

Aprendizaje transductivo con TSVM en la brecha semántica para el reconocimiento de imágenes

Carlos Peña Peña

Universidad Inca Garcilaso de la Vega, FISCT
Lima, Perú, Av. Bolívar 1848 – Lima 21
cpenapena@uigv.edu.pe

Resumen: La propuesta está conformada por dos procesos de aprendizaje TSVM: el primer proceso genera soluciones globales óptimas, y el segundo proceso toma la mejor solución global para seleccionar los grupos óptimos. Los grupos seleccionados serán separados del conjunto y pasaran a conformar la solución final.

Abstract: The proposal consists of two TSVM learning processes: the first process generates optimal global solutions, and the second process takes the best overall solution to select the optimal groups, selected groups will be separated from the set and move to form the final solution.

Palabras clave: Brecha semántica, aprendizaje transductivo, TSVM.

1. Introducción

Debido a los avances en la tecnología en diversos campos de la adquisición de imágenes, informática, procesamiento de imágenes, ciencias de la información y bases de datos, el crecimiento de las imágenes digitales y su gestión se ha convertido en un tema de investigación en diferentes campos. El crecimiento de las imágenes digitales en la web ha creado enormes datos. El rápido y continuo crecimiento en la visualización digital hace que sea cada vez más difícil de encontrar, organizar, acceder y mantener la información de los usuarios visuales. Cuando los usuarios requieren piezas selectivas de información de Internet, se envía una consulta a una posible distribución, un sistema informático que determina lo que es relevante a la consulta y lo que no es, en base a lo que está en su almacén de datos. Esta tarea se conoce como recuperación de información (IR). El método tradicional para la recuperación de datos utiliza la representación de texto de la información y que se conoce como recuperación de información basada en texto o palabra clave. La investigación en esta técnica se inició en los años 1940 y varias técnicas se han desarrollado y alcanzado un alto rendimiento de la recuperación en términos de la calidad de los resultados y la eficiencia computacional [2].

La búsqueda de una imagen en una inmensa colección de imágenes es una tarea difícil. Hay una gran necesidad de herramientas de búsqueda eficaces para navegar por las acumulaciones de imágenes. Los dos enfoques de la recuperación de la imagen son el basado en el texto y el basado en contenido.

El enfoque basado en el texto puede ser rastreado hasta 1970. Este enfoque se basa en las descripciones textuales de las imágenes. Todas y cada imagen, en la base de datos, se va a anotar las palabras textuales y la búsqueda de imágenes se basa en estas palabras clave. Las limitaciones del enfoque basado en el texto de la recuperación de la imagen son la falta de coherencia del texto para describir los contenidos de las imágenes y anotaciones manuales de las imágenes [3]. La segunda es la inexactitud de anotación debido a la subjetividad de la percepción humana.

Para superar los inconvenientes anteriores en sistema de recuperación basado en el texto, recuperación de la

imagen basada en el contenido (CBIR) se introdujo en la década de 1980. En CBIR, las imágenes se indexan por su contenido visual, como, el color, la textura y las formas [1]. La diferencia fundamental entre los sistemas de recuperación basadas en el contenido y el texto base es que la interacción humana es una parte indispensable de este último sistema. Los seres humanos tienden a utilizar características de alto nivel (conceptos), como palabras clave, descriptores de texto, para interpretar las imágenes y medir su similitud. Si bien, las características de forma automática extraídas utilizando técnicas de visión por computador son en su mayoría de bajo nivel las características (color, textura, forma, disposición espacial, etc.) En general, no existe una relación directa entre los conceptos de alto nivel y las características de bajo nivel [4]. El campo de la CBIR ha sido ampliamente investigado en los últimos años.

2. Brecha Semántica

Durante la recuperación de imágenes en el sistema CBIR, los usuarios alimentar el sistema de recuperación de imagen de ejemplo, que están internamente, transforma en vectores de características y se comparan con los de la base de datos de función de vector [5]. A pesar de que las características extraídas visuales son naturales y objetivas, existe una brecha significativa entre los conceptos de alto nivel (que percibe humanos) y las características de bajo nivel (que se utilizan en la descripción de imágenes) [6]. Como ejemplo, una flor roja puede ser considerada como un sol, y un pez lo mismo que un avión, etc. Esto significa que el sistema tradicional CBIR que sólo depende de la extracción y la comparación de características primitivas no tiene conocimiento de la imagen "el contenido semántico" y no pueden satisfacer las necesidades de los usuarios", debido a la "brecha semántica" entre las funciones de bajo nivel y la riqueza de la semántica humanos [7,8]. Cómo cerrar la brecha semántica es actualmente un problema de investigación importante en CBIR [9].

