,$$

  meaning that, for all i in one class $y_i = 1$,

  $$f(\mathbf{x}_i) > 0, \quad \text{one side of the hyperplane;}$$

  and for all i in the other class $y_i = -1$,

  $$f(\mathbf{x}_i) < 0, \quad \text{the other side of the hyperplane;}$$

- It can be equivalently expressed as

  $$y_i f(\mathbf{x}_i) > 0, \quad \text{for all } i = 1, .., n$$

- If such separating hyperplane exists, it can be our classification rule:
  For any new/old observation with $\mathbf{x}^*$ such that $f(\mathbf{x}^*) > 0$, classify it as in the class $+1$. Otherwise, classify it as in class -1.
- Problem: If the two classes in training data are indeed separable by a hyperplane, which hyperplane is the best?

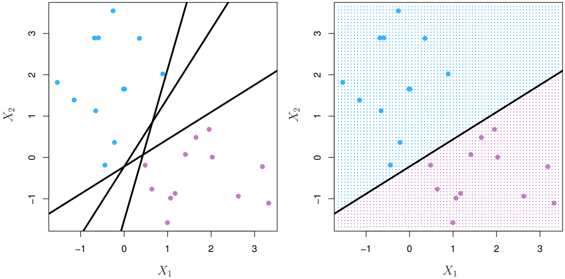


Figure: 9.2. Left: There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black. Right: A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.

Maximal margin classifier

- Maximal margin hyperplane: the separating hyperplane that optimal separating hyperplane is farthest from the training observations.
    - The optimal separating hyperplane is to **maximize the minimum distance** of any training point to the hyperplane (minimax problem).
    - Creates a widest gap separating the two classes.
- Points on the boundary hyperplane, those with smallest distance to the max margin hyperplane, are called **support vectors**.
    - They “support” the maximal margin hyperplane in the sense that if these points were moved slightly then the maximal margin hyperplane would move as well

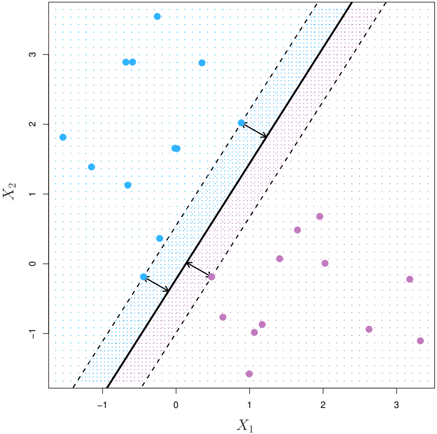


Figure: 9.3. There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the hyperplane is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.

Computing the max margin hyperplane

$$\begin{aligned}
\text{maximize}_{\beta_0,\beta_1,\ldots,\beta_p} \quad & M \\
\text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 = 1, \\
& y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M \text{ for all } i
\end{aligned}$$

This is a convex quadratic programming problem.

- Note that $M > 0$ is the half of the width of the strip separating the two classes.
- The eventual solution, the max margin hyperplane is determined by the support vectors. The max margin hyperplane may vary a lot when the support vectors vary (sensitivity to support vectors, high variance).
- If the non-support vectors x_i that lie on the correct side of the trip vary, the solution would remain same (robustness to non-support vectors).
- Yet, in general cases data points are not separable...

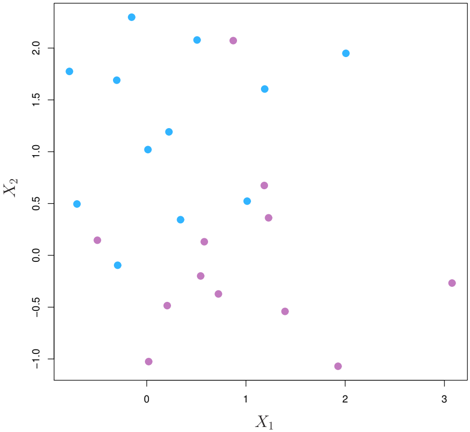


Figure: 9.4. There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.

Even separable, Max Margin classifier is not robust

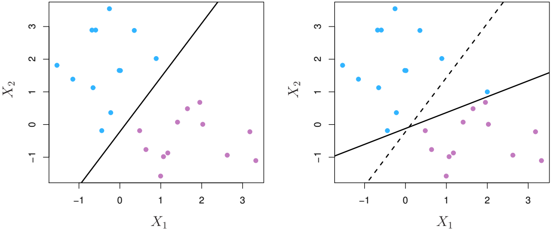


Figure: 9.5. Left: Two classes of observations are shown in blue and in purple, along with the maximal margin hyperplane. Right: An additional blue observation has been added, leading to a dramatic shift in the maximal margin hyperplane shown as a solid line. The dashed line indicates the maximal margin hyperplane that was obtained in the absence of this additional point.

