

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

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

Joint work with

Yi-Ren Yeh, Intel-NTU Connected Context Computing Center

Yuh-Jye Lee, Dept. CSIE, National Taiwan University Science & Technology

Yu-Chiang Frank Wang, Research Center for IT Innovation, Academia Sinica

May 21, 2013

Outline

- Introduction
- The Nyström Method for Kernel Approximation
- The Nyström Method for Linear SVM Classification
- Experiments and Conclusions

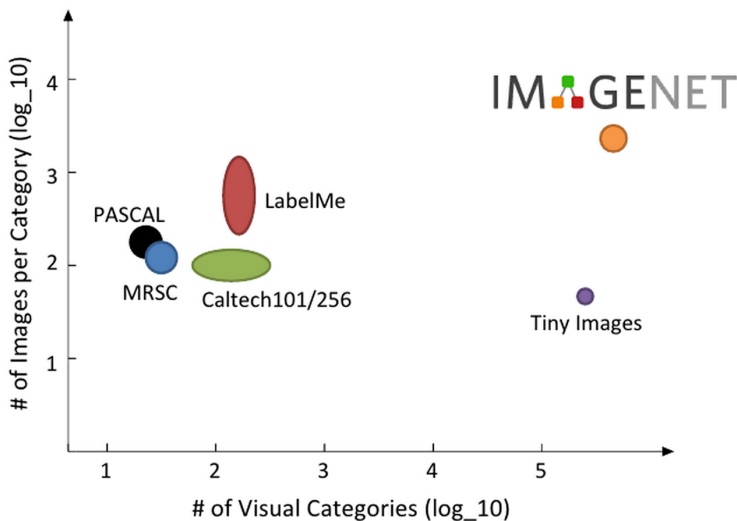
Outline

- Introduction
- The Nyström Method for Kernel Approximation
- The Nyström Method for Linear SVM Classification
- Experiments and Conclusions

Visual Classification: Faces, Objects, and Beyond



Example: Some Image Classification Data Sets



We Have Known...

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

Paper: “Training Linear SVMs in Linear Time” by Joachims
KDD’06

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.

Is this even possible?

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.

Is this even possible?

Yes. Do approximation!

Outline

- Introduction
- The Nyström Method for Kernel Approximation
- The Nyström Method for Linear SVM Classification
- Experiments and Conclusions

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 : d
- 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.
- For ease of representation, for $U = [\mathbf{u}_1, \dots, \mathbf{u}_{\ell}]$ and $V = [\mathbf{v}_1, \dots, \mathbf{v}_{\ell}]$, we define $Q = K(U, V)$, where $Q_{ij} = K(\mathbf{u}_i, \mathbf{v}_j)$

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 : d
- 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.
- For ease of representation, for $U = [\mathbf{u}_1, \dots, \mathbf{u}_{\ell}]$ and $V = [\mathbf{v}_1, \dots, \mathbf{v}_{\ell}]$, we define $Q = K(U, V)$, where $Q_{ij} = K(\mathbf{u}_i, \mathbf{v}_j)$
- Lower bound for data storage and training: $\Omega(\ell d)$.

Definitions and a Brief Review (2/2)

- Primal SVM:

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

- Dual SVM:

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

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

Kernel in Non-linear SVM

- Dual form:

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

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

Kernel in Non-linear SVM

- Dual form:

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

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

Kernel in Non-linear SVM

- Dual form:

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

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

Nyström Method

- A low-rank kernel approximation method. (Why?)

Nyström Method

- A low-rank kernel approximation method. (Why?)
- Sample $\tilde{\ell} (\ll \ell)$ feature vectors from X with a basis set $B = \{\mathbf{b}_i\}_{i=1}^{\tilde{\ell}}$.