Investigaciones recientes han sugerido diferentes enfoques para reducir la brecha semántica. En [1] las técnicas de reducción de la brecha semántica, son divididas en cinco categorías: categorías: (1) ontología del objeto para definir conceptos de alto nivel, (2) utilizando

aprendizaje automático para asociar las características de bajo nivel con los conceptos de consulta, (3) la introducción de realimentación por relevancia (RF) en el bucle de recuperación para el aprendizaje continuo de la intención del usuario, (4) la generación de la plantilla semántica (ST) para apoyar un alto nivel de recuperación de imagen, (5) haciendo uso del contenido visual de las imágenes y la información textual obtenida de la web de WWW (la web) para el reconocimiento de imágenes. En este trabajo nos enfocamos en el aprendizaje automático.

3. Problema

Según [10] el conjunto de entrenamiento J , tenemos N imágenes Web $J = \{I_1, \dots, I_n\}$ con las características visuales $X = \{x_1, \dots, x_n\} \subset R^b$ y la información textual $D = \{d_1, \dots, d_n\}$, donde x_i es a $b \times 1$ vector de la columna y d_i representa la descripción textual (documento) i de la imagen I_i . Este modo nos permite definir las diferentes características de un documento d_i a partir de modelos de

distintos texto. Por lo tanto, una imagen $I_i \in J$ puede ser representado por un par característico visual-textual: $I_i = (x_i, d_i)$. También denotan una imagen sin ningún tipo de información textual, como $I = (x, \Phi)$. El objetivo de CBIR es que, dada una imagen de consulta $I_q = (x_q, \Phi)$, el sistema de recuperación óptima debe devolver una lista π^* ranking que ordena las imágenes en J en función de su relevancia a la imagen de la consulta. Éste es un problema típico de clasificación. Por lo tanto, tenemos que crear una función f clasificación del conjunto de entrenamiento J a las puntuaciones de salida de rango $f(I_i | I_q)$ para cada

imagen $I_i \in J$ dado la consulta de la imagen I_q . Hay varios problemas que se deben abordar. Primero, con el fin de aprender el ranking función f del conjunto de entrenamiento, necesitamos saber la similitud semántica verdaderamente entre imágenes. En segundo lugar, tenemos que manejar el problema de escalabilidad. Aunque teóricamente se puede resolver con algoritmo aprendizaje global lineal [11] o no lineal [12], por otro lado, un simple clasificador lineal no puede separar bien las diferentes clases en un escenario complicado. Generalmente, para el problema de escalabilidad se utilizan métodos no lineales.

La medida de similitud utilizada será la distancia de Mahalanobis, en ella se representará los datos con una distribución no esférica que toma en cuenta las correlaciones entre las diferentes características. Los inconvenientes de esta distancia incluyen su alta complejidad computacional, y los problemas asociados con la covarianza computación e invirtiendo la matriz (Σ^{-1}) para altos espacios características dimensionales. Una solución sencilla que se tomará en cuenta para este problema es utilizar una matriz diagonal en lugar de la matriz de covarianza completa.

$$d^2(x, y) = (x - y)^T \sum^{-1} (x - y),$$

4. Aprendizaje automático

Existen tres técnicas que utilizan el aprendizaje automático para aprender conceptos de una data, [4, 12,

13,14], aprendizaje supervisado, aprendizaje no supervisado y aprendizaje semisupervisado.

El aprendizaje supervisado utiliza todos los datos etiquetados disponibles para entrenar o aprender un modelo. El objetivo del aprendizaje supervisado es predecir el valor de una medida de resultado basado en un conjunto de medida de entrada. El aprendizaje supervisado puede utilizar datos no etiquetados para la formación de un modelo. Escasos datos etiquetados pueden degradar el rendimiento de aprendizaje supervisado [15].