The non-separable case

- In general, the two classes are usually not separable by any hyperplane.
- Even if they are, the max margin may not be robust because of its high variance, and thus possible over-fit.
- The generalization of the maximal margin classifier to the non-separable case is known as the *support vector classifier* (SVC).
- SVC uses a soft-margin in place of the max margin.

Outline

Maximal margin classifier

- Non-separable Support Vector Classifier

Support Vector Machines

- Support vector machines with kernels
- SVMs with more than two classes
- Outlier Detection: 1-class SVM
- Support Vector Regression

* Supplementary Topics

- Kernel Approximation for Large Scale SVM
- Primal-Dual support vector classifiers
- Relationship with logistic regression and AdaBoost

Non-perfect separation

- Consider a classifier based on a hyperplane that does not perfectly separate the two classes, in the interest of
  1. Greater robustness to individual observations, and
  2. Better classification of most of the training observations without separability assumption.
- Soft-margin classifier (support vector classifier) allow some violation of the margin: some can be on the wrong side of the margin (in the river) or even wrong side of the hyperplane.

Computing the soft-margin classifier

$$\begin{aligned} \text{maximize} \quad & M \\ \text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 = 1, \qquad \text{and for all } i \\ & y_i(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \\ & \epsilon_i \geq 0 \\ & \sum_{i=1}^{n} \epsilon_i \leq C \end{aligned}$$

where C is a nonnegative tuning parameter, ϵ_i's are *slack variables*.

The support vector classifier

- Another equivalent formulation is (`Libsvm` and `sklearn.svm`) for $M = 1/\|\beta\|$:

$$
\begin{aligned}
\text{minimize}_{\beta_0,\beta_1,\dots,\beta_p} \quad & \frac{1}{2}\|\beta\|^2 + C' \sum_{i=1}^{n} \xi_i \\
\text{subject to} \quad & y_i(\beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}) \geq 1 - \xi_i, \\
& \xi_i \geq 0, \qquad \text{for all } i
\end{aligned}
$$

- The solution of these optimizations is the support vector classifier:

$$f(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p.$$

  And we classify an observation in $+1$ class if $f(\mathbf{x}) > 0$; else into -1 class.

Understaning the slack variable ϵ_i

- $\epsilon_i = 0 \iff$ the *i*-th observation is on the correct side of the margin
- $\epsilon_i > 0 \iff$ the *i*-th observation is on the *wrong* side of the **margin**
- $\epsilon_i > 1 \iff$ the *i*-th observation is on the *wrong* side of the **hyperplane**.

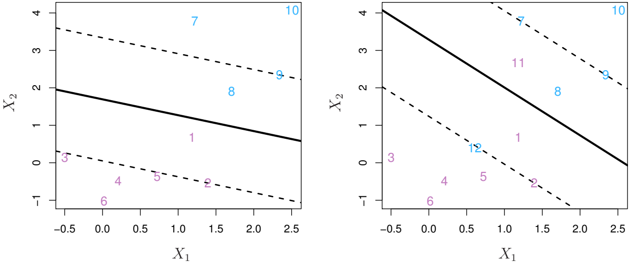


Figure: 9.6. next page

FIGURE 9.6. Left: A support vector classifier was fit to a small data set. The hyperplane is shown as a solid line and the margins are shown as dashed lines. Purple observations: Observations 3, 4, 5, and 6 are on the correct side of the margin, observation 2 is on the margin, and observation 1 is on the wrong side of the margin. Blue observations: Observations 7 and 10 are on the correct side of the margin, observation 9 is on the margin, and observation 8 is on the wrong side of the margin. No observations are on the wrong side of the hyperplane. Right: Same as left panel with two additional points, 11 and 12. These two observations are on the wrong side of the hyperplane and the wrong side of the margin.

Understanding tuning parameter C (C')

- C (C') is a *budget* for the amount that the margin can be violated by the n observations
- $C = 0 \iff$ no budget $\Rightarrow \epsilon_i = 0$ for all i.
  The classifier is a maximal margin classifier, which exists only if the two classes are separable by hyperplanes.
- Bias-Variance tradeoff by C (C'):
  - Larger C, more tolerance of margin violation, larger bias but smaller variance.
  - Small C, high variance and small bias.
  - C' is in the opposite way (as C in `Libsvm` and `sklearn.svm`).
- C and C' can be determined by using cross validation.

Support vectors

- A *margin error* corresponds to a sample that lies on the wrong side of its margin boundary: it is either misclassified, or it is correctly classified but does not lie beyond the margin.
- Observations that lie directly on the margin, or on the wrong side of the margin for their class, are known as *support vectors*.
- Only the support vectors affect the support vector classifier.
- Those strictly on the correct side of the margin do not. (robustness, analogous to median)
- Larger $C \implies$ more violations, $\implies$ more support vectors, $\implies$ smaller variance, larger bias and more robust classifier.

