A Novel Approach for Efficient SVM Classification with Histogram Intersection Kernel

The 24th British Machine Vision Conference
BMVC 2013, Bristol, UK

Gaurav Sharma    Frederic Jurie

GREYC – University of Caen, France

firstname.lastname@unicaen.fr
Outline

1. Context and problem addressed
2. Related works
3. Proposed method
4. Experimental results
5. Conclusion
Histories and SVM in Computer Vision

Scene classification
Is this a coast/city/forest ... ?

Object classification and detection
Is there a car/bus? Where is the car/bus/...? 

Facial Analysis
Is this a man? How old is this person?

Texture classification
Is this grassy texture?

Histogram features
Bag-of-features (BoF)
Local Patterns (LBP, LTP, LQP)

Non-linear Support Vector Machine based classifier

Efficient SVM with Histogram Intersection Kernel, BMVC 2013
Scene classification
Is this a coast/city/forest ... ?

Object classification and detection
Is there a car/bus? Where is the car/bus/...?

Facial Analysis
Is this a man? How old is this person?

Texture classification
Is this grassy texture?

Histogram features
Bag-of-features (BoF)
Local Patterns (LBP, LTP, LQP)

Non-linear Support Vector Machine based classifier

Time and space efficiency!
Context: SVM primal

\[
\min_w \left\{ \frac{\lambda}{2} ||w||^2_2 + \frac{1}{|\mathcal{I}_t|} \sum_i \xi_i \right\}
\]

s.t. \( y_i w^T x_i \geq 1 - \xi_i \) and \( \xi_i \geq 0 \),

\((x_i \in \mathbb{R}^d, y_i \in \{-1, +1\})\) training (vector, label) pair

\(\mathcal{I}_t\) set of all training images

\(w\) normal to the separating hyperplane

\(\xi_i\) slack variables
Context: SVM dual and non-linearity with *kernel trick*

\[
\max_{\alpha} \left\{ \sum_i \alpha_i + \left( \frac{1}{2} - \frac{1}{\lambda} \right) \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \right\},
\]

s.t. \( 0 \leq \alpha_i \leq \frac{1}{|I_t|} \)
Depends only on dot products between $x_i, x_j$
Context: SVM dual and non-linearity with *kernel trick*

\[
\max_{\alpha} \left\{ \sum_i \alpha_i + \left( \frac{1}{2} - \frac{1}{\lambda} \right) \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j) \right\}, \text{s.t. ...}
\]

Introduce a non-linear map \( \phi : \mathbb{R}^d \rightarrow F \)
Context and problem addressed

Related works

Proposed method

Experimental results

Conclusion

Context: SVM dual and non-linearity with \textit{kernel trick}

\[
\max_{\alpha} \left\{ \sum_i \alpha_i + \left( \frac{1}{2} - \frac{1}{\lambda} \right) \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right\}, \text{s.t. } ...
\]

Use \textit{kernel trick}: \( K(x_i, x_j) = \phi(x_i)^T \phi(x_j)^T \); define \( \phi \) implicitly
Context: Test complexity of SVM with kernel trick

\[ f(\mathbf{x}) = \sum_{i} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) \text{ with } i \in \{\text{Support Vectors}\} \]

- At test time, compute kernel value with all support vectors!
- Test time is \(O(d \times N_{sv})\)
  - with \(N_{sv}\) i.e. \(#\) support vectors = \(O(\#\) training vectors\)
Problem addressed

- SVM with Histogram Intersection kernel
  - Optimize primal directly in $\mathbb{R}^d$ without kernel trick
  - Fast classifier with $O(d)$ test complexity
## Fast Intersection Kernel SVM

[Maji et al., CVPR 2008]

- Interchange order of sum over SVs and dimensions

\[
f(x) = \sum_{i \in \{SV\}} \alpha_i y_i K(x, x_i) = \sum_{i \in \{SV\}} \alpha_i y_i \left( \sum_{j=1}^{d} \min(x(j), x_i(j)) \right)
\]

\[
= \sum_{j=1}^{d} \sum_{i \in \{SV\}} \alpha_i y_i \min(x(j), x_i(j))
\]

- Sort SVs for each dimension and
  - Either, use binary search; \(O(d \times \log N_{sv})\)
  - Or, do piecewise constant or linear approximation; \(O(d \times \log N'), N' \ll N_{sv}\)
Explicit Feature Maps
[Vedaldi and Zisserman, PAMI 2012]

- Obtain approximate $\phi(x)$ explicitly
  - Fourier transforms and sampling
- Linear classifier in the feature space
- At test time
  - Map test vector to feature space
  - Compute score as dot product with linear classifier
    $O(d')$, $d' > d$

- Applies to general additive kernels
  - E.g. Intersection, Hellinger’s and $\chi^2$
Proposed method: Recap

\[
\max_\alpha \left\{ \sum_i \alpha_i + \left( \frac{1}{2} - \frac{1}{\lambda} \right) \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right\}, \text{s.t.} \quad ...
\]

\[
f(x) = \sum_{i \in SV} \alpha_i y_i K(x, x_i)
\]
Proposed method

\[ L_\phi(w_\phi) = \frac{\lambda}{2} ||w_\phi||^2 + \frac{1}{|I_t|} \sum_{i} \max(0, 1 - y_i \langle w_\phi, \phi(x_i) \rangle) \]

\[
\langle w_\phi, \phi(x) \rangle = \sum_{i \in SV} \alpha_i y_i K(x, x_i)
\]

