Solving Non-Linear SVM in Linear Time – A Nyström Approximated SVM with Applications to Image Classification

Ming-Hen Tsai
Academia Sinica
(now at Google Inc.)

Joint work with Yi-Ren Yeh, Yuh-Jye Lee and Yu-Chiang Wang

Outline

- Introduction
- Nyström Method for Kernel Approximation
- Nyström Method for Linear Representation
- Experiment
- Conclusion and Application Areas

Outline

Outline

- Introduction
- Nyström Method for Kernel Approximation
- Nyström Method for Linear Representation
- Experiment
- Conclusion and Application Areas

We Have Known..

- Non-linear SVM: powerful but slow
- Linear SVM: simple but fast

  Paper: "Training Linear SVMs in Linear Time" by Joachims

  Software: LIBLINEAR, VW, etc.

We Had Always Wondered...

- But we want something **powerful** and **fast**: train faster than non-linear SVM and generate a more accurate model than linear SVM.

We Had Always Wondered...

- But we want something **powerful** and **fast**: train faster than non-linear SVM and generate a more accurate model than linear SVM.

  What should we do?

We Had Always Wondered...

- But we want something **powerful** and **fast**: train faster than non-linear SVM and generate a more accurate model than linear SVM.

  What should we do?
- Can we save some computations (for speed) and still get non-linearity (for more power)?

We Had Always Wondered...

- But we want something **powerful** and **fast**: train faster than non-linear SVM and generate a more accurate model than linear SVM.

  What should we do?
- Can we save some computations (for speed) and still get non-linearity (for more power)?

  Yes. Do approximation!

Outline

- Introduction
- Nyström Method for Kernel Approximation
- Nyström Method for Linear Representation
- Experiment
- Conclusion and Application Areas

Definitions and a Brief Review (1/2)

- Input Data: $\{(\mathbf{y}, X)\} = \{(y_i, \mathbf{x}_i)\}_{i=1}^{\ell}$. y_i: label, $\mathbf{x}_i$: feature vector.
- (Non-zero) dimension of $\mathbf{x}_i$: n
- Non-linear mapping $\phi(\mathbf{x})$ (e.g. bi-gram features)
- Kernel function: $K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u})^T \phi(\mathbf{v})$, usually computed in linear time.

Definitions and a Brief Review (1/2)

- Input Data: $\{(\mathbf{y}, X)\} = \{(y_i, \mathbf{x}_i)\}_{i=1}^{\ell}$. y_i: label, $\mathbf{x}_i$: feature vector.
- (Non-zero) dimension of $\mathbf{x}_i$: n
- Non-linear mapping $\phi(\mathbf{x})$ (e.g. bi-gram features)
- Kernel function: $K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u})^T \phi(\mathbf{v})$, usually computed in linear time.
- Lower bound for data storage and training: $\Omega(\ell n)$.

Definitions and a Brief Review (2/2)

- Primal:

$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{\ell} \max(1 - y_i\mathbf{w}^T\phi(\mathbf{x}_i), 0)$$

- Dual:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}$$
$$\text{s.t. } 0 \le \alpha_i \le C \quad \forall i.$$

- Primal-Dual Correspondence: $\mathbf{w} = \sum_{i=1}^{\ell} y_i\alpha_i\phi(\mathbf{x}_i)$

Kernel in Non-linear SVM

- Dual form:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha}$$
$$\text{s.t. } 0 \leq \alpha_i \leq C \quad \forall i.$$

- Kernel matrix Q with $Q_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.

Kernel in Non-linear SVM

- Dual form:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}$$
$$\text{s.t. } 0 \leq \alpha_i \leq C \quad \forall i.$$

- Kernel matrix Q with $Q_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.
  Computational time: $O(\ell^2)$ times kernel products $\sim O(\ell^2 n) \sim \ell$ times data size.
  Space: $O(\ell^2)$.

Kernel in Non-linear SVM

- Dual form:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \mathbf{e}^T \boldsymbol{\alpha}$$
$$\text{s.t. } 0 \leq \alpha_i \leq C \quad \forall i.$$

- Kernel matrix Q with $Q_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.
  Computational time: $O(\ell^2)$ times kernel products $\sim O(\ell^2 n) \sim \ell$ times data size.
  Space: $O(\ell^2)$.
- $1M$ images, $1k$ dimensional features: $1M$ times more computational time, and $1M/1k = 1k$ times more space than the linear counterpart.

Nyström Method

- Sample k feature vectors from X to a set $B = \{\mathbf{b}_i\}_{i=1}^{k}$.
- The apprxoimated kernel: $\tilde{Q} \sim CW^{-1}C^T$, where $C_{ij} = K(\mathbf{x}_i\mathbf{b}_j)$ and $W_{ij} = K(\mathbf{b}_i, \mathbf{b}_j)$.

  so C is ℓ by k, W is k by k. Usually $n \sim k \ll \ell$.
- Diffenernt studies on how to get a small and representative basis B.

Nyström Method

- Sample k feature vectors from X to a set $B = \{\mathbf{b}_i\}_{i=1}^{k}$.
- The apprxoimated kernel: $\tilde{Q} \sim CW^{-1}C^T$, where $C_{ij} = K(\mathbf{x}_i\mathbf{b}_j)$ and $W_{ij} = K(\mathbf{b}_i, \mathbf{b}_j)$.

  so C is ℓ by k, W is k by k. Usually $n \sim k \ll \ell$.
- Diffenernt studies on how to get a small and representative basis B.
- Time and space complexity:

  Time: $O(\ell kn)$ for C, $O(k^2 n) + O(k^3)$ for W^{-1}, and $O(\ell^2 k)$ for $\tilde{Q}$. $O(\ell^2 k)$ overall. (Alright...)

  Space: $O(\ell k)$. (Good!)

Outline

- Introduction
- Nyström Method for Kernel Approximation
- Nyström Method for Linear Representation
- Experiment
- Conclusion and Application Areas

An Equivalent Representation

The Equivalent Representation in Nyström Method

Outline

- Introduction
- Nyström Method for Kernel Approximation
- Nyström Method for Linear Representation
- Experiment
- Conclusion and Application Areas

Outline

- Introduction
- Nyström Method for Kernel Approximation
- Nyström Method for Linear Representation
- Experiment
- Conclusion and Application Areas

Conclusion and Future Work