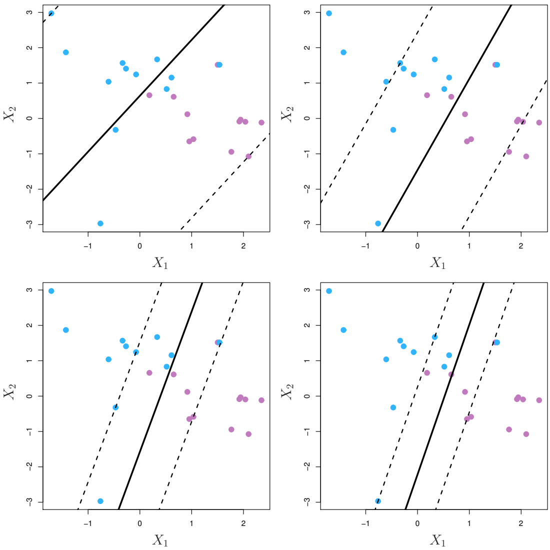


Figure: 9.7.

Figure 9.7. A support vector classifier was fit using four different values of the tuning parameter C in (9.12) and C' (9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance for observations being on the wrong side of the margin, and so the margin will be large. As C decreases, the tolerance for observations being on the wrong side of the margin decreases, and the margin narrows.

Python `sklearn.svm` Implementations

- It is Python wrapper based on `Libsvm` and `Liblinear` by Chih-Jen Lin et al.
- In `sklearn.svm`, parameter C refers to C' above.
- Schölkopf etal. [Neural Computation 12:12071245 (2000)] proposed ν-SVC, as a reparameterization of such a C-SVC in `sklearn.svm` and therefore mathematically equivalent.
- ν controls the number of support vectors and margin errors: is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors.

Nonlinear Decision Boundary?

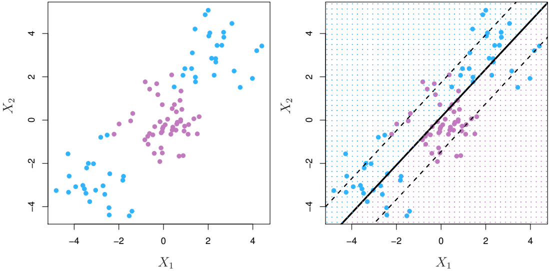


Figure: 9.8. Nonlinear boundaries. Left: The observations fall into two classes, with a non-linear boundary between them. Right: The support vector classifier seeks a linear boundary, and consequently performs very poorly.

Extending to nonlinear boundary

- In practice, we are sometimes faced with non-linear class boundaries
- Linear classifier could perform poorly.
- Need nonlinear classifier.
- As in the extension to the polynomial regression from linear regression, we can consider *enlarge the feature space* from the original p inputs to polynomials (of certain order) of the inputs.

Extending to quadratic inputs

- Rather than constructing the support vector classifier using p features:
$$X_1, ..., X_p.$$
- we use $2p$ features:
$$X_1, X_1^2, ..., X_p, X_p^2.$$
- Treat them as $2p$ original inputs, and fit the support vector classifier.
- The separating hyperplane is a hyperplane in R^{2p}, which should be a linear equation:
$$\beta_0 + \beta_1 X_1 + \beta_2 X_p + \beta_{p+1} X_1^2 + ... + \beta_{2p} X_p^2 = 0$$
- This is a quadratic equation in $X_1, ..., X_p$. Thus the separating surface in R^p in terms of $X_1, ..., X_p$ corresponds to a quadratic surface in R^p.

Extending to polynomial inputs

- Can extend to polynomial of any given order d.
- Could lead to too many features, too large feature space; thus overfit.
- Higher powers are unstable.

Key observation from dual problem

- The linear support vector classifier has a dual representation:

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- $\alpha_i \neq 0$ only for all support vectors.
- α_i can also be computed based on $\langle \mathbf{x}_j, \mathbf{x}_k \rangle$.
- Only the inner product of the feature space is relevant in computing the linear support vector classifier.

The kernel trick

- The inner product $\langle \cdot, \cdot \rangle$ is a bivariate function (symmetric and positive definite).
- It can be generalized to (Mercer) kernel functions

$$K(\mathbf{x}, \mathbf{z})$$

which is **symmetric, positive definite**.

- The classifier can be expressed as

$$f(\mathbf{x}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

Examples of kernels

- Examples of the kernel function are:
- linear kernel

$$K(x_i, x_j) = \langle x_i, x_j \rangle = x_i^T x_j.$$

- polynomial kernel of degree d:

$$K(x_i, x_j) = (1 + \langle x_i, x_j \rangle)^d.$$

- Gaussian radial kernel:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \quad \gamma > 0.$$

- Only the inner product of the feature space is relevant in computing the linear support vector classifier.