En el aprendizaje no supervisado, a diferencia de aprendizaje supervisado en el cual la presencia de la variable de resultado guía el proceso de aprendizaje, aprendizaje no supervisado no tiene ninguna medición de resultados, la tarea es más bien encontrar cómo la función de entrada son organizados o agrupados.

Aprendizaje semisupervisado (SSL) utiliza el conocimiento de datos etiquetados y sin etiquetar para conocer los modelos que pueden utilizarse para predecir los datos invisibles. Según [18] en el SSL, los datos de entrenamiento se complementa con un conjunto de muestras de datos sin etiquetar, es decir, tenemos

$$\{(x_1, y_1), \dots, (x_\ell, y_\ell), x_{\ell+1}, x_{\ell+2}, \dots, x_{\ell+u}\},$$

donde por lo general no hay pocos datos etiquetados y más datos no etiquetados, es decir $\ell \ll u$. Donde $N = \ell + u$ denota el tamaño del conjunto de datos. En el SSL el problema es asignar etiquetas a las datos sin etiquetar del conjunto de datos, utilizando la información proporcionada tanto por los datos etiquetados como por los datos no etiquetados.

También existen dos tipos de aprendizaje que se basan en la generalización: el aprendizaje inductivo y el aprendizaje transductivo [17]. En el aprendizaje inductivo, la tarea es construir un concepto de un conjunto de datos de entrenamiento que permitan generalizar a todos los datos, donde la data de prueba es desconocida en el momento de la elaboración. El aprendizaje transductivo [18] es similar al SSL que utiliza el conocimiento con datos sin etiqueta, sin embargo, el aprendizaje transductivo no aprende de un modelo. Para predecir nuevos datos no etiquetados utiliza los datos etiquetados, así como los datos sin etiquetar para predecir los datos sin etiquetar. La transducción se basa en empezar a resolver el problema más sencillo, es decir, no construir un clasificador inductivo para el conjunto del dominio X , sino encontrar la función de la decisión sólo de los puntos de prueba, $f: X_u \rightarrow Y$, donde X_u denota el conjunto $\{x_{\ell+1}, \dots, x_{\ell+u}\}$.

5. Aprendizaje transductivo

En esta sección, revisaremos algunos métodos aplicados al aprendizaje transductivo:

5.1. Transductive SVM de Joachims

Joachims [19] resuelve el problema de optimización combinatoria sobre las pseudoetiquetas \hat{y}_j de las no etiquetadas x_j . Estos parámetros adicionales de

optimización combinatoria se pueden eliminar por completo cuando las restricciones $\hat{y}_j \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 - \xi_i$ se expresan mediante valores absolutos $\xi_j = \max\{1 - |\langle \mathbf{w}, \mathbf{x}_j \rangle|, 0\}$. El resultado del

problema permanece no-convexa, pero ahora es continua y tiene un menor número de parámetros. Por lo tanto, se puede optimizar más rápidamente, y la recuperada son sustancialmente mejores.

```

Algorithm TSVM:
Input:      - training examples  $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$ 
            - test examples  $\vec{x}_1^*, \dots, \vec{x}_k^*$ 
Parameters: -  $C, C^*$ : parameters from OP(2)
            -  $num_+$ : number of test examples to be assigned to class +
Output:     - predicted labels of the test examples  $y_1^*, \dots, y_k^*$ 

 $(\vec{w}, b, \vec{\xi}, -) := solve\_svm\_app([\vec{x}_1, y_1] \dots [\vec{x}_n, y_n]); [C, 0, 0];$ 
Classify the test examples using  $\langle \vec{w}, b \rangle$ . The  $num_+$  test examples with
the highest value of  $\vec{w} * \vec{x}_j^* + b$  are assigned to the class + ( $y_j^* := 1$ );
the remaining test examples are assigned to class - ( $y_j^* := -1$ ).
 $C_-^* := 10^{-5};$  // some small number
 $C_+^* := 10^{-5} + \frac{num_+}{k - num_+};$ 
while( $\langle C_-^* < C^* \parallel \langle C_+^* < C^* \rangle$ ){ // Loop 1
   $(\vec{w}, b, \vec{\xi}, \vec{\xi}^*) := solve\_svm\_app([\vec{x}_1, y_1] \dots [\vec{x}_n, y_n]); [(\vec{x}_1^*, y_1^*) \dots (\vec{x}_k^*, y_k^*)]; C, C_-^*, C_+^*);$ 
  while( $\exists m, l : (y_m^* * y_l^* < 0) \& (\xi_m^* > 0) \& (\xi_l^* > 0) \& (\xi_m^* + \xi_l^* > 2)$ ){ // Loop 2
     $y_m^* := -y_m^*;$  // take a positive and a negative test
     $y_l^* := -y_l^*;$  // example, switch their labels, and retrain
     $(\vec{w}, b, \vec{\xi}, \vec{\xi}^*) := solve\_svm\_app([\vec{x}_1, y_1] \dots [\vec{x}_n, y_n]); [(\vec{x}_1^*, y_1^*) \dots (\vec{x}_k^*, y_k^*)]; C, C_-^*, C_+^*);$ 
  }
   $C_-^* := \min(C_-^* + 2, C^*);$ 
   $C_+^* := \min(C_+^* + 2, C^*);$ 
}
return  $(y_1^*, \dots, y_k^*);$ 

