STATISTICAL LEARNING SYSTEMS

LECTURE 6: CLASSIFICATION contd.: KERNEL- OR SIMILARITY-BASED CLASSIFICATION

Jacek Koronacki

Institute of Computer Science, Polish Academy of Sciences

Ph. D. Program 2013/2014

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

For a linear (maximum margin) SVM, assuming separability of (two) classes, we get the optimization task

$$\text{minimize}_{\mathbf{w},b} \frac{\|\mathbf{w}\|^2}{2}$$

subject to

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geqslant 0, \quad i = 1, \dots, n.$$

The Lagrangian for this problem is

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{n} \alpha_i \Big\{ [(\mathbf{x}_i \cdot \mathbf{w}) + b] y_i - 1 \Big\},$$

where $\boldsymbol{\alpha}$ is the vector of nonnegative Lagrange multipliers α_i, $i = 1, \dots, n$. Our task is thus to find the saddle point where the function attains its maximum wrt $\alpha_i \geqslant 0$, $i = 1, \dots, n$ and minimum wrt $\mathbf{w}$ and b.

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

By the Karush-Kuhn-Tucker Theorem, we seek $\boldsymbol{\alpha}$, $\mathbf{w}$ and b such that

$$\mathbf{w} = \sum_{i=1}^{n} y_i \; \alpha_i \; \mathbf{x}_i, \tag{1}$$

and

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2}$$

Substituting (1) and (2), the Lagrangian assumes the form

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j), \tag{3}$$

which is to be maximized under the conditions that

$$\alpha_i \geqslant 0, \quad i = 1, \ldots, n, \qquad \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{4}$$

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

By the Karush-Kuhn-Tucker Theorem we also know that

$$\alpha_i \Big\{ \big[(\mathbf{x}_i \cdot \mathbf{w}_0) + b_0\big] y_i - 1 \Big\} = 0, \quad i = 1, \ldots, n. \tag{5}$$

Condition (5) implies that all the α_i corresponding to the $\mathbf{x}_i$ which are not support vectors must equal zero.

From relations (3) and (4) we get the optimal Lagrange multipliers $\boldsymbol{\alpha}^0 = (\alpha_1^0, \ldots, \alpha_n^0)$. Hence, taking into account (1), the optimal discriminating hyperplane has the form

$$\sum_{\text{support vectors}} y_i \; \alpha_i^0 \; (\mathbf{x}_i \cdot \mathbf{x}) + b^0 = 0, \tag{6}$$

where b^0 satisfies condition (5).

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

One can show that the following b^0 satisfies (5),

$$b^0 = \frac{1}{2}\left[(\mathbf{w}\cdot x^*(1)) + (\mathbf{w}\cdot x^*(-1))\right],$$

where $x^*(1)$ is any fixed support vector from class 1, $x^*(-1)$ is any fixed support vector from class -1 and $\mathbf{w}$ is given by (1) with $\alpha_i = \alpha_i^0$.

Finally the optimal decision rule is,

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{\text{support vectors}} y_i\ \alpha_i^0\ (\mathbf{x}_i\cdot\mathbf{x}) + b^0\right). \qquad (7)$$

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

If the classes are not separable, as is most often the case, we relax constraints by adding nonnegative slack variables, i.e., the constraints take the form

$$\mathbf{x}_i \cdot \mathbf{w} + b \geqslant 1 - \xi_i, \quad \text{gdy} \quad y_i = +1, \tag{8}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leqslant -1 + \xi_i, \quad \text{gdy} \quad y_i = -1, \tag{9}$$

and

$$\xi_i \geqslant 0, \quad i = 1, \ldots, n. \tag{10}$$

Given these constraints, our task is to minimize

$$\frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^{n} \xi_i, \tag{11}$$

where C is a fixed positve constant.

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

It can be shown that we get the same Lagrangian,

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j),$$

which is to be maximized under the conditions that

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, n, \qquad \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{12}$$

The optimal decision rule for this (soft margin) SVM is again given by (7), albeit with another coefficients α_i^0 and another b^0.

PhD STUDIES
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

The so-called kernel trick makes the switch from linear to nonlinear SVMs extraordinarily simple and straightforward.

Before we introduce it, let us note how simple it is to move from a linear SVM to the one with a polynomial decision rule. Indeed, it suffices to recall that all we need is to build a richer feature space, i.e. to replace the $x \in R^p$ by some suitably defined $\mathbf{h}(\mathbf{x})$; e.g., a quadratic decision rule can be introduced as a linear decision rule in $p(p+3)/2$ dimensional vector space with attributes:

$$z^{(1)} = x^{(1)}, \ldots, z^{(p)} = x^{(p)},$$

$$z^{(p+1)} = \left(x^{(1)}\right)^2, \ldots, z^{(2p)} = \left(x^{(p)}\right)^2,$$

$$z^{(2p+1)} = x^{(1)}x^{(2)}, \ldots, z^{(d(d+3)/2)} = x^{(p)}x^{(p-1)}$$

(note that there are $p + p + \frac{p(p-1)}{2} = \frac{p(p+3)}{2}$ features $z^{(j)}$).