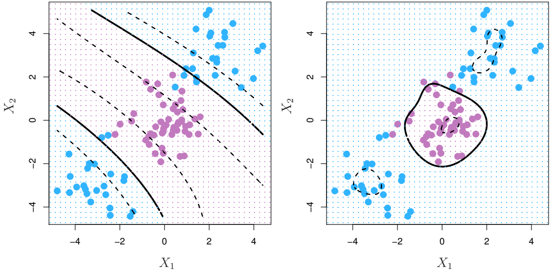


Figure: 9.9. Left: An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule. Right: An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.

- The radial kernel has local behavior.
- To predict the class for a new observation with input $\mathbf{x}$,

$$f(\mathbf{x}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)$$

- A training data point $\mathbf{x}_i$ being far away from $\mathbf{x}$ will have little effect to the sign of $f(\mathbf{x})$.

The enlarged feature space.

- The support vector machine actually enlarges the original feature space to a space of kernel functions.

$$\mathbf{x}_i \to K(\cdot, \mathbf{x}_i).$$

- The original space of p inputs has dimension p.
- The enlarged space of features, the function space, is **infinite dimension** (**reproducing kernel Hilbert space**)!
- In actual fitting of the support vector machine, we only need to compute the $K(x_i, x_j)$ for all x_i, x_j in training data.
- Do not have to work with the enlarged feature space of infinite dimension.

Example: the Heart data.

- In Chapter 8 we apply decision trees and related methods to the Heart data.
- The aim is to use 13 predictors such as Age, Sex, and Chol in order to predict whether an individual has heart disease.
- We now investigate how an SVM compares to LDA on this data.
- 297 subjects, randomly split into 207 training and 90 test observations.

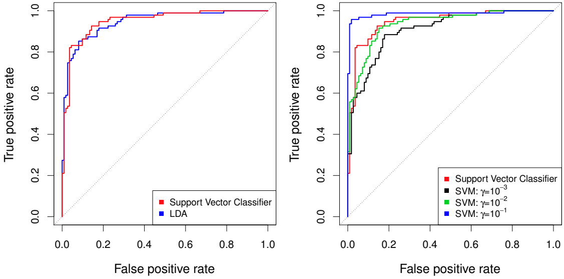


Figure: ROC curves for the Heart data training set. Left: The support vector classifier and LDA are compared. Right: The support vector classifier is compared to an SVM using a radial basis kernel with $\gamma = 10^{-3}, 10^{-2}$ and 10^{-1}.

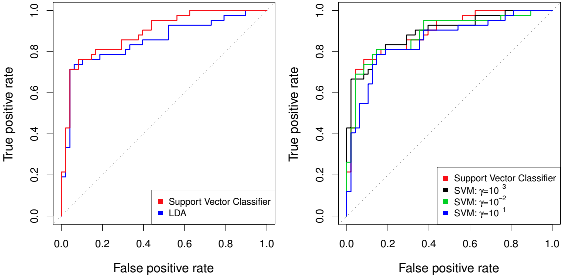


Figure: 9.11. ROC curves for the test set of the Heart data. Left: The support vector classifier and LDA are compared. Right: The support vector classifier is compared to an SVM using a radial basis kernel with $\gamma = 10^{-3}, 10^{-2}$ and 10^{-1}.

Multiclass SVM: One-versus-one approach

- With $K > 2$ classes.
- Run a SVM on each of the $\binom{K}{2}$ pairs of classes.
- We obtain $\binom{K}{2}$ SVMs.
- For every test observations, compute the number of times it is classified into class k by all the SVMs, denote as s_k.
- Classify it into the class with highest score s_k (majority vote).
- This is implemented in `sklearn.svm` for `SVC` and `nuSVC`.
- Shortcoming: train $O(K^2)$ SVMs, intractable for large K.

Multiclass SVM: One-versus-all(rest) approach

- With $K > 2$ classes.
- Run a SVM on class k (coded as $+1$) versus class "not-k" (coded as -1): $f_k(\mathbf{x})$. (Note that the larger $f_k(\mathbf{x})$, the more likely $\mathbf{x}$ is in class k.
- For a new test observation with $\mathbf{x}$, assign to the class with largest $f_k(\mathbf{x})$.
- This is implemented in `sklearn.svm.linearSVC`
- Shortcoming: imbalance issue.