\( w_\phi \) implicit as dual optimization gives \( \{ \alpha_i \} \)
Proposed method: Pre-image of $w_\phi$

$$L(w) = L_\phi(w_\phi) = \frac{\lambda}{2} \|\phi(w)\|^2_2 + \frac{1}{|I_t|} \sum \max(0, 1 - y_i \langle \phi(w) \phi(x_i) \rangle)$$
### Proposed method

Nonlinear SVM in input space

\[ L(\mathbf{w}) = \frac{\lambda}{2} \| \mathbf{w} \|_1 + \frac{1}{|\mathcal{I}_t|} \sum_i \max(0, m - y_i f(\mathbf{w}, \mathbf{x}_i)) \]

\[ f(\mathbf{w}, \mathbf{x}) = \sum_d \frac{w_d}{|w_d|} \min(x_d, |w_d|) \quad \text{(as } \mathbf{x} \in \mathbb{R}_+^d) \]

\[ \|\phi(\mathbf{w})\|_2^2 = \langle \phi(\mathbf{w}), \phi(\mathbf{w}) \rangle = K(\mathbf{w}, \mathbf{w}) \quad \text{(Histogram Intersection Kernel)} \]

\[ = \sum_d \frac{w_d \cdot \mathbf{w}_d}{|w_d| \cdot |w_d|} \min(|w_d|, |w_d|) = \sum_d |w_d| = \|\mathbf{w}\|_1 \]
Quasi convexity of objective

The objective is non-convex, but **quasi-convex**
- All level sets \( S_a = \{ x \mid f(x) < a \} \) are convex sets
- Intuitively the function has local ‘flat’ areas

**Proof in paper, hint in the curves below**

\[
f(x, w) = \frac{w}{|w|} \min(x, |w|)
\]

Decision function as a function of \( w \) in 2D
Successive relaxation

\[ f(w) = \begin{cases} \frac{wx}{t}, & w \leq t \\ w \tan\left\{ \frac{\theta(t-x)}{1-x} \right\}, & w > t \end{cases} \]

\[ \theta = \arctan(x) \]

Relaxation to linear decision function by varying parameter \( t \):
- \( t = x \) (Intersection)
- \( t = 1 \) (Linear)

**Related works**

\[ x \]

**Proposed method**

\[ w \leq t \]

\[ w > t \]

\[ f(w) = \min(x, w) \]

**Experimental results**

\[ t = 1 \]

**Conclusion**

Efficient SVM with Histogram Intersection Kernel, BMVC 2013
In practice, a stochastic gradient descent solver

Sub-gradient given by

\[ \nabla L_s(x_d) = \begin{cases} 
- y^i & \text{if } y^i f(w, x^i) < m \text{ and } |w_d| < x_d \\
0 & \text{otherwise} 
\end{cases} \]

The final algorithm (in paper) uses approximate sub-gradient
1. Context and problem addressed

2. Related works

3. Proposed method

4. Experimental results

5. Conclusion
Databases: Scene-15

- 15 scene classes (office, kitchen, coastal...), 4492 images
- Train on 100 random images per class and test on rest
- Repeat 10 times, report mean and std of mean class accuracy

Sharma and Jurie
Efficient SVM with Histogram Intersection Kernel, BMVC 2013
Databases: PASCAL VOC 2007

- 20 object classes (sofa, car, bike, horse etc.), 9963 images
- Split into train, val and test sets
- Report average precision (AP) for each class
Implementation details

- **VLfeat library**
- **Multiscale dense SIFT**
  - Grayscale images
  - 8 scales 1.2x apart
  - step size 3 px
- **Bag-of-features**
  - K-means on 50k randomly sampled features
  - ANN based hard quantization
  - Visual vocabulary size 1024
Scene-15 database

- **Mean class accuracy:**
  - $84.0 \pm 0.5$ for presented algorithm
  - $84.7 \pm 0.4$ for Linear SVM with feature maps
PASCAL VOC 2007 database

- Mean average precision (mAP):
  - **50.5** for presented algorithm
  - **49.7** for Linear SVM with feature maps
Testing time

<table>
<thead>
<tr>
<th>Feature map with VLfeat library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter D=3, best performing in [Vedaldi and Zisserman]</td>
</tr>
<tr>
<td>Times for classification only</td>
</tr>
<tr>
<td>Excludes time for feature computation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secs</td>
<td>Speedup</td>
</tr>
<tr>
<td>Feature map</td>
<td>3.8</td>
</tr>
<tr>
<td>Present Alg.</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Convergence for different parameter values

Fixed $m = 0.02$

Fixed $\lambda = 1e^{-4}$

Average precision

Iterations

$\lambda = 1e^{-3}$
$\lambda = 1e^{-4}$
$\lambda = 1e^{-5}$

$\lambda = 1e^{-4}$

$\lambda = 1e^{-4}$
$m = 0.04$
$m = 0.03$
$m = 0.02$
Conclusion

- Histogram Intersection Kernel SVM directly in input space
- Efficient stochastic (sub-)gradient based learning
- Efficient $O(d)$ non-linear classification

Future work
- Explore better training algorithms
- Extend to general class of kernels
- Address more applications
Thank you for your attention!

I will be happy to answer your questions.