```

Figura 1. Algoritmo TSVM de Joachims [19]

5.2. Graph-based transductive approach

Para [20] el aprendizaje transductivo basado en Gráfico utiliza el conocimiento a partir de datos etiquetados y sin etiquetar, restringe sus predicciones sólo a los datos disponibles no etiquetados inicialmente. En lugar de construir un modelo que puede clasificar nuevas

instancias, el objetivo es clasificar todos los datos sin etiquetar. Los nodos corresponden a los datos con etiqueta y sin etiqueta y los bordes entre los nodos son ponderados sobre la base de la similitud entre las instancias correspondientes. El objetivo es definir una función, que es suave sobre los datos completos. El algoritmo es como sigue.

Graph-based transductive approach pseudo code

```

Input:    Instances of both labeled and unlabeled data represented as
            feature vectors.
Output:  Labels for all the unlabeled instances.

Algorithm:

Step 1:    Calculate the weight matrix using Equation 2.3.

Step 2:    Compute matrix Y by assigning 1 to all the labeled instances
            for the corresponding columns and 0 to the rest of the
            instances according to Equation 2.2.

Step 3:    Compute matrix S using Equation 2.4 and P as defined.

Step 4:    Compute F using the consistency method as defined in
            Equation 2.7 as well as from variant methods 1 and 2 defined.

Step 5:    The weight of a cell in matrix F represents the similarity of a
            particular instance (Row) to a particular class (Column).

Step 6:    We assign the labels by  $c_i = \operatorname{argmax}_{j \in c_i} F^*_{ij}$ 

```

Figura 2. Algoritmo Graph-based transductive approach [20].

5.3. Transductive Feature Selection (TFS)

En la figura 3, se describe el enfoque TFS. Este procedimiento integrado se puede utilizar junto para la selección de de mejores características.

1. Let \mathbf{z} an all-one vector with the same length as the input vector in \mathcal{X} .
2. Let $\bar{\mathbf{f}}$ be a set of correlation scores. Set $\mathbf{z} \leftarrow \mathbf{z} * \bar{\mathbf{f}}$. Optionally discretize \mathbf{z} to 0 and 1 by choosing a threshold.
3. Solve the following TSVM variant of **P2**

Minimize over $(y_1^*, \dots, y_k^*, \mathbf{w}, b, \xi_1, \dots, \xi_m, \xi_1^*, \dots, \xi_k^*)$:

$$\frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^m \xi_i + C^* \sum_{j=1}^k \xi_j^*$$

$$\text{subject to: } \forall_{i=1}^m : y_i (\langle \mathbf{w} \cdot \phi(\mathbf{z} * \mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \xi_i > 0$$

$$\forall_{j=1}^k : y_j^* (\langle \mathbf{w} \cdot \phi(\mathbf{z} * \mathbf{x}_j) \rangle + b) \geq 1 - \xi_j^*, \xi_j^* > 0$$

if the above TSVM is hard to converge, switch to solve the inductive SVM formulated in the **P5**.

4. Let $\bar{\mathbf{f}}$ be the approximation to the \mathbf{w} according to (Eq.1 or 2). Set $\mathbf{z} \leftarrow \mathbf{z} * \bar{\mathbf{f}}$, possibly discretize \mathbf{z} to binary as in step 2, or set part of small entries of \mathbf{z} to zero.
5. Go back to 4 until enough iterations reached or an expected size of features obtained.
6. Report the features and the predictions based on a final TSVM.