One-Class Support Vector Classification

- Schölkopf et al. (2001), for l samples $\boldsymbol{x}_i \in \mathbb{R}^d$ without y_i,

$$\begin{array}{ll} \min_{\boldsymbol{w},\boldsymbol{\xi},\rho} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \rho + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ \text{subject to} & \boldsymbol{w}^T\phi(\boldsymbol{x}_i) \geq \rho - \xi_i \\ & \xi_i \geq 0, i = 1, \ldots, l \end{array}$$

- It tries to include most of samples into the super-level set $f(\boldsymbol{x}_i) = \boldsymbol{w}^T\phi(\boldsymbol{x}_i) \geq \rho$ (inliers), with possible violation $\xi_i \geq 0$ (outliers)
  - The smaller ρ, the less violations ξ_i
  - A trade-off between ρ and $\sum_i \xi_i$ with hyper-parameter ν
  - The larger ν, the larger ρ, the less inliers, the more outliers

Dual of One-Class Support Vector Classification

- Dual formulation:

$$\begin{array}{cl} \min_{\alpha} & \frac{1}{2}\alpha^T K\alpha, \qquad K_{ij} = \langle\phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle, \\ \text{subject to} & 0 \leq \alpha_i \leq 1/(\nu l), i = 1, \ldots, l \\ & e^T\alpha = 1 \end{array}$$

- Decision function:

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) - \rho\right)$$

Example: One-Class Support Vector Classification

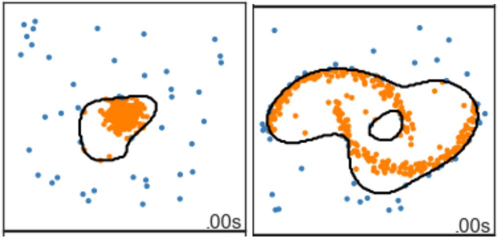


Figure: One-Class Support Vector Classification for outlier detection: outlier (blue) and inlier (yellow). Large ν on the left, small ν on the right.

ε-Support Vector Regression

- Vapnik (1998) proposed

$$\min_{w} C \sum_{i=1}^{n} V_{\varepsilon}(y_i, f_w(x_i)) + \frac{1}{2}\|w\|^2,$$

  where

  - $V_{\epsilon}(y, \hat{y}) = \max(0, |y - \hat{y}| - \varepsilon)$ is the ε-insensitive loss.
  - Decision function $f_w(x) := \langle w, \phi(x) \rangle + b$

Dual of ε-Support Vector Regression

- $K_{ij} = K(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle$
- Dual formulation:

$$\begin{aligned} \min_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad & \frac{1}{2} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^* \right)^T K \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^* \right) + \epsilon \sum_{i=1}^{n} \left(\alpha_i + \alpha_i^* \right) + \sum_{i=1}^{n} y_i \left(\alpha_i - \alpha_i^* \right) \\ \text{subject to} \quad & e^T \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^* \right) = 0 \\ & 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, \ldots, n \end{aligned}$$

- Decision function
  $f_w(x) := \langle w, \phi(x) \rangle + b = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) K(x_i, x) + b$

Example: Support Vector Regression

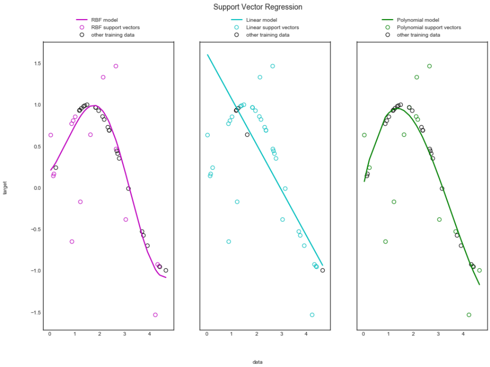


Figure: Support Vector Regression with RBF, linear and polynomial kernels

Outline

Maximal margin classifier

Non-separable Support Vector Classifier

Support Vector Machines

Support vector machines with kernels

SVMs with more than two classes

Outlier Detection: 1-class SVM

Support Vector Regression

* Supplementary Topics

Kernel Approximation for Large Scale SVM

Primal-Dual support vector classifiers

Relationship with logistic regression and AdaBoost

Kernel Approximation for Large Scale SVM

- SVM needs a n-by-n Kernel matrix
- Hence not scalable for big data $n >> 10k$
- Kernel Approximation: `sklearn.kernel_approximation`
  - Nyström (`Nystroem`) method
  - Random Fourier Features (Rahimi, A. and Recht, B, NIPS'2007)
  - and other methods

Nyström Method

- Let

$$\mathbf{K}^{n,n} = \left[\begin{array}{c|c} \mathbf{A}^{k,k} & \mathbf{B}^{k,n-k} \\ \hline \mathbf{B}^T & \mathbf{C} \end{array}\right] \succeq 0$$

$$\Rightarrow \mathbf{K} = \begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{Y} \\ \mathbf{Y}^T\mathbf{X} & \mathbf{Y}^T\mathbf{Y} \end{bmatrix}$$

  where

$$\mathbf{A} = \mathbf{X}^T\mathbf{X}$$
$$\mathbf{B} = \mathbf{X}^T\mathbf{Y}$$

- Nyström approximates **K** by

$$\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} \end{bmatrix}$$

  with approximation error $\|\mathbf{C} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}\|$.

