Least-Squares Support Vector Gépek

Bevezetés

Készítette: Valyon József (valyon@mit.bme.hu)

Ez a bevezető a Least-Squares Support Vector Gépek (LS-SVM) alapváltozatait mutatja be az osztályozási és regressziós feladatokra. Az LS-SVM a hagyományos, Vapnik által bevezetett Support Vector Gépek (SVM) egy módosított verziója, ami egy lineáris egyenletrendszer megoldására vezet, ellentétben a hagyományos módszerben alkalmazott idő és erőforrás-igényes kvadratikus programozással. A fő előnye ennek a módszernek, hogy a számítási komplexitás ezáltal csökken.

Az LS-SVM-et először Suykens mutatta be.

Az LS-SVM a hagyományos SVM egy módosítása, a kiinduló ötletek, illetve képletek azonosak, ezért a továbbiakban az eltérésekre koncentrálunk. A fő különbség a két módszer között az, hogy az SVM feltételeket megfogalmazó egyenlőtlenségeit egyenlőségekre cseréljük. Ez megfelel az ε-insensitive költségfüggvény négyzetes veszteségfüggvényre való lecserélésének.

Az LS-SVM osztályozó

Adott egy $\{\mathbf{x}_i, d_i\}_{i=1}^N$ tanító ponthalmaz, ahol $\mathbf{x}_i \in R^p$ egy p-dimenziós bemeneti vektor, illetve a $d_i \in [0,1]$ a kívánt kimenet. A cél egy olyan modell megadása, ami jól reprezentálja a tanítópontok által leírt kapcsolatot.

Az osztályozó esetében az optimálási feladat az alábbi:

$$\min_{w,b,e} J_p(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2$$

a

$d_k \left[\mathbf{w}^T \varphi(\mathbf{x}_i) + b \right] = 1 - e_k$, ahol $k = 1,...,N$

feltételekkel.

Az osztályozó ekkor a következő alakban írható fel:

$$y(x) = sign\left[\mathbf{w}^T \varphi(\mathbf{x}_i) + b \right]$$

ahol a $\varphi(.) : \Re^n \rightarrow \Re^{n_h}$ leképezés az SVM-ben is használt nemlineáris leképezés egy magasabb dimenziós térbe (feature space).

A fenti képletben a négyzetes hibát megadó e_k –a négyzetes költségfüggvényből adódó (squared loss function)– hibaértékek szerepe az SVM-ben bevezettet ξ_k slack (gyengítő) változóknak felel meg. Látható, hogy így a feltételeket megadó egyenlőtlenségek helyett egyenlőségeket adhatunk meg.

A fenti egyenletekből az alábbi Lagrange multiplikátoros egyenlet írható fel:

$L(w,b,e;\alpha) = J_p(w,e) - \sum_{k=1}^{N} \alpha_k \{d_k [\mathbf{w}^T \varphi(\mathbf{x}_i) + b] - 1 + e_k\}$, ahol az α_k értékek a Lagrange multiplikátorok, amelyek az egyenlőségi feltételek miatt pozitív és negatív értékeket is felvehetnek.

Az optimumra vonatkozó feltételek az alábbiak.

$$\frac{\partial L}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{k=1}^{N} \alpha_k d_k \varphi(x_k)$$
$$\frac{\partial L}{\partial b} = 0 \quad \rightarrow \quad \sum_{k=1}^{N} \alpha_k d_k = 0$$
$$\frac{\partial L}{\partial e_k} = 0 \quad \rightarrow \quad \alpha_k = \gamma e_k \qquad k = 1,...,N$$
$$\frac{\partial L}{\partial \alpha_k} = 0 \quad \rightarrow \quad d_k [\mathbf{w}^T \varphi(\mathbf{x}_i) + b] - 1 + e_k = 0 \qquad k = 1,...,N$$

A fenti egyenletekből a következő egyenletrendszer írható fel. A megoldás során a w és az e változókat fejezzük ki:

$$\begin{bmatrix} 0 & \mathbf{d}^{\mathbf{T}} \\ \mathbf{d} & \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix},$$

ahol:
$\mathbf{d}^{\mathbf{T}} = [d_0, d_1, ..., d_N]$,
$\vec{\mathbf{1}}^{\mathbf{T}} = [1,...,1]$,
$\boldsymbol{\alpha}^{\mathbf{T}} = [\alpha_0, \alpha_1, .., \alpha_N]$,
$\boldsymbol{\Omega}_{i,j} = d_i d_j \varphi(x_i)^T \varphi(x_j) = d_i d_j K(x_i, x_j), \quad k = 1,...,N$.