Figura 3. Algoritmo Transductive Feature Selection [21].

5.4. Unconstrained Transductive SVMs

5.5. En la figura 4 se describe detalladamente el algoritmo Unconstrained Transductive SVMs

Input: Labeled data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, unlabeled data $\{\mathbf{x}_j^*\}_{j=n+1}^{n+m}$; parameters $C_l, C_u, \epsilon_\alpha > 0$.

```

1: repeat
2:   for each labeled example  $(\mathbf{x}_i, y_i)$  do
3:      $\bar{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \neq y_i} \{\Delta(\mathbf{y}_i, \mathbf{y}) - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle\}$  // compute worst margin violater
4:     if  $\ell_{\Delta(\mathbf{y}_i, \bar{\mathbf{y}}), \epsilon}(\langle \mathbf{w}, \Phi(\mathbf{x}_i, y_i) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle) > 0$  then
5:        $W \leftarrow W \cup \{(i, y_i, \bar{\mathbf{y}})\}$  // add difference vector to working set
6:     end if
7:   end for
8:   for each unlabeled example  $\mathbf{x}_j^*$  do
9:      $\hat{\mathbf{y}}_j^* \leftarrow \operatorname{argmax}_{\mathbf{y}} \{\langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \mathbf{y}) \rangle\}$  // compute top scoring output
10:     $\bar{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \neq \hat{\mathbf{y}}_j^*} \{\Delta(\mathbf{y}_j^*, \mathbf{y}) - \langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \mathbf{y}) \rangle\}$  // compute runner-up
11:    if  $\exists \mathbf{y}_j^* \in W \wedge \mathbf{y}_j^* \neq \hat{\mathbf{y}}_j^*$  then
12:       $\forall \bar{\mathbf{y}} : W \leftarrow W \setminus \{(j, \mathbf{y}_j^*, \bar{\mathbf{y}})\}$  // delete old constraints
13:    end if
14:    if  $u_{\Delta(\mathbf{y}_j^*, \bar{\mathbf{y}}), \tau}(\langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \mathbf{y}_j^*) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}_j^*, \bar{\mathbf{y}}) \rangle) > 0$  then
15:       $\mathbf{y}_j^* \leftarrow \hat{\mathbf{y}}_j^*$ 
16:       $W \leftarrow W \cup \{(j, \mathbf{y}_j^*, \bar{\mathbf{y}})\}$  // add difference vector to working set
17:    end if
18:   end for
19:    $\alpha \leftarrow \operatorname{argmin}_{\alpha'} \text{TSVM}(\alpha', W)$  // minimize Eq. 5 by conjugate gradient descent
20:    $\forall \alpha_{k\mathbf{y}\bar{\mathbf{y}}} < \epsilon_\alpha : W \leftarrow W \setminus \{(k, \mathbf{y}_k, \bar{\mathbf{y}})\}$  // delete unnecessary constraints
21: until convergence

```

Output: Optimized α , working set W .

Figura 4. Algoritmo unconstrained Transductive SVMs [22]

6. Solución propuesta

Algoritmo Selección de conjuntos TSVM

Entrada: S^1, S^4

1. Inicio

2. Calcular en S^1

$$2.1 \quad \min f(S_k^1) = \sum_{x_{i,j,n} \in X} f_{ca}(x_{i,j,n}) + \sum_{c_u \in C} f_{dm}(c_u) + \sum_{x_{i,j,n} \in X} f_{cc}(x_{i,j,n}) + \sum_{c_u \in C} f_{ea}(c_u)$$