Nyström Approximation: computing $\mathbf{X}, \mathbf{Y}$

- Take
$$\mathbf{A} = \mathbf{U}\boldsymbol{\Gamma}\mathbf{U}^T$$
and
$$\mathbf{X} = \boldsymbol{\Gamma}_{[k]}^{1/2}\mathbf{U}_{[k]}^T$$
where the subscript $[k]$ indicates the submatrices corresponding to the eigenvectors with the k largest positive eigenvalues. Then
$$\mathbf{Y} = \mathbf{X}^{-T}\mathbf{B} = \boldsymbol{\Gamma}_{[k]}^{-1/2}\mathbf{U}_{[k]}^T\mathbf{B}$$

Random Fourier Features

- Rahimi, A. and Recht, B, NIPS'2007
- Mercer Theorem: continuous, positive definite $K(x, x')$ on compact domain admits

$$K(x, x') = \sum_{\alpha} \lambda_\alpha \phi_\alpha(x) \phi_\alpha(x')$$

  where $\lambda_\alpha \geq 0$ and ϕ_α are orthonormal eigenvectors.
- For translation invariant kernels (e.g. rbf), use random Fourier basis instead of unknown ϕ:

$$K(x - x') = \sum_{\omega} \widehat{q}_\omega \exp(-i \left\langle \omega, x - x' \right\rangle)$$

  where $\widehat{q}_\omega$ is some probability measure.

Equivalent reformulation of Hard Margin

$$\text{maximize}_{\beta_0,\beta_1,\ldots,\beta_p} M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

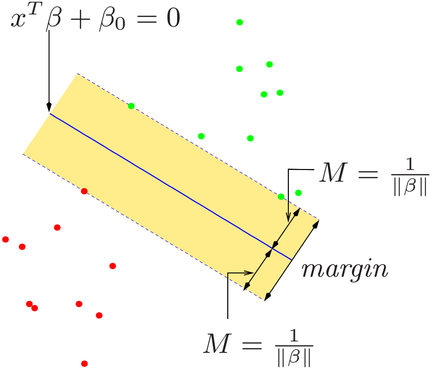
$$\text{and } y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M \text{ for all } i$$

Using $M = 1/\|\beta\|$,

$$\Leftrightarrow$$

$$\text{minimize}_{\beta_0,\beta_1,\ldots,\beta_p} \|\beta\|^2 := \sum_j \beta_j^2$$

$$\text{subject to } y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq 1 \text{ for all } i$$


$x^T\beta + \beta_0 = 0$
$M = \frac{1}{\|\beta\|}$
margin
$M = \frac{1}{\|\beta\|}$

On why $M = 1/\|\beta\|$?

- Separating hyperplane P_0, $x^T\beta + \beta_0 = 0$, has a unit normal vector $\beta/\|\beta\|$.
- Choose a point $\mathbf{x}_0$ on P_0, $\mathbf{x}_0^T\beta + \beta_0 = 0$.
- Consider $\mathbf{x}_1 = \mathbf{x}_0 + M\beta/\|\beta\|$, a normal vector of length M, starting from $\mathbf{x}_0$ and intersecting the margin hyperplane P_1, $x^T\beta + \beta_0 = 1$. Then

$$
\begin{aligned}
1 &= \mathbf{x}_1^T\beta + \beta_0 = \mathbf{x}_0^T\beta + M\frac{\beta^T\beta}{\|\beta\|} - \beta_0 \\
\Longrightarrow 1 &= \mathbf{x}_0^T\beta + M\|\beta\| - \beta_0 \\
\Longrightarrow 1 - M\|\beta\| &= \mathbf{x}_0^T\beta - \beta_0 = 0 \\
\Longrightarrow M &= \frac{1}{\|\beta\|}
\end{aligned}
$$

Lagrangian

Lagrangian: for $\alpha_i \geq 0$,

$$\max_{\alpha \geq 0} \min_{\beta, \beta_0} L(\beta, \alpha) := \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) - 1)$$

So

$$0 = \frac{\partial L}{\partial \beta} \Rightarrow \hat{\beta} = \sum_i \hat{\alpha}_i y_i \mathbf{x}_i$$

$$0 = \frac{\partial L}{\partial \beta_0} \Rightarrow \sum_i \hat{\alpha}_i y_i = 0$$

and complementary condition

$$\hat{\alpha}_i (y_i(\hat{\beta}_0 + \left\langle \hat{\beta}, \mathbf{x}_i \right\rangle) - 1) = 0, \quad \text{for all } i$$

Support Vectors

Complementary condition $\hat{\alpha}_i(y_i(\hat{\beta}_0 + \left\langle \hat{\beta}, \mathbf{x}_i \right\rangle) - 1) = 0, \quad$ for all i implies that

$$y_i(\hat{\beta}_0 + \left\langle \hat{\beta}, \mathbf{x}_i \right\rangle) > 1 \Rightarrow \hat{\alpha}_i = 0$$

$$y_i(\hat{\beta}_0 + \left\langle \hat{\beta}, \mathbf{x}_i \right\rangle) = 1 \Rightarrow \hat{\alpha}_i \geq 0$$

Those sample point i's such that $\hat{\alpha}_i > 0$, are called *support vectors* (sv), which decided the maximal margin hyperplane

$$\hat{\beta} = \sum_{i \in sv} \hat{\alpha}_i y_i \mathbf{x}_i$$

and $\hat{\beta}_0$ can be uniquely decided by any support vector s using $\hat{\beta}_0 = y_s - \left\langle \hat{\beta}, \mathbf{x}_s \right\rangle$.