Az osztályozó –hasonlóan a hagyományos SVM-hez– a következő alakban írható fel:

$$y(x) = \sum_{k=1}^{N} \alpha_k d_k K(x, x_k) + b.$$

Az α_k súlyok és a b bias a fenti egyenletrendszerből számíthatók. Az eredmény jellemzője, hogy az összes tanítópont support vektor, azaz az egyenlőség hatására nincsenek nulla súlytényezők, azaz elveszítjük az SVM ritkasági (sparse) tulajdonságát, nevezetesen, hogy a bemeneti vektorok csak egy kis része lesz support vector. Az α_k súlyok nagyság szerinti sorba rendezéséből látszik, hogy melyek a fontosabb illetve kevésbé fontos bemeneti vektorok. Ezen alapul egy „pruning” (metszés) eljárás, ami az SVM „sparseness” (ritkasági) tulajdonságát adja vissza.
A súlytényezők nagyságából még egy súlyozó eljárás is származtatható, ami a zajos adatok kezelésében használatos. A továbbiakban először az LS-SVM regressziós változatát, majd ezeket a módszereket mutatjuk be.

Az LS-SVM regresszió

Adott egy $\{\mathbf{x}_i, d_i\}_{i=1}^N$ tanító ponthalmaz, ahol $\mathbf{x}_i \in \Re^p$ egy p-dimenziós bemeneti vektor, illetve a $d_i \in \Re$ a kívánt kimenet. A cél egy $y = f(\mathbf{x})$ függvény megadása, ami jól reprezentálja a tanítópontok által leírt kapcsolatot.

A regresszió esetén az optimálási feladat az alábbi:

$$\min_{w,b,e} J_p(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2$$

a

$d_k = \mathbf{w}^T \varphi(\mathbf{x}_i) + b + e_k$, ahol $k = 1,...,N$

feltételekkel.

Az osztályozó ekkor a következő alakban írható fel:

$y(x) = \mathbf{w}^T \varphi(\mathbf{x}_i) + b$

ahol a $\varphi(.) : \Re^n \to \Re^{n_h}$ leképezés az SVM-ben is használt nemlineáris leképezés egy magasabb dimenziós térbe (feature space).

A Lagrange multiplikátoros felírás az alábbi:

$L(w,b,e;\alpha) = J_p(w,e) - \sum_{k=1}^{N} \alpha_k \{\mathbf{w}^T \varphi(\mathbf{x}_i) + b + e_k - d_k\}$, ahol az α_k értékek a Lagrange multiplikátorok.

Az optimumra vonatkozó feltételek az alábbiak.

$$\frac{\partial L}{\partial w} = 0 \quad \to \quad w = \sum_{k=1}^{N} \alpha_k \varphi(x_k)$$

$$\frac{\partial L}{\partial b} = 0 \quad \to \quad \sum_{k=1}^{N} \alpha_k = 0$$

$$\frac{\partial L}{\partial e_k} = 0 \quad \to \quad \alpha_k = \gamma e_k \qquad k = 1,...,N$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \quad \to \quad \mathbf{w}^T \varphi(\mathbf{x}_i) + b + e_k - d_k = 0 \quad k = 1,...,N$$

A $\mathbf{w}$ és e kifejezése után a következő lineáris egyenletrendszer írható fel:

$$\begin{bmatrix} 0 & \vec{\mathbf{1}}^{\mathbf{T}} \\ \vec{\mathbf{1}} & \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{d} \end{bmatrix}$$

ahol:

$\mathbf{d}^{\mathbf{T}} = [d_0, d_1, ..., d_N]$,

$\vec{\mathbf{1}}^{\mathbf{T}} = [1,...,1]$,

$\boldsymbol{\alpha}^{\mathbf{T}} = [\alpha_0, \alpha_1, .., \alpha_N]$,

$\Omega_{i,j} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$, $k = 1,...,N$.

Az eredmény –hasonlóan a hagyományos SVM-hez– a következő alakban írható fel:

$$y(x) = \sum_{k=1}^{N} \alpha_k K(x, x_k) + b .$$

Ahol az α_k és b a fenti egyenletrendszer megoldása.

Az LS-SVM pruning

A Least-Squares megoldás egyik hátránya, hogy a megoldás nem „sparse", azaz minden tanító vektor support vektor. Ha a végeredményt egy neurális hálózatként értelmezzük, akkor ebben *N* (mintapontok száma) nemlineáris neuron található, azaz az eredmény komplexitása nagyobb, mint a hagyományos SVM-el nyert hálózaté, mely valóban szelektál a bementek közül. Ez abból is látszik, hogy a megoldásban felhasználjuk a $\alpha_k = \gamma e_k$ egyenletet. Ebből az is látszik, hogy a *k*. tanítópontban kapott hiba arányos a tanítóponthoz, mint support vektorhoz tartozó α_k súllyal. Ahhoz hogy a hagyományos SVM ritkasági tulajdonságát visszanyerjük további lépésekre van szükség.
Intuitíven állíthatjuk, hogy a kisebb α_k-k kevésbé járulnak hozzá a megoldáshoz, azaz a kialakított modellhez. A következő „pruning" (metszési) eljárás egy ezen alapuló iteratív módszer, mellyel az LS-SVM alkalmazásával is egy egyszerűbb eredményre (kisebb hálózatra) jutunk. A megoldás menete az alábbi:

1. Tanítsuk a hálózatot az összes rendelkezésre álló (*N*) tanítóponttal.
2. Távolítsuk el egy kisebb részét (pl. 5%–át) a pontoknak úgy, hogy azokat hagyjuk el, melyekhez a legkisebb $|\alpha_k|$ tartozik.
3. Tanítsuk újra az LS-SVM-et a kisebb tanítókészlettel.
4. Goto 2. , amíg a válasz minősége nem romlik. Ha a teljesítmény romlik, akkor a γ és σ (RBF hálózat esetén) hangolásával esetleg még tovább csökkenthetjük a megoldás méretét.

Weighted LS-SVM

A Least-Squares hibafüggvény a nem Gauss zaj (például „outlier"-ek) esetén nem optimális, ezért ilyen tanítómintáknál a modellt egy további módosítással hangolhatjuk. A módszer, hasonlóan a „pruning"-hoz az $\alpha_k = \gamma e_k$ egyenleten alapul. Ebből látszik, hogy az egyes pontokban kapott hiba e_k arányos az eredményként kapott α_k súlyokkal. A cél, hogy a hiba változókat, azaz az egyes tanítópontok szerepét az optimálásban v_k súlyokkal módosítsuk. Az egyenlet (a regressziós esetre felírva) ekkor a következőképpen módosul:
A regresszió esetén az optimálási feladat az alábbi:

$$\min_{w,b,e} J_p(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} v_k e_k^2$$

a

$d_k = \mathbf{w}^T \varphi(\mathbf{x}_i) + b + e_k$, ahol $k = 1,...,N$

feltételekkel.

A lineáris egyenletrendszerben ez a következőket jelenti:

$$\begin{bmatrix} 0 & \vec{\mathbf{1}}^{\mathbf{T}} \\ \vec{\mathbf{1}} & \mathbf{\Omega} + \mathbf{V}_\gamma \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{d} \end{bmatrix}$$

ahol a $\mathbf{V}_\gamma$ diagonál mátrix:

$$\mathbf{V}_\gamma = diag\left(\left[\frac{1}{\gamma v_1}; ...; \frac{1}{\gamma v_N} \right] \right)$$

A v_i súlyokat az $e_k = \frac{\alpha_k}{\gamma}$ értékek alapján választhatjuk meg, például a következő egyenlet szerint: $v_k = \begin{cases} 1 & ha \quad |e_k/s| \le c_1 \\ \frac{c_2 - |e_k/s|}{c_2 - c_1} & ha \quad c_1 \le |e_k/s| \le c_2 \\ 10^{-4} & egyébként \end{cases}$

ahol c_1, c_2 és s megválasztása a statisztikában ismert módszerek alapján történhet.

Az algoritmus menete a következő:

1. Tanítsuk a hálózatot súlyozás nélkül az összes rendelkezésre álló (N) tanítóponttal és határozzuk meg az $e_k = \frac{\alpha_k}{\gamma}$ értékeket.
2. Határozzuk meg a v_i súlyokat a fentiek szerint.
3. Számítsunk ki egy súlyozott LS-SVM modellt v_i súlyok segítségével.

Irodalom:

[1] S. Haykin: "Neural networks. A comprehensive foundation", Prentice Hall, N. J. 1999

[2] V. Vapnik: "The Nature of Statistical Learning Theory", New–York: Springer–Verlag. 1995

[3] J. A. K. Suykens, “Nonlinear Modeling and Support Vector Machines”, *IEEE Instrumentation and Measurement Technology Conference*, Budapest, Hungary, May 21–23, 2001

[4] J. A. K. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers”, *Neural Processing Letters*, vol. 9, no. 3, Jun. 1999, pp. 293–300.

[5] J. A. K. Suykens and J. Vandewalle. “Multiclass least squares support vector machines” In *IJCNN'99 International Joint Conference on Neural Networks*, Washington, DC, 1999.

[6] J. A. K. Suykens, P. Van Dooren, B. De Moor, and J. Vandewalle. “Least squares support vector machine classifiers: a large scale algorithm” In *European Conference on Circuit Theory and Design, ECCTD'99*, pages 839-842, 1999.

[7] J. A. K. Suykens, L. Lukas, and J. Vandewalle. “Sparse approximation using least squares support vector machines” In *IEEE International Symposium on Circuits and Systems ISCAS'2000*, 2000.

[8] J. A. K. Suykens, L. Lukas, and J. Vandewalle. “Sparse least squares support vector machine classifiers” In *ESANN'2000 European Symposium on Artificial Neural Networks*, pp. 37–42, 2000.