donde $k = 1, \dots, |S^1|$

3. $sw := 1$

4. Para cada $k = 1, \dots, |S^1|$

4.1. Si $f(S_k^1) > 0$

4.1.1. $sw := 0$

4.2. Fin de Si

5. Fin de Para

6. Si $sw := 1$

7. Para cada $k = 1, \dots, |S^1|$

7.1. Si $S_k^1 = S_i^4$ para $i = 1, \dots, |S^1|$

7.1.1. $\Rightarrow S^5 := S^5 \cup S_k^1$

7.1.2. $A := A - S_k^1, C := C - S_k^1,$

7.1.3. $G := G - S_k^1, P := P - S_k^1,$

7.1.4. $T := T - S_k^1$

7.2. Fin de Si

8. Fin de Para

9. Fin de Si

10. Retornar S^5

11. Fin

Referencias bibliográficas

- [1] Y. Liua, D. Zhanga, G. Lua, and W.-Y. Mab, "A survey of content-based image retrieval with high-level semantics," Elsevier Journal of Pattern Recognition, vol. 40, p. 262282, 2007.
- [2] R. Tansley, The Multimedia Thesaurus: Adding A Semantic Layer to Multimedia Information, United Kingdom: University of Southampton. 2000.
- [3] Y. Alemu, Jong-bin Koh, M. Ikram, and Dong-Kyoo Kim, "Image Retrieval in Multimedia Databases: A Survey," Intelligent Information Hiding and Multimedia Signal Processing, 2009. IHH-MSP '09. Fifth International Conference on, 2009, pp. 681-689.
- [4] I. Sethi and I. Coman, "Mining association rules between low-level image features and high-level concepts," in Proceedings of the SPIE Data Mining and Knowledge Discovery, vol. III, 2001, p. 279290.
- [5] Zheng, Q.-F., and Gao, W. 2008. Constructing visual phrases for effective and efficient object-based image retrieval. ACM Trans. Multimedia Computing and Communication.
- [6] Chandrika, P., Jawahar, C.V., .2010. Multi Modal Semantic Indexing for Image Retrieval. In Proceedings of the ACM International Conference on Image and Video Retrieval, China.
- [7] Huang, R.-B., Dong, S.-L., and Du, M.-H. 2008. A Semantic Retrieval Approach by Color and Spatial Location of Image Regions. In CISP '08, Image and Signal Processing conference, pp: 466- 470, China.
- [8] Nguyen, N.V., Boucher, A., Ogier, J.-M. And TABBONE, S. 2009. Region-based semi-automatic annotation using the Bag of Words representation of the keywords. In Fifth International Conference on Image and Graphics.

- [9] Lin, W.-C., Oakes, M., and Tait, J. 2010. "Improving image annotation via representative feature vector selection", *Neuro computing* Vol. 73, pp: 1774–1782.
- [10] Wang, C., Zhang, L., and Zhang, H. 2008. Learning to Reduce the Semantic Gap in Web Image Retrieval and Annotation. In SIGIR'08, Singapore
- [11] Goldberger, J., Roweis, S., Hinton, G., and Salakhutdinov, R. Neighbourhood components analysis. In NIPS, 2005.
- [12] Torresani, L. and Lee, K. Large margin component analysis. In NIPS, 2007.
- [13] C.P. Town, D. Sinclair, Content-based image retrieval using semantic visual categories, Society for Manufacturing Engineers, Technical Report MV01-211, 2001.
- [14] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, H.J. Zhang, Image classification for content-based indexing, *IEEE Trans. Image Process.* 10 (1) (2001) 117–130.
- [15] J. Luo, A. Savakis, Indoor vs outdoor classification of consumer photographs using low-level and semantic features, International Conference on Image Processing (ICIP), vol II, October 2001, pp. 745–748.
- [16] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, 2001.
- [17] K. Tangirala. *Semi-supervised and transductive learning algorithms for predicting alternative splicing events in genes*. Thesis Master of Science. Manhattan, Kansas. Kansas State University. 2011
- [18] Z. Bodó. *Semi-supervised Learning with Kernels*. Thesis Ph.D. Cluj-Napoca Rumania. Babeş-Bolyai University. 2009
- [19] T. Joachims. Transductive inference for text classification using support vector machines. *Proceedings of the International Conference on Machine Learning*. 1999
- [20] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.
- [21] Zhi-li Wu, Chun-hung Li, Feature selection with transductive Support Vector Machine. To be presented in *Neural Information Processing Systems (NIPS 2003) workshop of feature extraction and feature selection challenge*.
- [22] Zien, A., Brefeld, U., & Scheffer, T. (2007). Transductive Support Vector Machines for Structured Variables. In Z. Ghahramani (Ed.), *ICML 2007: proceedings of the Twenty-Fourth International Conference on Machine Learning* (pp. 1183–1190). Madison, Wisconsin, USA: ACM.