Dual Formulation

After plugging $\hat{\beta}$ in the Lagrangian, it gives

$$\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

subject to

$$\alpha_i \geq 0$$

$$\sum_i \alpha_i y_i = 0$$

Equivalent reformulation of Soft-margin Classifier

$$\begin{aligned}
\text{maximize} \quad & M \\
\text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 = 1, \\
& y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \\
& \epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C, \qquad \text{and for all } i
\end{aligned}$$

Taking $M = 1/\|\beta\|$, then it is equivalent to

$$\begin{aligned}
\text{minimize}_{\beta_0,\beta_1,\ldots,\beta_p} \quad & \frac{1}{2}\|\beta\|^2 + C' \sum_{i=1}^{n} \xi_i \\
\text{subject to} \quad & y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq 1 - \xi_i, \\
& \xi_i \geq 0, \qquad \text{for all } i
\end{aligned}$$

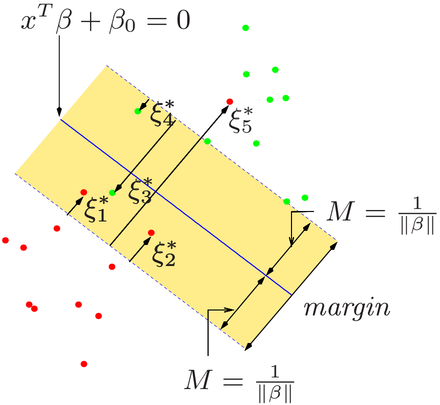


Figure: Separating hyperplane with margin

The Lagrangian

Lagrangian: for $\alpha_i \geq 0$, $\mu_i \geq 0$, $\xi_i \geq 0$,

$$L(\beta, \xi, \alpha, \mu) = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(\beta_0 + \mathbf{x}_i^T\beta) - (1-\xi_i)] - \sum_{i=1}^{n}\mu_i\xi_i,$$

So for all i,

$$0 = \frac{\partial L}{\partial \beta} \Rightarrow \hat{\beta} = \sum_i \hat{\alpha}_i y_i \mathbf{x}_i \qquad (1)$$

$$0 = \frac{\partial L}{\partial \beta_0} \Rightarrow \sum_i \hat{\alpha}_i y_i = 0 \qquad (2)$$

$$0 = \frac{\partial L}{\partial \xi_i} \Rightarrow \hat{\alpha}_i + \mu_i = C \qquad (3)$$

and complementary condition

$$\hat{\alpha}_i[y_i(\hat{\beta}_0 + \mathbf{x}_i^T\hat{\beta}) - (1 - \xi_i)] = 0$$

$$\mu_i\xi_i = 0$$

Support Vectors

Complementary condition $\hat{\alpha}_i[y_i(\hat{\beta}_0 + \mathbf{x}_i^T \hat{\beta}) - (1 - \xi_i)] = 0$, $\xi_i \geq 0$, $\Rightarrow$

$$y_i(\hat{\beta}_0 + \left\langle \hat{\beta}, \mathbf{x}_i \right\rangle) > 1 \Rightarrow \hat{\alpha}_i = 0$$

$$y_i(\hat{\beta}_0 + \left\langle \hat{\beta}, \mathbf{x}_i \right\rangle) = 1 - \xi_i \Rightarrow C \geq \hat{\alpha}_i \geq 0$$

$$\mu_i \xi_i = 0 \Rightarrow \xi_i > 0, \mu_i = 0, \hat{\alpha}_i = C - \mu_i = C$$

Those samples such that $C \geq \hat{\alpha}_i > 0$, are called *support vectors* (sv),

$$\hat{\beta} = \sum_{i \in sv} \hat{\alpha}_i y_i \mathbf{x}_i$$

and those s.t. $\hat{\alpha}_i = C$ are within margin (violators), $\xi_i = 0$ are on margin (deciding $\hat{\beta}_0$).

Dual Problem

Substituting (1)-(3) into the Lagrangian, it gives the dual optimization problem

$$\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^{n} \alpha_i$$

subject to

$$C \geq \alpha_i \geq 0$$

$$\sum_{i} \alpha_i y_i = 0$$

One can replace $\mathbf{x}_i^T \mathbf{x}_j$ by $K(\mathbf{x}_i, \mathbf{x}_j)$ using the kernel trick for nonlinear SVMs.