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

It goes without saying that, at least conceptually, it is equally simple to build a polynomial decision rule with a polynomial of any fixed order d - only the feature space, $\mathbf{h}(\mathbf{x})$, assumes a (possibly much) larger dimension.

But how does it relate to SVMs? It does in an extraordinarily simple way, this being due to two facts:

- Actually, it is not the new, enlarged, feature space which is involved in any calculations. We even do not have to know its form! All we need is to know how to calculate the needed scalar products $\mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j)$ and $\mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x})$ (cf. (3), (5), (6) and (7)).
- It is the kernel trick which gives the way to calculate the needed scalar products.

Still another fact of utmost importance is that a kernel function can be defined over a set of points of next to any structure (bags of words, strings, trees, graphs, etc.)

PhD STUDIES

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

The project is co-financed by the European Union within the framework of European Social Fund

Support Vector Machines (SVM) - contd.

For a brief intro to kernel methods please see the last slides and for a bit more see a separate handout.

Here, let us note loosely that a kernel is a symmetric function that defines a scalar (inner, dot) product in some feature space. In particular, a polynomial kernel of order d,

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x} \cdot \mathbf{x}')^d, \tag{13}$$

defines a scalar product in the feature space whose features are all products of order d and lower of the original coordinates $x^{(j)}$. For any given kernel, the optimal decision rule is of the form

$$f(\mathbf{x}) = \mathrm{sgn}\ \Big(\sum_{\text{support vectors}} y_i\ \alpha_i^0\ K(\mathbf{x}_i, \mathbf{x}) + b^0 \Big). \tag{14}$$

While in the original space the discrimination surface is nonlinear (in our example polynomial of order d or lower), it is linear in the feature space and all the calculations are performed in full analogy with those for the linear SVM.

Support Vector Machines (SVM) - contd.

We shall conclude these remarks by mentioning or elaborating a bit on the following issues:

- At least as popular as a polynomial kernel is the radial kernel:
$$\exp\left(-\|\mathbf{x}-\mathbf{x}'\|^2/c\right). \tag{15}$$
- In general, our aim is to classify any new observation into the class whose training examples are more similar to this observation than those from the other class. It is the kernel which provides a measure of this similarity.
- It should be emphasized that, e.g., the soft margin SVM includes regularization in its formulation.
- Proper value of the SVM's regularization parameter C (or λ - see below) can be found by cross-validation; a path algorithm for the SVM classifier is known.
- The problem with more than two classes has to be dealt with separately.

Support Vector Machines (SVM) - contd.

For a discussion of kernel-based SVMs as a similarity-based classifiers see a separate handout (note also how kernels relate to distances in the feature space).

The problem with more than two classes will be dealt with during the lecture.

PhD STUDIES
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

SVM as a regularized (or penalization) method

For $\tilde{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \cdot \mathbf{w} + b$ consider the optimization problem

$$\text{minimize}_{\mathbf{w},b} \sum_{i=1}^{n} [1 - y_i \tilde{f}(\mathbf{x}_i)]_+ + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

It can be shown that the solution to this problem, with $\lambda = 1/C$, is the same as that for the problem (8) - (11); here the feature space is the space of points $\mathbf{h}(\mathbf{x})$. The classifier then is: classify $\mathbf{x}$ to the class equal to $\text{sign}\{\tilde{f}(\mathbf{x})\}$.

Note that the above casts the SVM (with hinge loss) as a regularized function estimation problem. Note also that SVMs for other loss functions can be considered too.

It also follows from our earlier discussion that the last optimization problem, as well as a host of other problems of this type, can be solved within the broad framework of Reproducing Kernel Hilbert Spaces (RKHS).

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

A remark on primal and dual optimization problems

When studying papers on SVMs, we most often learn that we in fact solve the dual optimization problem. In our study we also switched to the dual problem. Let us recall that the primal problem reads:

$$\text{minimize } f(\mathbf{w}), \ \ \mathbf{w} \in \Omega \subseteq R^d,$$

$$\text{subject to } g_i(\mathbf{w}) \leqslant 0, \ i = 1, \ldots k,$$

while the dual of the primal problem is:

$$\text{maximize } \theta(\boldsymbol{\alpha}),$$

$$\text{subject to } \boldsymbol{\alpha} \geqslant \mathbf{0},$$

where $\theta(\boldsymbol{\alpha}) = \inf_{\mathbf{w} \in \Omega} L(\mathbf{w}, \boldsymbol{\alpha})$ with $L(\cdot, \cdot)$ being the corresponding Lagrangian function.