The low-rank approximated kernel: $Q \sim \tilde{Q} = P^T W^{-1} P$,
 where $P_{ij} = K(\mathbf{x}_i, \mathbf{b}_j)$ and $W_{ij} = K(\mathbf{b}_i, \mathbf{b}_j)$.

$$\begin{array}{ccccccc}
 & Q & & P & & W_{\tilde{\ell}}^{-1} & & P^T \\
 & \square & \approx & \square & \times & \square & \times & \square
 \end{array}$$

Nyström Method

- A low-rank kernel approximation method. (Why?)
- Sample $\tilde{\ell} (\ll \ell)$ feature vectors from X with a basis set $B = \{\mathbf{b}_i\}_{i=1}^{\tilde{\ell}}$.

The low-rank approximated kernel: $Q \sim \tilde{Q} = P^T W^{-1} P$,
 where $P_{ij} = K(\mathbf{x}_i, \mathbf{b}_j)$ and $W_{ij} = K(\mathbf{b}_i, \mathbf{b}_j)$.

$$\begin{array}{ccccccc}
 Q & & P & & W_{\tilde{\ell}}^{-1} & & P^T \\
 \square & \approx & \square & \times & \square & \times & \square
 \end{array}$$

- Time: Basis Selection + $O(\ell^2 \max(\tilde{\ell}, d))$ overall. (Alright...)
 Space: $O(\ell \max(\tilde{\ell}, d))$. (Good! Don't know how large is $\tilde{\ell}$..)

Nyström Method

- A low-rank kernel approximation method. (Why?)
- Sample $\tilde{\ell} (\ll \ell)$ feature vectors from X with a basis set $B = \{\mathbf{b}_i\}_{i=1}^{\tilde{\ell}}$.

The low-rank approximated kernel: $Q \sim \tilde{Q} = P^T W^{-1} P$,
 where $P_{ij} = K(\mathbf{x}_i, \mathbf{b}_j)$ and $W_{ij} = K(\mathbf{b}_i, \mathbf{b}_j)$.

$$\begin{array}{ccccccc}
 & Q & & P & & W_{\tilde{\ell}}^{-1} & & P^T \\
 & \square & \approx & \square & \times & \square & \times & \square
 \end{array}$$

- Time: Basis Selection + $O(\ell^2 \max(\tilde{\ell}, d))$ overall. (Alright...)
- Space: $O(\ell \max(\tilde{\ell}, d))$. (Good! Don't know how large is $\tilde{\ell}$..)
- Set our basis size, $\tilde{\ell}$ to d . The space consumption is optimal.

Outline

- Introduction
- The Nyström Method for Kernel Approximation
- The Nyström Method for Linear SVM Classification
- Experiments and Conclusions

An Equivalent Representation (1/2)

- Bottleneck: kernel matrix Q of size $O(\ell^2)$.
But observe the following...

An Equivalent Representation (1/2)

- Bottleneck: kernel matrix Q of size $O(\ell^2)$.
But observe the following...
- $Q = \tilde{X}^T \tilde{X}$ by Cholesky decomposition. (This is valid because Q is PSD by definition.)
- Investigate the columns of $\tilde{X} = [\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_\ell]$. We have $\tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j = Q_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$.

An Equivalent Representation (1/2)

- Bottleneck: kernel matrix Q of size $O(\ell^2)$.
But observe the following...
- $Q = \tilde{X}^T \tilde{X}$ by Cholesky decomposition. (This is valid because Q is PSD by definition.)
- Investigate the columns of $\tilde{X} = [\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_\ell]$. We have $\tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j = Q_{ij} = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$.
- We represent the kernelized linear products as regular linear products. Let's call \tilde{X} a compact representation.

An Equivalent Representation (2/2)

- SVM Dual:

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

An Equivalent Representation (2/2)

- SVM Dual:

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

- SVM Primal:

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

An Equivalent Representation (2/2)

- SVM Dual:

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

- SVM Primal:

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

- Issue: computing $\tilde{\mathbf{x}}_i$ requires the computation of the ℓ^2 -sized kernel.

Nyströms' Equivalent Representation

- Let's go back to kernel approximation!

Nyströms' Equivalent Representation

- Let's go back to kernel approximation!
- As $\tilde{Q} = P^T W^{-1} P$, we can find a matrix R such that $W^{-1} = R^T R$, and RP is our a compact representation.