Recall the "Loss + Penalty" formula

- Minimize, for f in certain space,

$$\sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i)) + \lambda P(f)$$

where in binary classification, $\ell(u) = \ell(y_i f(\mathbf{x}_i))$ with $y_i = \pm 1$.

- Examples:
  - the hinge loss function : $\ell(u) = (1 - u)_+$.
  - the logistic loss function: $\ell(u) = \log(1 + e^{-u})$.
  - the exponential loss function: $\ell(u) = e^{-u}$.
  - the perceptron loss function: $\ell(u) = u_+$.

Logistic regression for classification

- Data: $(y_i, \mathbf{x}_i)$, $y_i = \pm 1$.
- The logistic model assumes

$$P(Y = 1|X) = 1/(1{+}e^{-f(X)}); \quad P(Y = -1|X) = 1/(1{+}e^{f(X)})$$

That is

$$P(Y = y|X) = 1/(1 + e^{-yf(X)})$$

Cross Entropy and Exponential Loss

- The negative of logistic likelihood (a.k.a. binomial deviance, cross-entropy)

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(\mathbf{x}_i)})$$

- Note that in AdaBoost, exponential loss is used

$$\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

Logistic regression and AdaBoost with regularization

- Logistic loss with ridge l_2 penalty

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(\mathbf{x}_i)}) + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Logistic loss with Lasso l_1 penalty:

$$\sum_{i=1}^{n} \log(1 + e^{-y_i f(\mathbf{x}_i)}) + \lambda \sum_{j=1}^{p} |\beta_j|$$

- AdaBoost uses *coordinate descent* to solve l_1-penalty approximately

$$\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} + \lambda \sum_{j=1}^{p} |\beta_j|$$

SVM

- Data: $(y_i, \mathbf{x}_i)$, $y_i = \pm 1$.
- SVM is a result of "hinge loss + ridge penalty":

$$\sum_{i=1}^{n} \max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^{p} \beta_j^2.$$

where $f(\mathbf{x}) = \beta_0 + \beta_1 x_1 + ... + \beta x_p$. Note that $\xi_i = \max[0, 1 - y_i f(x_i)] \geq 0$.

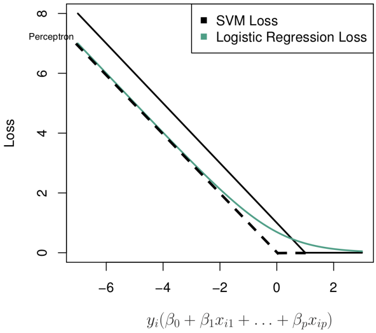
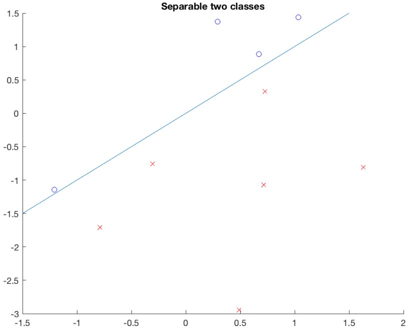


Figure: 9.12. The SVM and logistic regression loss functions are compared, as a function of $y_i(\beta_0 + \beta_1 x_{i1} + ...\beta_p x_{ip}) = \hat{y}_i$. When $y_i(\beta_0 + \beta_1 x_{i1} + ...\beta_p x_{ip})$ is greater than 1, then the SVM loss is zero, since this corresponds to an observation that is on the correct side of the margin. Overall, the two loss functions have quite similar behavior. Note that the dashed curve is the perceptron loss.

A Separable Two-Class Problem



Logistic Regression with Gradient Descent meets Max-Margin Solution Asymptotically

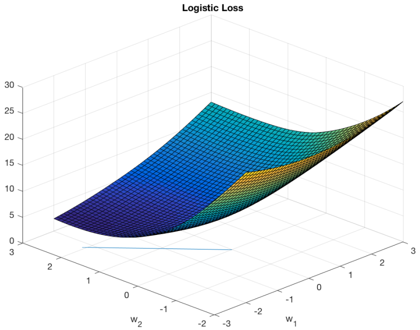


Figure: Logistic loss surface: the line indicates the direction of max-margin solution of support vector machine for the separable two-class problem. By Soudry, Hoffer, Nacson, Gunasekar, Srebro. ICLR 2018.

Exponential Loss Minimization with Gradient Descent meets Max-Margin Solution Asymptotically

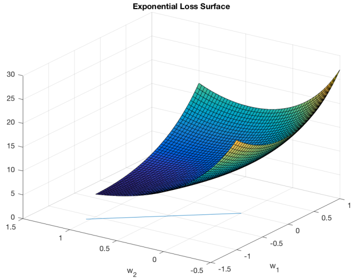


Figure: Exponential loss surface and the max-margin solution of SVM (line direction). By Matus Telgarsky. ICML 2013.