PhD STUDIES
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
The project is co-financed by the European Union within the framework of European Social Fund

A brief intro to kernels: Kernels as inner products

Definition: A kernel is a function $\mathcal{K}$ that for all $\mathbf{x}, \mathbf{z} \in \mathbf{X}$, $\mathbf{X}$ being a nonempty set,

$$\mathcal{K}(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z}),$$

where ϕ is a mapping from $\mathbf{X}$ to an inner product (dot product) space $\mathbf{F}$,

$$\phi : \mathbf{x} \mapsto \phi(\mathbf{x}) \in \mathbf{F}.$$

$\mathbf{F}$ will be called a feature space.

Given a set $\mathbf{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$, the $\ell \times \ell$ matrix $\mathbf{G}$ with entires

$$G_{ij} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$$

is called the kernel matrix (or Gramm matrix); it gives the inner products in a feature space with feature map ϕ.

Kernels as inner products

Proposition: Kernel matrix is positive semi-definite.

Definition: A function

$$\mathcal{K} : \mathbf{X} \times \mathbf{X} \rightarrow R \tag{16}$$

satisfies the finitely positive semi-definite property if it is a symmetric function for which the matrices formed by restriction to any finite subset of $\mathbf{X}$ are positive semi-definite.

Theorem: A function (1), which is either continuous or has a countable domain, can be decomposed

$$\mathcal{K}(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$$

into a feature map ϕ into a Hilbert space $\mathbf{F}$ applied to both its arguments followed by the evaluation of the inner product in $\mathbf{F}$ if and only if it satisfies the finitely positive semi-definite property.

Kernels as inner products

Kernels satisfy several closure properties. In addition, we have

Proposition: Let $\mathcal{K}_1(\mathbf{x}, \mathbf{z})$ be a kernel over $\mathbf{X} \times \mathbf{X}$ and $p(x)$ be a polynomial with positive coefficients. Then the following functions are also kernels:

$$\mathcal{K}(\mathbf{x}, \mathbf{z}) = p(\mathcal{K}_1(\mathbf{x}, \mathbf{z})),$$
$$\mathcal{K}(\mathbf{x}, \mathbf{z}) = \exp(\mathcal{K}_1(\mathbf{x}, \mathbf{z})),$$
$$\mathcal{K}(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2/(2\sigma^2)).$$

The above observations, when combined with the fact that we can treat similarities as kernels or kernels as similarities, as well as with the fact that many known classification algorithms depend only on inner products, reveal huge advantages we can gain using kernel methods properly. Indeed, it is of utmost importance that a kernel function can be defined over a set of points of next to any structure (bags of words, strings, trees, graphs, and much more). Moreover, we can use the kernel trick (for, say, SVMs) or resort to spectral decomposition of the Gramm matrix within the clustering framework.

Kernels for text

Given a bag-of-words (within a vector space model) write the document-term matrix $\mathbf{D}$ and take as kernel

$$\mathcal{K} = \mathbf{D}\mathbf{D}',$$

where i-th, $i = 1, \dots, \ell$, row of $\mathbf{D}$ corresponds to the i-th document in the feature space obtained by the mapping

$$\phi : d \mapsto \phi(d) = (tf(t_1, d), \dots, tf(t_N, d))'$$

with obvious notations.

Or, one can take, e.g.,

$$\bar{\mathcal{K}}(d_i, d_j) = (\mathcal{K}(d_i, d_j) + 1)^d.$$

Kernels for text

Of course, you can (or should) normalize if documents are of different lengths:

$$\hat{\mathcal{K}}(d_i, d_j) = \frac{\mathcal{K}(d_i, d_j)}{\mathcal{K}(d_i, d_i)\mathcal{K}(d_j, d_j)}.$$

Moreover, it is equally straightforward to extend the above to a semantic kernel by replacing ϕ with $\tilde{\phi} = \phi\mathbf{S}$; i.e.,

$$\tilde{\mathcal{K}}(d_i, d_j) = \tilde{\phi}(d_i)\mathbf{S}\mathbf{S}'\tilde{\phi}(d_j),$$

where

$$\mathbf{S} = \mathbf{R}\mathbf{P},$$

R is a diagonal matrix which assigns weights to the terms (say, *idf* weights) and **P** is a matrix which defines semantic affinities between the terms.