Nyströms' Equivalent Representation

- Let's go back to kernel approximation!
- As $\tilde{Q} = P^T W^{-1} P$, we can find a matrix R such that $W^{-1} = R^T R$, and RP is our compact representation.

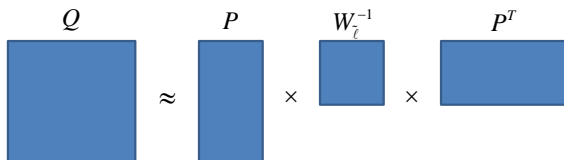
Time: Basis selection time + $O(\ell \tilde{\ell} d)$ overall.

Space: $O(\ell \max(\tilde{\ell}, d))$.

Nyströms' Equivalent Representation

- Let's go back to kernel approximation!
- As $\tilde{Q} = P^T W^{-1} P$, we can find a matrix R such that $W^{-1} = R^T R$, and RP is our compact representation.
Time: Basis selection time + $O(\ell \tilde{\ell} d)$ overall.
Space: $O(\ell \max(\tilde{\ell}, d))$.
- Needs a linear feature selection method!

Brief Recap

$$Q \approx P W_{\ell}^{-1} P^T$$


Another View of Basis Selection (1/2)

- Recall that: $P_{:,i} = [K(\mathbf{x}_i, \mathbf{b}_1), \dots, K(\mathbf{x}_i, \mathbf{b}_{\tilde{\ell}})]^T$
- Can we view RP as input data and do feature selection?

Another View of Basis Selection (1/2)

- Recall that: $P_{:,i} = [K(\mathbf{x}_i, \mathbf{b}_1), \dots, K(\mathbf{x}_i, \mathbf{b}_{\tilde{\ell}})]^T$
- Can we view RP as input data and do feature selection?
- No! R introduces dependency between dimensions.

Another View of Basis Selection (1/2)

- Recall that: $P_{:,i} = [K(\mathbf{x}_i, \mathbf{b}_1), \dots, K(\mathbf{x}_i, \mathbf{b}_{\tilde{\ell}})]^T$
- Can we view RP as input data and do feature selection?
- No! R introduces dependency between dimensions.
- We remove R ! Each dimension is then independent of other basis vector.
- L1-regularized linear SVM which runs linear time in data size is used in the work.

Another View of Basis Selection (2/2)

- Does it hurt by removing R ?

Another View of Basis Selection (2/2)

- Does it hurt by removing R ?
- Say \mathbf{p}_+ and \mathbf{p}_- are two samples with different labels.

Another View of Basis Selection (2/2)

- Does it hurt by removing R ?
- Say \mathbf{p}_+ and \mathbf{p}_- are two samples with different labels.
- If for a \mathbf{w} , $\mathbf{w}^T \mathbf{p}_+ > \mathbf{w}^T \mathbf{p}_-$ (separable), then $\mathbf{w}^T R^{-1} R \mathbf{p}_+ > \mathbf{w}^T R^{-1} R \mathbf{p}_-$.

Another View of Basis Selection (2/2)

- Does it hurt by removing R ?
- Say \mathbf{p}_+ and \mathbf{p}_- are two samples with different labels.
- If for a \mathbf{w} , $\mathbf{w}^T \mathbf{p}_+ > \mathbf{w}^T \mathbf{p}_-$ (separable), then $\mathbf{w}^T R^{-1} R \mathbf{p}_+ > \mathbf{w}^T R^{-1} R \mathbf{p}_-$.
- $R^{-1} \mathbf{w}$ separates $R \mathbf{p}_+$ and $R \mathbf{p}_-$.

Another View of Basis Selection (2/2)

- Does it hurt by removing R ?
- Say \mathbf{p}_+ and \mathbf{p}_- are two samples with different labels.
- If for a \mathbf{w} , $\mathbf{w}^T \mathbf{p}_+ > \mathbf{w}^T \mathbf{p}_-$ (separable), then $\mathbf{w}^T R^{-1} R \mathbf{p}_+ > \mathbf{w}^T R^{-1} R \mathbf{p}_-$.
- $R^{-1} \mathbf{w}$ separates $R \mathbf{p}_+$ and $R \mathbf{p}_-$.
- When doing the linear transformation by R , we preserve separability.

The Algorithm Workflow

- 1 Input data: (\mathbf{y}, X) , a kernel K .
- 2 Run basis selection on $K(X, B)$.
- 3 Compute the new data $\tilde{X} = RK(X, B)$, where $R^T R = K(B, B)^{-1}$.
- 4 Train on (\mathbf{y}, \tilde{X})

The Algorithm Workflow

- 1 Input data: (\mathbf{y}, X) , a kernel K .
- 2 Run basis selection on $K(X, B)$.
- 3 Compute the new data $\tilde{X} = RK(X, B)$, where $R^T R = K(B, B)^{-1}$.
- 4 Train on (\mathbf{y}, \tilde{X})
If we use $\tilde{\ell}$ samples to do 2....

The Algorithm Workflow

- 1 Input data: (\mathbf{y}, X) , a kernel K .
- 2 Run basis selection on $K(X, B)$.
- 3 Compute the new data $\tilde{X} = RK(X, B)$, where $R^T R = K(B, B)^{-1}$.
- 4 Train on (\mathbf{y}, \tilde{X})

If we use $\tilde{\ell}$ samples to do 2....

Time: $O(\ell \tilde{\ell} d)$ overall.

Space: $O(\ell \max(\tilde{\ell}, d))$.

Outline

- Introduction
- The Nyström Method for Kernel Approximation
- The Nyström Method for Linear SVM Classification
- Experiments and Conclusions

Experimental Settings

- We conduct experiments on two benchmark datasets: USPS and MNIST.
- We randomly split 70% of the data for training and the remaining 30% for testing. (Repeat 5 times)
- We perform a five-fold cross validation to select the parameters γ and C .

Table: Dataset descriptions (with the number ℓ of instances and the dimension d of the data). The sizes for storing the data ℓd and the associated kernel matrices ℓ^2 are also listed.

	ℓ	d	kernel size ℓ^2	data size ℓd
USPS	7291	256	53M	2M
MNIST	60000	780	3.6G	47M

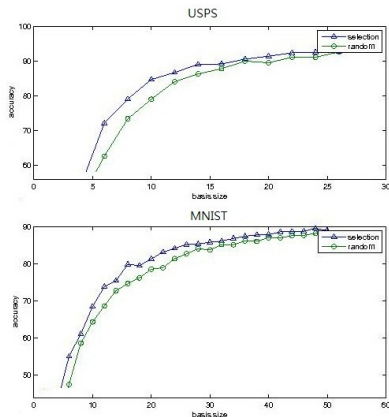
Compare Accuracy between Different Types of Methods

- Our method achieved improved accuracy than linear SVMs
- The time for training and testing using our proposed model is comparable to that of linear SVMs
- The standard nonlinear SVM utilizes the full kernel matrix whose time complexity is quadratically scaled-up with ℓ .

USPS	Accuracy	Training Time	Testing Time
Nyström primal SVM	97.057 ± 0.402	3.764	0.320
nonlinear SVM	98.007 ± 0.198	14.507	4.129
linear SVM	95.009 ± 0.275	2.274	0.079
MNIST	Accuracy	Training Time	Testing Time
Nyström primal SVM	93.833 ± 0.115	24.6092	2.006
nonlinear SVM	98.547 ± 0.066	3650.334	524.487
linear SVM	91.917 ± 0.077	14.510	0.381

Comparisons to Random Basis Selection

- Compare basis matrix determined by our method to
- the random sampling basis selection strategy



Conclusion

- Primal low-rank representation of Nyström-approximated dual SVM.

Training: linearly in time and space.

Accuracy: almost as good as non-linear SVM.

Prediction: nearly as fast as linear SVM. (e.g. Realtime robot vision applications.)

- Connect feature selection with basis selection in Nyström method.

A basis selection method that preserves separability.

- Applications: Large-Scale Image Retrieval, Realtime Machine Vision (training/test should be nearly as fast as feature extraction,) etc.